Trace-Aware Creativity Tools

Noor Hammad

CMU-HCII-26-114
September 2026

Human-Computer Interaction Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Jessica Hammer, Co-Chair, Carnegie Mellon University
Erik Harpstead, Co-Chair, Carnegie Mellon University
Nikolas Martelaro, Carnegie Mellon University
Bryan Wang, Adobe
Max Kreminski, Cornell University

Submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy.

This research has been partially or fully funded by the
Natural Sciences and Engineering Research Council of Canada (NSERC),
the National Science Foundation (NSF), The Center For Transformational Play, and Adobe.

Copyright © 2026 **Noor Hammad**

To Adam and Elias

Keywords

Creative Activity Traces; Trace-Aware Tools; Creator Data; Personal Brand; Long-Term Creative Process; Human–AI Interaction; Creativity Tools; Creativity Support Tools; Live Streaming; Mixed Reality; Mobile Augmented Reality

Abstract

Digital media creators, including video editors, live streamers, and game developers, rely on diverse creativity tools such as game engines, video editing software, and live streaming platforms to produce their work. These ecosystems are powerful, however they often impose constraints on their users, leading to rigid interfaces that fail to adapt to how creators want to work. Existing tools can limit a creator's ability to realize ambitious ideas, offer inadequate mechanisms for communication between creators and their audiences, and provide interfaces that do not align with how creators naturally express their preferences. The first half of this thesis explores these misalignments through the development of data-driven tools, resulting in three systems that expand the design possibilities in mobile augmented reality, create novel mechanisms for streamer–audience engagement on live streaming platforms, and support video creators in expressing musical preferences through a vibe-based creative assistant. Taken together, my work on these projects revealed that the content creation process produces a wealth of valuable data, yet this data is often leveraged ad hoc to power isolated features, as opposed to being treated as a foundational resource for designing creator-aligned systems.

Building upon this realization, I argue that the future of creativity tools relies on the deliberate use of *creative activity traces* as a primary design material. I define creative activity traces as the process-related data generated as creators work, which accumulates into an evolving record of the decisions, explorations, and refinements that unfold over time. This includes things such as preferred color palettes , abandoned sketches, or successive prototype iterations. These traces are often multi-modal, distributed across tools, and deeply personalized. To understand how the HCI community currently works with traces, I conducted a systematic review of 133 creativity tools and developed a preliminary design framework for *trace awareness*. This review revealed that modern creativity tools either treat activity traces as atemporal raw material, or utilize them only as needed as input to a specific feature, thereby limiting both the temporal scope and design opportunities possible with traces. Capturing traces, in other words, does not automatically mean a system makes use of them as traces, since a creator's process becomes visible as data accumulates. Rather than using traces as isolated inputs or for post-hoc analysis, my thesis calls for traces to be treated as a design material that systems are built around. This approach in turn leads to adaptive systems that stay aligned with how creators work and how their processes unfold.

The final phase of this research builds upon my theory of creative activity traces and trace-aware tools through the study of video content creators' practices for creating, maintaining, and evolving a recognizable brand across video projects. Through a trace mapping study with seven video creators, I cataloged the trace data their practice accumulates over time, the behaviors

associated with their accumulated traces across projects, and the frictions they face as a result of their cross-project practices. Working with traces as a design material showed that trace data is not uniform, but comes in differing *grains* that afford and resist different kinds of treatment. As trace data accumulates across projects, it also forms a *texture*: a whole with a shape of its own, whose individual pieces can be difficult to disentangle.

This thesis contributes to HCI and creativity tool research by framing creative practice as an inherently data-rich process that requires more intentional engagement from the HCI research community. I formalize the definitions of creative activity traces and trace-aware tools, and I offer a design framework for trace awareness. I show what it means to treat traces as a critical building block for designing creativity tools rather than as a byproduct of creative work. Taken together, these contributions set forth a trace-aware agenda for the next generation of creativity tools, one that keeps creators and their evolving practices in view as digital creative production grows more automated and uncertain.

Acknowledgments

Six years. That is how long it took to get this dissertation across the line. Six years can feel like a long time. Like, a really long time. Enough time for one to go from moving to a foreign country straight out of undergrad in the middle of a pandemic, to being privileged enough to experience so many amazing career and personal milestones (including becoming a mother) that I would have only dreamed of. My PhD has certainly not been a straightforward journey, and as silly as it sounds, I often thought about how satisfying it would be to sit down and write the acknowledgements section of my thesis so that I can thank the people who have made all of this possible.

So, here we are, six years later. Here's to the most wonderful people in my life. This dissertation would not exist were it not for the amazing family, friends, mentors, and colleagues who have supported me through all of it.

I would first like to extend my deepest gratitude to the best PhD advisors I could have asked for, Jessica Hammer and Erik Harpstead. Your unwavering guidance and endless patience have been critical in shaping who I am as a researcher today. Looking back, I am so grateful for how you both struck the perfect balance between giving me the autonomy to experiment and make mistakes while also providing the right structure for me to continue growing throughout our time together. No one else has been as close to my research trajectory as you two have been, and have seen all of the rejections, research ideas that didn't work out, and random tangents that it took to get here. And yet, you have never shown me anything but a genuine interest in my growth and an unwavering belief that I would find my research path. Thank you for welcoming me every single week in advising meetings regardless of how much progress I had made or the challenges I was dealing with. Working with you both has been one of the great honors of my life, and you have set the perfect example for what mentorship should be. I hope that you will look back on our time together with fondness and pride just as I will.

To my thesis committee, Max Kreminski, Bryan Wang, and Nikolas Martelaro. I am 100% certain you three do some of the coolest work in the field, and I am so thankful that you not only agreed to be on my committee but were so open to talking with me about my work. I look up to each of you, and I always walk away from our meetings feeling excited and honored that I get to learn from you all. Your feedback on this dissertation has been critical to getting to this stage and for setting me up for success after the PhD. Thank you for being so inspiring and for your amazing mentorship.

This work would not have been possible without the contributions of my many co-authors and research assistants over the years. To David Lin, Amy Smith, Frank Elavsky, Sanika Moharana, Jessie Chen, Seyoung Lee, Patrick Carrington, Dominik Moritz, Thomas Eiszler, Robert Gazda,

John Cartmell, Michelle Hsieh, Ziyu Li, Qiao Zhang, and Chunyi Wang. I am sincerely thankful for your instrumental contributions to this dissertation and the ideas within it. To my research assistants, Jessie Chen, Seyoung Lee, and Shailaja Bhardwaj, thank you for your day to day collaboration and for your assistance in bringing these ideas to fruition.

My PhD has been filled with some incredible memories and great opportunities, the best of which are the internships I have had. To my mentors at Adobe, Ailie Fraser and Mira Dontcheva, thank you for your trust and for the opportunity to work at one of the coolest places in the world. To my friends at RamenVR, particularly Andy Tsen, Kristani Alcantara, and Hisham Bedri, thank you for the amazing work that you do everyday and for letting me witness what a top tier startup looks like. To my dear friends at B5 Studios, Sean Bailey, Michael Sauvage, Justin Springer, and Michael Koperwas. I am so incredibly excited about the work that you do and am so honored that you have let me into your world.

I have also had the privilege of being mentored by some wonderful people over the years. I want to extend a sincere thank you to Kory Mathewson, Gonzalo Ramos, Kathryn Eccles, Sarah Sterman, Srishti Palani, and Melissa Roemmele for being incredible mentors, especially over the last couple of years. I am grateful to be part of your circle and to know that you are the people I can turn to when I am unsure of my footing. Kory and Gonzalo especially, thank you for going out of your way for me time and again. I cannot wait to keep learning from you in the next stage of my career.

To the CTP community, Maila Rible, Elaine Fath, Jessica Kaminsky, Kaitlin Heller, and Avonelle Wing. You are all such wonderful people and it was a pleasure to work with you all these years. Thank you for being so welcoming. Thank you in particular to Kaitlin for editing this thesis and for encouraging me to believe in my own work. I would like to especially thank Avie for the dozens of coworking sessions, Slack check-ins, snack runs, and so much more. Without you, I would not have survived this PhD.

To the 2020 cohort, we did it! I'll never forget those first few months of our PhD where we were all stranded in different parts of the world and just beginning to learn what we have all gotten ourselves into. Thank you for sharing part of your journey with me and for being the first community I had during the PhD.

To the many friends I have made at CMU and the fellow PhD students who have brought so much joy into my life. Alicia DeVrio, Nathan DeVrio, Cella Sum, Lisa Egede, Sanika Moharana, Adinawa Adjagbodjou, Faria Huq, Alice Tang, Alicia Lee, Ji Youn Jung, Kimi Wenzel, Katelyn Morrison, Erica Principe Cruz, Hawra Rabaan, and Frederic Gmeiner. Thank you all for the board game nights, late night chats, delicious meals, photoshoots, and fire escapes (if you know you know). Alicia and Nathan, I am so happy that I get to call you both my dear friends, and that we were able to experience the PhD together. Who would have thought that we would be where we are today. To Cella, thank you for being an amazing friend, for entering my life at a time when I needed you most and for teaching me what justice is.

To my friends back home in Calgary, who have helped set me on this path and without whom I wouldn't have had the same passions I have today. Thank you to Ore Arowobusoye, Laura Segato, and Maiy Latif. Maiy, my fellow nerd, thank you for being the best person to talk to about movies, games, and TV shows. Laura, thank you for being my oldest friend and for making me laugh the hardest. Ore, thank you so much for choosing me to be your friend. You are the most optimistic person I know and life is so much better having known you. Thank you

for the decade plus friendship and for the endless discussions we have over text.

I am deeply indebted to the friends I made here in Pittsburgh who made all of this possible: Sarah Ashmeg, Doris Dimitriadou, Maryam Alomair, Sawsan Abu Oshaibah, Gitouf Elnema, and Asma Bedri. I came here as a single twenty-something with no family and no community, and you simply decided I was yours to look after. Thank you for all of the memories, the times you've lent me your cars, the dinner parties, the road trips, the moving days, and so much more. You are all my sisters and I am incredibly grateful for your love and kindness. Special thanks to Suhair AlAttari and Hana Mekdache, though we did not get to spend as much time together as I would have liked, I am still grateful to have known you. To my dear friend Rebecca Adaimi, your brilliance is so inspiring and I am so happy to call you my dear friend.

To my extended family in the US and Jordan, I cannot name you all here but I hope I have made you all proud. Thank you to my aunts, uncles, and grandmothers for your love and your prayers. Most especially, thank you to my family in Virginia, my aunts Inas and Mona and my Uncles Bassam and Said for welcoming me every time. And to my cousins Asma, Suemya, Moutaz, Ennis, and Farah. You are some of my closest friends. Thank you for the times you drove to Pittsburgh to see me, and for a lifetime of memories that goes back to our childhood. Being far from home was so much easier knowing you were only a drive away. I love you and am forever indebted to you all.

To Shada, Ahmad, and Jenna, the family I found away from home. I cannot even begin to express how grateful I am to have you in my life. Thank you for "adopting" me and for treating me like part of your family. Thank you for the many times that you picked me up to go somewhere to eat or have fun. Thank you for all the times we have played tarneeb together, even though we always let you win on purpose. Thank you for hosting our engagement party and for celebrating so many happy memories with me. Thank you for helping me move houses a million times, and for all the times we called you to fix something in our house. Thank you for all the laughs and inside jokes. But above all thank you for treating me as one of your own, for caring for me as much as you have and never hesitating to help when I am in need. You were a gift from above and I without a doubt would not have made it here without your kindness and love.

To my family, without whom none of this would even exist. My mom and dad, thank you for your love and support. Thank you mom for all the times you have come to Pittsburgh to care for me. For the times that you have made my favorite foods and snacks so that I can focus on my work. For introducing me to the community I now have in Pittsburgh. Dad, thank you for your encouragement and belief that your little girl can do anything. And of course thank you for your weekly reminders to cite your papers in my work to help solve whatever research problem I am facing (dear reader, my father has never published a paper in his life). To my siblings, the greatest friends in the whole world, Abood, Isra, and Lulu. Abood, thank you for being the funniest person alive and for letting me be your shadow growing up. I fell in love with creative media and technology because you let your annoying little sister follow you everywhere you went. I am so grateful for you and so proud to see you become the man you are today. To Lulu, the world's greatest travel partner and the one I have the funniest, most absurd, and meaningful memories with. Thank you for helping me more times than I can count during my time in Pittsburgh and for being here when I needed you. Thank you for being you, and for inspiring me, and for being such a force that I admire and aspire to be. To Isra, I owe everything to you. The care that you have for each of us is astounding and I am so blessed that you have

had my back all these years. Thank you for being the best siblings on the planet, for your belief, and your love. To my brother-in-law Moh, for being an amazing big brother and for taking care of me for so long. To my nephews Yousef and Karim, and my niece Layan. I love you all more than words can express and I hope that I have made you proud.

To Freya, the best dog in the whole world. To Kiwi, it would not be fair to write any of this without mentioning you, my ride or die for the last sixteen years. The world's greatest cat who has been with me every step of the way.

And finally, and most importantly, my best friend, hero, and greatest love, my husband Mahmoud. We met right at the end of my second year at a time when my confidence in my research abilities was at its lowest. You are the reason I am here today. The reason I ever believed I could genuinely succeed in anything. Thank you for your endless love. Thank you for never doubting my abilities and always believing in me. Thank you for all the sacrifices you have made for us and for believing in my dreams more than I did. Thank you for being steadfast and for being my emotional support. Thank you for all the times you have celebrated my wins and for comforting me during my losses. Thank you for being the greatest husband, and for being my confidante and my best friend. Thank you for the hundreds of hours of Overwatch we have played together and all the bets we have made while watching soccer. Thank you for pushing me to be my best self. Thank you for all the times you have helped me get my work done and for helping me focus. Every single success I have had in my PhD has been because of you, and I cannot wait to support your dreams as much as you have supported mine. Most important above all of this, thank you for the two little blessings in our lives, Elias and Adam.

Contents

Keywords iii

Abstract v

Acknowledgments x

1 Introduction 1
1.1 Creative Activity Traces and Trace-Aware Tools 2
1.2 Thesis Overview 4

2 Background 6
2.1 Creativity Tools 6
2.2 Process and Intermediate Outputs 7
2.3 Activity Traces in HCI 8
2.4 Positioning this thesis 10

I Data-Driven Creativity Tools 11

3 V-Light: Removing Computational Constraints to Support Creative Vision 12
3.1 Introduction 12
3.2 Related Work 13
3.2.1 Challenges of mAR Games 13
3.2.2 Edge Computing and Edge-Native Applications 14
3.3 The V-Light System 15
3.3.1 Interaction Model for Shared-World AR 15
3.3.2 System Architecture 16
3.3.3 Game Design and Development 19
3.4 Technical and Design Challenges 20
3.4.1 Fostering Player Body Motion 20
3.4.2 Scaffolding Real-Time Interaction 21
3.4.3 Preserving Immersion 22
3.5 Discussion 23
3.5.1 Design Implications 24
3.5.2 Technical Implications 26

3.6 Limitations and Future Work . . . 28
3.7 Conclusion . . . 29

4 MARS: Metadata-Augmented Real-Time Streaming for Audience Engagement 30
4.1 Introduction . . . 30
4.2 Related Work . . . 32
4.2.1 Augmented Streamer Interfaces . . . 32
4.2.2 Viewer Needs . . . 34
4.3 The MARS System . . . 35
4.3.1 System Overview . . . 36
4.3.2 Metadata Structure . . . 36
4.3.3 Unity Metadata Tracker . . . 38
4.3.4 Redis Middleware Service . . . 38
4.3.5 Twitch Overlay Extension . . . 39
4.4 Demos . . . 40
4.4.1 Contextual Information Cues - *Tower Defense* . . . 40
4.4.2 Accessibility Support - *Bloomwood Stories* . . . 41
4.4.3 Asymmetric Information - *Invisible Maze* . . . 42
4.5 Discussion . . . 44
4.5.1 From Capabilities To User Needs . . . 44
4.5.2 A Distributed Development Ecosystem . . . 45
4.5.3 Streaming in non-game contexts . . . 46
4.6 Limitations & Future Work . . . 46
4.7 Conclusion . . . 48

5 MARS: Towards a Design Framework for Data-Driven Game Streaming 49
5.1 Introduction . . . 49
5.2 Related Work . . . 51
5.2.1 Data-Driven Game Streaming Systems . . . 51
5.2.2 Streaming Stakeholders . . . 53
5.3 Ideation and Design Study . . . 54
5.3.1 Participants . . . 55
5.3.2 Phase One: Collaborative Ideation . . . 56
5.3.3 Inter-Phase Analysis and Phase Two Preparation . . . 57
5.3.4 Phase Two: Design Workshop . . . 58
5.4 Design Framework For Data-Driven Game Streaming . . . 61
5.4.1 Abstract Actions . . . 63
5.4.2 Experience . . . 65
5.4.3 User Interaction . . . 65
5.4.4 Data Behavior . . . 68
5.5 Analysis of An Existing Data-Driven Game Streaming System . . . 70
5.6 Discussion . . . 72
5.6.1 Insights From Designs . . . 72
5.6.2 Managing Stakeholder Incentives . . . 73

5.6.3 Toward a Unified Design Language . . . 74
5.7 Limitations and Future Work . . . 75
5.8 Conclusion . . . 76

6 VibeClip: Supporting Expressive Alignment in Generative Music Tools 78
6.1 Introduction . . . 78
6.2 Related Work . . . 79
6.2.1 Human-AI Co-Creation . . . 80
6.2.2 Music Search and Recommendation . . . 80
6.2.3 AI Music Generation . . . 81
6.2.4 Music Generation for Videos . . . 81
6.3 Formative Study . . . 82
6.3.1 Participants . . . 82
6.3.2 Procedure . . . 83
6.3.3 Results . . . 84
6.4 Design and Prototyping . . . 86
6.4.1 Design Goals . . . 86
6.4.2 Design Process . . . 88
6.4.3 Creative Assistant Prototype . . . 90
6.5 User Study . . . 94
6.5.1 Baseline Prototype . . . 94
6.5.2 Participants . . . 95
6.5.3 Procedure . . . 95
6.5.4 Data Analysis . . . 96
6.5.5 Results . . . 96
6.6 Discussion and Takeaways . . . 98
6.6.1 Offer Initial Guidance Based on User Content . . . 98
6.6.2 Surface Intermediate Structure for Productive Iteration . . . 99
6.6.3 Allow Users to Navigate Between Levels of Control . . . 100
6.7 Limitations and Future Work . . . 101
6.7.1 Ethics and Social Impact . . . 102
6.8 Conclusion . . . 103

II Creative Activity Traces and Trace-Aware Tools 104

7 Tracing Creativity: Mapping Activity Traces in Creativity Tools Research 105
7.1 Introduction . . . 105
7.2 Related Work . . . 107
7.2.1 Literature Reviews on Creativity Tools . . . 107
7.3 Method . . . 109
7.3.1 Selecting a Corpus . . . 109
7.4 Results – The Living Framework For Trace Awareness . . . 112
7.4.1 Data Characteristics . . . 113

7.4.2 Trace Awareness . . . 117
7.5 Discussion . . . 121
7.5.1 Traces as a Design Material . . . 122
7.5.2 Leveraging Traces In The Future . . . 122
7.5.3 Challenges and Risks Of Trace Usage . . . 124
7.6 Limitations and Future Work . . . 126
7.6.1 Refining The Framework . . . 126
7.6.2 Methodological Challenges . . . 126
7.6.3 Applying The Findings to Commercial Products . . . 127
7.7 Conclusion . . . 128

III Exploring Traces as a Design Material 131

8 Exploring Traces In Longitudinal Video Content Creation Practices 132
8.1 Introduction . . . 132
8.2 Background . . . 134
8.2.1 Personal Brand in Video Content Creation . . . 135
8.2.2 Traces as a Design Material . . . 135
8.3 Trace Mapping Study . . . 137
8.3.1 Participants . . . 137
8.3.2 Procedure . . . 137
8.3.3 Results . . . 139
8.4 Trace Lifecycle Model . . . 147
8.4.1 Genesis . . . 148
8.4.2 Elevation . . . 149
8.4.3 Maintenance . . . 149
8.4.4 Meaning-Making . . . 150
8.5 Inspirational Bits . . . 152
8.5.1 Bit 1: Elevation Palette . . . 153
8.5.2 Bit 2: The Oracle . . . 156
8.6 Discussion . . . 161
8.7 Limitations . . . 163

9 Conclusions 165
9.1 Personal Brand and Beyond . . . 166
9.2 Towards a Trace-Aware Research Agenda . . . 167
9.2.1 Framework Refinements . . . 168
9.2.2 Trace-Aware Research Toolkit . . . 168
9.2.3 Trace Ethics . . . 169
9.2.4 Future Applications . . . 169

References 171

List of Figures

3.1 The V-Light tech stack . . . 17
3.2 V-Light running on an Android device . . . 18

4.1 A schematic description of the MARS architecture . . . 35
4.2 An example metadata frame and corresponding game state screenshot. This example involves a simple red character that walks around the screen following set waypoints. The walker has a randomly generated secretName, which is not visible to the player. Note that this example uses a key frame rate of 1 and a tween frame rate of 3. . . . 37
4.3 The *Tower Defense* game, played by the streamer, and overlay, seen by the viewers 40
4.4 *Bloomwood Stories* game view as seen by the streamer, and overlay features as enabled by MARS . . . 41
4.5 The *Invisible Maze* game and overlay as enabled by MARS. With both sets of information, the streamer and players can successfully complete the level . . . 43

5.1 The design framework for data-driven game streaming . . . 61
5.2 A design for visually impaired users using the game *The Elder Scrolls V: Skyrim* . 62
5.3 A design for streamers interested in visualizing data generated from their stream of the game *Sid Meier's Civilization VI* . . . 63
5.4 A design for viewers to contribute to gameplay in *Portal 2* . . . 64

6.1 Early prototype sketches created during our iterative design process . . . 87
6.2 The creative assistant prototype featuring a (1) video player, (2) a multi-track timeline, and (3) a music generation panel. . . . 88
6.3 Interface shown in the music generation panel when the user selects a song for refinement. They can optionally edit any of the prompt's properties before clicking "Re-Generate," after which the "Previous Song" button will be enabled. If the user clicks "Previous Song," the UI will update to show the previous prompt, and the "Re-Generate" button will change to "Next Song," allowing them to navigate between versions. . . . 91
6.4 Overview of our prototype implementation, starting from the (1) Video Understanding step that generates transcripts and captions for (2) the vibe recommendation. Once the user submits the vibe (3) the system translates the vibe into structured music prompts for (4) the text-to-music model to generate three songs. To iterate on the songs, the user can (5) refine the structured prompt. . . . 92

6.5 The baseline prototype used in the user study. The music generation panel features a single text box where the user enters their prompt. 93
6.6 Participants' responses when asked about the recommended vibe for their videos 97
6.7 Participants' responses when asked about ease of exploring ideas using each prototype 98
6.8 Participants' responses when asked about their ability to track ideas throughout their use of each system 99
6.9 Participants' responses when asked about system clarity 100
6.10 Participants' responses when asked about feedback provided by both systems . . 101

7.1 Overview of our methodology 107
7.2 The Data Characteristics Of The Framework 108
7.3 The Trace Awareness Dimensions Of The Framework 109
7.4 Heatmap distribution of modalities across domains. 112
7.5 Distribution of systems across creative domains, with each bar showing the proportion of trace-aware and non-trace-aware systems (green is trace-aware). 117
7.6 Quadrant visualizations for three creative domains showing how they each surface or implicitly use traces. Red dots indicate non trace-aware systems 129
7.7 Distribution of trace functions across each creative domains. 130
7.8 Trace visibility and implicit usage by function 130

8.1 The trace lifecycle model for brand markers. Each stage lists its actions (blue) and frictions (red). 147
8.2 The Elevation Palette's reuse panel, asking whether a marker will be used again. . 153
8.3 The Ledger Panel, revealing which markers were reused across videos 154
8.4 The Oracle Bit's opening screen 157
8.5 The Oracle bit's three shape views 159

List of Tables

3.1 Summary of features and interaction types of *V-Light*. 19

4.1 Summary of Relevant Systems 33

5.1 Design goals derived from Phase One and the games they were matched with in Phase Two. 59
5.2 Summary of design considerations found in Phase One. 60
5.3 Example of the interaction dimensions with the *Sid Meier's Civilization VI* design. 65
5.4 Example of interaction dimensions with the *Portal 2* design. 66

5.5 Example of responsiveness with the *Portal 2* design. 68
5.6 An example of the data behavior component with *Skyrim*. 69
5.7 An example of the data behavior component with *Portal 2*. 69
5.8 A breakdown of the interaction dimensions in *Helpstone*. 70
5.9 *Helpstone's* interaction responsiveness. 71
5.10 The data behavior breakdown of *Helpstone*. 72

6.1 Information about our formative study participants, including types of videos they make, the platforms they post their videos on, and the platforms they use for music. 83
6.2 Information about our user study participants, including types of videos they make, the platforms they post their videos on, and the platforms they use for music. 94

7.1 The data characteristics of the framework 114
7.2 The trace awareness aspects of the framework 115

8.1 Information about the trace mapping study participants, including the types of videos they make, the platforms they post on, the tools they edit in, and the tools outside the editor where their trace data accumulates. 138
8.2 Brand markers and non-brand-marker trace data surfaced across the seven participants, grouped by cluster. 141

Chapter 1
Introduction

Digital media production has become a defining force in the creative media landscape [80]. From video creators with millions of followers on YouTube [223], to live streamers who showcase their gameplay on Twitch [100], and to game designers developing collaborative play experiences on mobile [312], creators across domains now use digital tools to produce increasingly sophisticated and high-fidelity work [138]. These advances have expanded what is possible creatively, but they have also introduced new demands; creators must now navigate complex workflows, integrate multiple tools and platforms, and balance creative decision-making with technical and performative labor.

Despite the apparent differences between the above creative domains, creators share several key characteristics. First, their work is inherently time-based and multimodal [126]. Unlike static media such as photography or illustration, videos, games, and live streams unfold over time and rely on sequencing, pacing, and audiovisual composition to convey a creator's intent and engage audiences.

Second, digital creators must also develop expertise across multiple areas of production and tool use. A streamer, for instance, might simultaneously play a game, perform for their audience, monitor real-time analytics, and troubleshoot broadcasting software such as OBS [195]. Video creators juggle writing, filming, editing, music and sound design [166], while game developers balance programming, level design, and narrative development using tools like Unreal Engine, Blender, and Adobe Illustrator [159]. The need to master and integrate creative and technical skills, often with little formal training, compounds the challenges of digital media production, making it an intensely demanding, albeit rewarding, form of work.

Third, these digital creators work within audience-driven ecosystems that foster strong parasocial relationships [129]. These relationships can be deeply rewarding, helping creators feel connected to their audiences and providing a sense of purpose in their work [190]. However, audience feedback, engagement metrics, and community expectations often influence creative decisions along with the creators' own creative or technical considerations [32]. In these ecosystems, creators act not only as artists or performers but also as public figures, community managers, and brand representatives. Their creative decisions are thus shaped not only by self-expression but also by audience expectations and social feedback.

Taken together, such challenges reveal an enduring gap in HCI and creativity research: how can we design better tools that meet the realities of modern digital creative work? While the social and professional pressures of digital creation are well documented, many of the difficulties creators face stem directly from the tools themselves [176, 194]. For instance, technical constraints, such as limited computational capacity, can restrict a creator's ability to realize ambitious ideas or experiment with novel designs. Furthermore, despite the importance of audience engagement, many tools provide inadequate mechanisms for facilitating meaningful interaction between creators and their audiences. Finally, interfaces often fail to reflect how creators naturally express their preferences, forcing them to adapt their workflows in undesirable ways. Prior research on creativity tools has emphasized the need for systems that align with creators' pro-

cesses and values rather than imposing rigid or uniform workflows [176, 194]. Yet many tools still fall short of this goal, offering limited responsiveness to the evolving, collaborative, and iterative nature of creative practice. If we want to support creators more effectively, we need to design tools that reflect how creative work actually unfolds, allowing creators to focus on executing their vision, engaging their audiences, and expressing themselves on their own terms.

Building on these concerns, I have explored the above challenges through the design and study of computational systems in three creative domains: mobile augmented reality, live streaming, and video content creation. V-Light, a mobile augmented reality architecture, addressed the technical constraints of mobile augmented reality game development by leveraging edge computing to support more computationally demanding gameplay experiences. Through this work, I demonstrated how offloading computation to the edge can enable designers to pursue more ambitious ideas, expanding the creative possibilities available on mobile platforms. MARS, a metadata-augmented real-time streaming system, explored how tools might better support live stream audience engagement. By rendering streamer gameplay data in real time on a per-viewer basis, MARS personalized the viewing experience and enabled new forms of sensemaking, accessibility, and audience interaction. This design helped alleviate some of the audience-management burden from streamers while also providing new mechanisms for engagement. In a follow-up study, I used MARS as co-design material with moderators, game developers, and streamers to develop a design framework for data-driven game streaming, highlighting the distinct roles different stakeholders play in shaping and responding to live stream data. VibeClip, a creative assistant for video content creators, focused on the expressive alignment between creators and their tools. It supported creators in generating emotionally aligned music for their videos, revealing that many struggled to describe desired music using traditional search interfaces. The design of VibeClip showed how generative tools can leverage video data to adapt to how creators naturally articulate their preferences.

Across these projects, one insight became increasingly clear, that is that the content creation process *itself* produces a wealth of valuable data.

1.1 Creative Activity Traces and Trace-Aware Tools

Creative production is an iterative process: creators develop their work over time [117, 267], grow their skills through the act of making [224], and rely on tools and workflows that suit their needs as they produce work for their audiences [194, 216]. Along the way, they produce *intermediate outputs* such as reference materials, annotations, partial artifacts, and iteration histories alongside the final products shown to the world [42, 123, 249, 259]. A video creator may record multiple takes before settling on the one that makes the final cut [179]; a game developer may produce dozens of prototypes, sprite sheets, and visual assets while refining a single level [159]; a streamer may experiment with different modes of audience engagement before settling into a performative persona. Although these intermediate outputs manifest differently across domains, they capture the creator's evolving decisions, provide a temporal record of how work develops, and support key aspects of creative practice such as reflection, backtracking, reuse, and adaptation across projects [267].

I refer to this material as *creative activity traces* (or simply traces), which I define as the

process-related *data* generated as creators work, forming an evolving record of the decisions, explorations, and refinements that unfold *over time*. My use of this construct is deliberately broad, encompassing data that ranges from low-level interaction logs (e.g. a streamer's gameplay metadata, edits to a script draft) to higher-level intermediate materials (e.g. sketches, drafts, prototype iterations). This breadth allows me to explore an emerging space without first having to settle what does or does not count as a trace, and to bring together concepts that have historically been studied separately under different names (e.g. design histories, data provenance, linkography [117]) and within different research traditions (e.g. design research, creativity psychology, and software engineering).

Because digital creativity tools are software, they are capable of capturing parts of this record as a creator works, which makes traces a useful lens for comparing what different tools attend to, ignore, or make visible about creative practice. Recent work in HCI has begun examining how traces can be interpreted, which techniques best make sense of them, and how trace analysis might evaluate creativity tools or inform broader theories of the creative process [163]. This dissertation contributes to that conversation by focusing on how traces are operationalized *within* creativity tools, specifically which traces a system captures, how it represents them, and how it uses them to drive its features.

Capturing traces does not mean a system makes use of them as traces. Many creativity tools hold substantial process data yet respond only to the current state of the work, treating each action in isolation from what came before. A creator's process, however, is rarely legible in any single state. The directions they abandoned, the choices they keep returning to, and the ways their work has shifted only become visible as data *accumulates* over time. I refer to systems that engage with this accumulation as *trace-aware*. A trace-aware system treats creator data not as isolated inputs to individual features, but as activity unfolding over time. It may draw on that history to help creators look back on how their work developed, surface past material for them to reuse, remix, and reorganize [267], or shape what comes next, such as suggestions aligned with a creator's style and intent.

To illustrate the difference, consider a video creator preparing the next upload in an ongoing series. Over dozens of videos, she has settled on a title style, a hand-drawn intro, and a set of transitions that her audience has come to associate with her channel. In her current editor, each new project begins empty. She reconstructs her brand by opening an older project and copying elements across, or by searching through a folder of assets she has saved over the years, and when she remembers a title treatment from a few videos ago that she never thought to save, she digs back through old projects to find it. The editor holds a detailed record of every project she has made, but it treats each one in isolation, and the work of carrying her brand from one video to the next falls entirely on her.

Now consider an editor that treats her projects as a continuing body of work. When she begins a new video, it surfaces the stylistic elements she has actually reused across recent projects, including a title style she has copied into her last several videos but never deliberately saved, and asks whether she wants to keep it on hand. As her channel grows, it can reflect back how her use of these elements has shifted over time, making visible a drift in her work that she might otherwise only sense in retrospect. It does not decide for her what her brand should be, and the parts of her practice she considers most her own, such as her intro, remain hers to author. Both editors have access to the same projects and the same data. The difference lies in how they read

it: the first treats each project as a separate snapshot, while the second follows creator data as it accumulates over time.

1.2 Thesis Overview

This dissertation follows a single question across three phases of work: how can we design creativity tools that align with the reality of creative work? The first phase approaches this question through system building. Across three system-building projects and one co-design project in mobile augmented reality, live streaming, and video content creation, I surface the underutilized role of process-related creator data in tool design. Although these systems were not built under an activity traces definition or a trace-aware framework, they reveal recurring themes about how creator data can be used to meet creator needs, and they motivate the rest of this work. The second phase formalizes creative activity traces and trace-aware tools and examines how the field currently uses them, and the third explores what it means to deliberately engage with traces as a design material within a specific creative practice.

This thesis is organized into eight subsequent chapters. Chapter 2 begins with an overview of related work in creativity tools, creative process, and activity traces in HCI. Chapter 3 presents V-Light, a mobile augmented reality architecture that leverages edge computing to offload compute-intensive tasks from mobile devices. This system was designed to help game designers prototype synchronous, co-located multiplayer AR experiences that would otherwise be infeasible due to platform limitations. While the project does not directly implement trace-aware design, it illustrates how manipulating device limitations can shape and expand the expressive possibilities available to creators. By addressing platform-level constraints in this way, V-Light supports creators in realizing technically ambitious ideas, which motivates my work towards building tools that respond to creators' goals via robust data pipelines.

Chapter 4 describes MARS (Metadata-Augmented Real-Time Streaming), a system I developed that enables novel viewing experiences by utilizing streamer gameplay data in real-time. MARS enables features like sensemaking overlays, asymmetric gameplay, and personalized viewer interfaces. In shifting some of the labor of audience engagement from streamers to data-powered overlays, MARS highlights the value of leveraging real-time creator data to support creators with their audience management needs, and suggests novel creator–audience dynamics. While not explicitly framed as trace-aware, MARS provides an early example of a system that draws from creator-side data in real time to enable richer, adaptive experiences for live streaming stakeholders, and lays the groundwork for broader considerations of creator data in tool design.

Chapter 5 builds on MARS through a follow-up co-design study with stakeholders across the live streaming ecosystem, including streamers, game developers, researchers, and moderators. Although MARS was originally built to explore novel methods for viewer engagement, the system revealed broader implications for how creator-generated data can shape a multi-stakeholder, data-driven ecosystem. Using MARS as a design probe, this study uncovered stakeholder needs, frictions around data ownership and control, and differing expectations around how features should be implemented. The resulting framework for data-driven game streaming maps out how creator data circulates across an ecosystem and reveals key tensions that arise when data be-

comes central to tool design. This chapter demonstrates the value of intentional engagement with creator data and the role that this data plays in a live ecosystem.

Chapter 6 introduces VibeClip, a creative assistant that I created to help video creators generate background music that is aligned with the emotional tone of their videos. This project began with a formative study in which creators described difficulties articulating their musical goals using formal descriptors like genre or instrumentation, which are commonly used by stock music platforms. The study revealed that non-musically trained content creators gravitate towards subjective and emotion-based descriptors, or "vibes," when describing their desired music. VibeClip was designed to support this expressive mode of communication through editable, vibe-based recommendations. In surfacing video data as a trace-like input into the generation process, the system demonstrates how generative tools can better align with creator intent and personalized working styles. My work on VibeClip was, along with MARS, the main inspiration for pursuing activity traces, as the use of creator-generated data proved to be an excellent method for aligning with creative intent.

Chapter 7 describes the theoretical bedrock of this thesis. Here, I conducted a systematic literature review of 133 creativity tool research papers in HCI to investigate how creative activity traces are currently being used to power system features. The findings were formalized into the Living Framework for Trace Awareness, which characterizes both the trace data systems capture and how systems engage with that data over time.

Finally, building upon my work in Chapter 7, Chapter 8 outlines my investigation into the longitudinal practices of video content creators as it relates to cultivating their personal brand. Through a trace mapping study with seven video creators, I cataloged the trace data their practice accumulates across projects and derived a trace lifecycle model that follows brand markers from their origin inside a single project to the long-term scope at which they accumulate into a creator's identity. I then used a material-driven prototyping method known as inspirational bits to work traces directly as a design material [277], putting the design-material stance into practice and surfacing how different kinds of trace data behave.

Together, these chapters show my trajectory from identifying creator pain points, to surfacing the underutilized value of process-related data, to establishing a theory and language around creative activity traces and trace-aware system design. By treating activity traces as central to tool design, this dissertation contributes a research agenda toward building expressive creative systems that intentionally leverage the rich potential of creative activity traces.

Chapter 2
Background

This chapter draws together several strands of HCI and creativity research that jointly ground the argument that creativity tools must engage with the creative process via the data they capture. I begin with describing the current landscape of creativity tools and Creativity Support Tools, and how their designs both enable and constrain creative practice. I then turn to research on the creative process and intermediate outputs, which establishes that the materials a creator produces while working are meaningful in their own right and can accumulate into longer creative histories. Finally, I synthesize HCI work that engages, directly or indirectly, with activity traces, showing how process data has thus far been conceptualized and where its treatment falls short. These three strands build on one another, moving from the tools creators work within, to the process those tools mediate, to the traces that process leaves behind.

2.1 Creativity Tools

Creativity support has been called one of the "Grand Challenges" of HCI research [266]. For much of the field's history, Creativity Support Tools (CSTs) have been framed as empowering technologies which enable users to be "more creative" [104, 105]. Shneiderman's seminal work positioned CSTs as systems that provide "low thresholds, high ceilings, and wide walls," which can support creative tasks such as search, collaboration, and iteration [255, 256]. However, Hsieh et. al argue that creativity systems encompass much more than just CSTs. They appear under a wide range of names and epistemic framings, including human–AI co-creative systems and domain-specific tools such as those for video content creation or game design [141] . The goal of such systems is to leverage computing to advance processes that result in creative *outputs*, rather than to support processes internal to the creator [141, 255]. For this reason, I use the broader term creativity tools, rather than creativity support tools, to capture this diverse body of systems contributions. The term also broadens the range of domains in view, encompassing not only systems built expressly to support creativity but the game engines, streaming software, and other general-purpose tools that creators appropriate to do their work.

A growing body of work shows that creativity tools are not simply neutral enablers of expression, but deeply entangled with the processes they are meant to support [267]. For example, CSTs impose a normative ground on creators, structuring how they think, act, and express themselves [176]. Mahler extends this critique to creativity tools more generally by calling for artist-driven tools that meet practitioners "in their world" rather than forcing mismatched workflows [194]. Critically, Mahler does not argue that tools should have *no* point of view on the creative process, even if such a thing were possible. Instead, the tools' implied point of view should align with practitioner knowledge, craft skills, and desired workflows. Surfacing these implicit assumptions can then help creators select tools that fit both their needs and their practices, rather than struggling with a tool whose point of view does not align with their own. Moreover, because tools are so entangled with process, they are also a productive site for intervention, and addressing these

critiques in future tools is a direct way to affect how the process unfolds.

The work of Li et al. and Mahler et al. are a direct call for the kind of research this thesis undertakes. Collectively, these studies point to a growing recognition that understanding creativity tools requires attending to how they intervene in and structure the creative process, not only to the artifacts they produce

The emergence and integration of generative AI amplifies this concern. Pang et al. note the "LLM-ification" of HCI, signaling the field's pivot toward AI-enabled systems that can ingest large volumes of creator data [218]. Recent studies of these tools reveal strong process-level effects, including the homogenization of creative outputs and influence of AI generated content on future content authorship [19, 241]. As these recent breakthroughs in AI have made it possible to act upon larger volumes and new forms of creator data [218, 220], they illustrate the urgency for a thorough examination of how creative processes are being surfaced, ignored, or overwritten, as well as for ways to reflect this knowledge back to creators.

Taken together, prior work reveals a landscape in which creativity tools exert significant influence on creative process, yet do not adequately reflect or adapt to creators' processes or points of view. This gap motivates the rest of this chapter. If creativity tools shape the creative process and can negatively constrain it, then a natural question is what the process itself produces and how tools might attend to it.

2.2 Process and Intermediate Outputs

Creativity research has long emphasized that creative work is not defined solely by its outcomes, but by the processes through which those outcomes emerge [124]. Classic models, such as Rhodes's 4P framework of creativity (people, process, product, and press) situate creative production as an interaction among individuals, their environments, and the evolving work they produce [235]. Plucker et al. describe creativity as "the interaction among aptitude, process, and environment by which an individual or group produces a perceptible product that is both novel and useful as defined within a social context" [224], whether or not the product is within a field culturally understood as creative. While these models include factors external to the individual, such as their environment, traditional creativity research has tended to conceptualize this process in primarily psychological terms, focusing on internal mechanisms rather than the concrete materials and representations creators produce as they work [103].

Critically, creative processes unfold *over time*, and externalizations accumulate into creative trajectories that extend beyond single projects [125, 211]. In writing, drafts and annotations map the path from early ideas to finished pieces. In video production, footage, rough cuts, and audio fragments are recombined or reused across projects. This temporal accumulation of materials supports reflection, backtracking, reinterpretation, and reuse, which are core activities in creative practice [267]. Provenance research both inside and out of HCI argues that the history of how something was assembled is essential for interpretation, accountability, and understanding [24, 26]. Creative work exhibits these same properties: intermediate materials form a temporal record of "where the creator has been," enabling them to situate current decisions within an ongoing trajectory that does not map cleanly to individual creative products.

In other words, the iterative process of creative practice produces a wide range of externalized

material. Seen from the perspective of creative *products*, it can seem ephemeral, and may not be used or even relevant in the final product. However, from the perspective of creative *practice*, this externalized material is highly functional. Regardless of whether it becomes part of a product, externalized material enables processes of reflection, reuse, adaptation, and meaning-making, both within individual projects and as part of the creator's longer-term development.

Creativity tools mediate how creators externalize ideas, organize materials, and navigate creative trajectories [105]. This means digital tools are not only capable of capturing these externalizations, they routinely generate and store vast amounts of process-relevant *data* through histories, snapshots, metadata, and user-created materials [259]. In principle, this positions creativity tools to support creators by making these temporal records available for reflection, adaptation, or guidance. Yet despite this potential, relatively little is known about how contemporary creativity tools capture, represent, or operationalize the rich intermediary materials of creative practice. The work surveyed in this section makes clear that intermediate outputs are relevant to creative process and accumulate into meaningful creative histories. I take up the question of how tools capture, represent, and operationalize them through creative activity traces, the process-related data that accumulates as creators work (Section 1.1), asking how that data is made visible to, and acted on by, systems, users, or both.

2.3 Activity Traces in HCI

Process-related data has long appeared in creativity research, but rarely under a consistent name or lens. One prominent tradition that has looked at process-related data is Goldschmidt's Linkography [117], an analytical tool to visualize relations among design moves to study convergence, divergence, or fixation [114, 117, 136, 158, 169]. Linkography relies on researchers to manually annotate design protocols to create linkographs, but recent work has sought to computationally create linkographs via Fuzzy Linkography [259]. Workflow graphs similarly encode action sequences and snapshots to compare strategies across designers [48, 169]. Collectively, this work shows that creator process data can be studied, but it has largely been treated as a post-hoc analytic artifact rather than as part of system behavior. Prompt-based generative AI studies similarly highlight temporal trace use, capturing users' iterative journeys through evolving text prompts[83, 193, 259].

Process data can also be *surfaced* within the features of creativity tools. For example, creative version control systems use traces as a form of material interaction for branching, diffing, and recombining, shaping collaboration and exploration [229, 267]. Additionally, mixed-initiative tools have appropriated traces for adaptive guidance, whether through predicting next steps in the interaction, gauging creative momentum, or deciding when and how to intervene [40, 77, 162]. Other systems use traces to generate suggestions by comparing a user's current state to prior work [51] , or to profile co-creative interactions [78]. Still others employ long-horizon traces to personalize system output, clustering style trajectories [15], profiling information-seeking behaviors [322], reifying past conversations with chatbots as manipulable objects [327], or learning creator preferences from user-created content [178].

Outside of creativity tools, research on the temporality of personal data echoes similar dynamics. Elsden et al.'s work on the Quantified Past shows how people engage in "rhetorical

data-work" to reinterpret their personal histories [90], while Gulotta et al.'s Curatorial Agents demonstrate how systems can reframe and surface temporal records in meaningful ways [125]. Speed and Oberlander introduce the ablative framework, which distinguishes designing from, with, and by data [263]. They argue that data is not a static entity, and as it becomes more "performative," its role shifts from a static resource that informs design to a live condition that co-structures interaction and can even automate aspects of system behavior [263]. Similarly, Odom et al leveraged temporal data in the form of music listening histories in the design of OLO Radio, advocating for the use of metadata as a design material [211]. Although these works do not focus on creativity tools and creator process data, they reinforce my view of treating process data as a rich temporal resource that can be leveraged as a design material, which motivates me to understand how process data can participate in, and sometimes drive, system behavior rather than serving only as a record of past use.

Looking across these scattered landscapes of trace usage highlights a gap. System designers and researchers in many subfields already appropriate process data, yet there has been no systematic effort to examine how such data is mobilized within creativity tools as part of their design. Most prior work either isolates traces within a single analytic tradition (e.g., linkography), uses them opportunistically for a specific feature (e.g., suggestions, personalization), or studies the temporal features of data independently of creative practice (e.g. Gulotta et al's Curatorial Agents). What is missing is a broader view of how traces circulate across creativity systems and what roles they play in shaping tool-mediated creative practice.

This gap persists even as interest in creative activity traces has grown quickly in the HCI research community. A recent CHI workshop on making sense of creative activity traces, which I helped organize, gathered researchers across visual art, writing, programming, music, and video creation to consolidate what had been a scattered set of approaches [163]. Current conversations around traces are concerned with how traces can be interpreted, which techniques best make sense of them, and how trace analysis might evaluate creativity tools or inform broader theories of the creative process. This thesis contributes to that conversation from a specific angle: I treat traces less as a form of post-hoc analysis for understanding creative behavior, and more as a *resource* that systems must be built around. Deliberately building tools around traces, so that they become trace-aware, is what remains comparatively underexplored.

In this dissertation, I leverage the term creative activity traces from Smith et al. [259], who used it to describe post-hoc analyses of interaction logs in systems such as Fuzzy Linkography. My use of the term differs in two important ways. First, I generalize beyond interaction logs to encompass the full range of process-related data generated as creators work, which accumulates into an evolving record over time (Section 1.1). Second, I focus on how traces are leveraged *within* tools to drive features and shape interaction, rather than how they are used externally for evaluation or visualization. Throughout the dissertation, I use the terms creative activity traces, activity traces, or simply traces interchangeably. The systematic review in Chapter 7 develops this orientation into a framework, surfacing how existing systems already embed process data into their behavior and establishing the foundation for treating traces as a design material.

2.4 Positioning this thesis

Across the projects in this thesis, I situate myself within creativity tools research and the problems creators face in their work. The systems in the first half of this thesis draw on that research, and were each designed to solve a problem particular to its domain. The data-driven solutions I built across mobile augmented reality, live streaming, and video content creation each taught me the value of the process data captured within these platforms, and reframed my understanding of how systems should treat accumulated material in the creative process. In looking across my work, this common thread became central to how I see the future of creativity tools research. Drawing on the bodies of work above, I argue that process data, in the form of creative activity traces, must be woven into how the creativity tools community approaches system design and development.

The chapters below chart the path I took to that argument. Chapters 3–6 present the data-driven systems where the value of process data first surfaced. I then synthesize and build on the disparate strands of HCI research on process, intermediate outputs, and activity traces, formalizing creative activity traces and trace-aware tools in Chapter 7 and exploring traces as a design material in Chapter 8.

Part I

Data-Driven Creativity Tools

Chapter 3
V-Light: Removing Computational Constraints to Support Creative Vision

This chapter introduces V-Light, a mobile AR architecture that leverages edge computing to enable technically ambitious, synchronous co-located multiplayer experiences on mobile devices. By developing an infrastructure which changes how data is processed and stored, V-Light demonstrates how infrastructural interventions can expand creative possibilities for game designers. Although V-Light predates the trace-oriented framing of this dissertation, it is where I first saw how the way a system handles data can determine what creators are able to design via their tools.

This work was previously published: Noor Hammad, Thomas Eiszler, Robert Gazda, John Cartmell, Erik Harpstead, and Jessica Hammer. 2023. V-Light: Leveraging Edge Computing For The Design of Mobile Augmented Reality Games. In Foundations of Digital Games 2023 (FDG 2023), April 12–14, 2023, Lisbon, Portugal. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3582437.3582456

3.1 Introduction

Mobile augmented reality (mAR) has been growing at a rapid pace, making AR experiences accessible anywhere and anytime. Because mAR-supporting devices travel with their users, mobile devices afford a portable AR experience with real-time environment interaction, location-based content, and multi-user spaces, all without dedicated AR hardware. In turn, these qualities allow designers to create mAR experiences that preserve social context, leverage real-time body tracking during movement, and support both physical and virtual co-presence [36].

mAR has great promise for game designers, as suggested by the success of *Ingress* and *Pokémon Go*. However, the technical limitations of mAR platforms make it difficult for game designers to take full advantage of the medium. Mobile devices introduce constraints around memory, processing power, battery life, and more [50, 70]. While these limitations could in principle be addressed by offloading processing to the cloud, a cloud-based solution introduces latency, which in turn eliminates game designs where real-time interaction is needed [89]. Game designers are therefore limited in their ability to work with computationally demanding features of mAR, such as synchronous multi-user interaction. Even major companies have had to limit their mAR experiences to off-screen, asynchronous multiplayer and low fidelity visuals [201].

We propose edge computing as a means of enabling computationally demanding, shared-world mAR games. Edge computing is an approach to cloud architecture that brings computation close to end user devices using an intermediate compute tier between device and cloud [243, 244]. It therefore allows mobile devices to access high computational power with low latency, in ways that neither on-device computation nor cloud computing can provide. While some interactive edge computing applications exist [113, 183], edge computing systems have not yet been applied to the exploration of the *design space* of mAR games.

In this paper, we demonstrate how we extended an existing edge computing system to support key features of mAR gaming. As a proof of concept, we share our game *V-Light*: an *edge-native* real-time two-player shooter for mobile devices. We show that our game supports the five key elements of synchronous mAR play [36], and that our system is able to support mAR interactions that cannot currently be implemented with either on-device or cloud computation. We share key design challenges and opportunities of working with mAR and edge computing, such as managing network stability and fostering player body motion.

Our work contributes a prototyping tool for game designers, so they can create games that current mAR systems could not otherwise support. We also contribute insights about how to work with mAR as a game design material. While we focus here on mAR games, our work has implications for the design of mAR beyond games, particularly where body motion, multi-party relationships, and portability are important.

3.2 Related Work

To frame *V-Light*'s capabilities and our contributions, we examine the literature in HCI and edge computing. First, we use mAR games as a lens to explore the current design challenges with mobile-native AR. Next, we discuss edge computing, its applications, and the key technologies that anchor *V-Light*. Finally, we outline how mAR fits within edge computing, and how *V-Light* compares to previous AR applications.

3.2.1 Challenges of mAR Games

Existing mAR games illustrate the challenges of designing highly interactive mAR systems, given limitations on interface, device power, and bandwidth. Mobile devices incorporate cameras, sensors and computer vision technologies that mAR can leverage, but the rendered AR environment is limited to the field of view of the device itself [70]. The limitations on graphical processing mean that current AR renders often lack complexity and have relatively low polygon counts, which in turn reduces the options for game art styles. Additionally, complex body detection is not supported; mAR games, for the most part, do not account for fine player movement, and instead incentivize off-screen movement using GPS tracking or gross rotational movement of the device that can be tracked using on-device accelerometers [207]. Our use of body-detection algorithms in *V-Light* led us to draw on research in body-aware gaming for our design [205]. Prior work in this space has relied on instrumenting spaces [29] or people's bodies [6]. Integrating body detection into *V-Light* allows body-centered play on a mobile device.

Another source of complexity in AR design is environment processing. In tabletop AR experiences, the device's camera is expected to be pointed at a flat surface such as a table. Movement is computed relative to that surface [36], which is assumed to remain static, providing an anchor point for augmentation. This reduces the computational load of AR processing and allows more system resources to be directed to gameplay. By comparison, room-scale AR expects that devices will be moving around a room (or a larger space), where a fixed anchor point is not possible. Additionally, room-scale mAR games have greater variability in the background and must

accommodate more potential objects in the scene [36]. This increases the AR computational demands for the game.

Adding multi-player interaction also increases game complexity. Single-player AR games such as AR Sports Basketball are the simplest, as they require no coordination among multiple players or devices [95]. When adding multi-player functionality, designers can limit the computational load of gameplay by designing away from real-time interactions and the use of shared game spaces. For example, *Pokémon Go* pushes player interaction to asynchronous, non-AR and/or off-screen parts of the game [214]. Players can leave their Pokemon behind to defend "gyms" (physical locations where virtual Pokemon can battle one another). An attacking player plays in real-time against the Pokemon's AI, not against the other player [151]. Synchronous multi-player is supported in the form of PvP battles, but does not use the device's AR features (e.g., camera, GPS, etc.). In addition to potential computational challenges, real-time multi-player games highlight the need for low-latency network capabilities, as player actions need to be propagated quickly to the other players in the game [89].

Some synchronous multi-player mAR games already exist, such as *Lightboard* [119], *Swift-shot* [120], and *Brick* [35]. However, these games demonstrate the tradeoffs described above. *Lightboard* and *Swiftshot* are both tabletop-scale games, while *Brick* chooses room-scale gameplay at the cost of simpler art and limited player interactions to reduce processing times [36]. Our work offers designers the opportunity to explore the synchronous multi-player mAR design space without needing to make such tradeoffs.

3.2.2 Edge Computing and Edge-Native Applications

Traditional cloud architecture consists of two tiers: devices such as smartphones, drones, or IoT sensors, and scalable computational resources in the cloud. Edge computing introduces a third tier that sits between the other two. Edge computing relies on the concept of the cloudlet, or "a trusted, resource-rich computer or cluster of computers that's well-connected to the Internet and available for use by nearby mobile devices" [243]. These cloudlets provide a high-bandwidth, low-latency connection to local devices, allowing for offloading of compute-heavy workloads to the edge. Cloudlets have network proximity to devices, but this does not imply physical proximity [245]. "On-premises" (On-prem) cloudlets are physically located close to the device, whereas cloudlets that are managed by telcos could reside at the base of cell towers or further towards the operator's central office. Positioning computation close to the device tier yields other benefits that extend beyond gaming. Applications may have privacy constraints that do not allow data to leave physical (e.g. country-specific) or logical (e.g. company-specific) boundaries. Cloudlets can ensure privacy preservation in these cases [76]. Edge computing also provides resiliency in poorly connected scenarios (e.g. disaster recovery or military outposts) where access to the wide-area network may be disrupted or entirely disconnected [87].

Research on offloading mixed-reality operations to the cloud or edge has been conducted in the past [145, 233, 254]. Toczé et. al, for example, created MR-Leo, a prototype that enhances video streams by offloading rendering to the edge [284], while Su et. al utilized a 5G edge architecture to enhance a mAR demo application that can recognize famous buildings in Paris [274]. There are also research efforts to improve edge computing technology for AR purposes by focusing on reducing latency [186], creating higher quality AR objects [181, 182], or scaling

to multi-user scenarios [228, 334]. The mAR industry has also taken notice of edge computing; *Pokémon Go* developer Niantic, along with global 5G carriers, announced that they are building an AR platform using 5G and edge technologies [152]. We build off of these established efforts to leverage edge computing, focusing on what we call edge-native AR applications for mobile devices.

Edge-native applications are those that could not exist without the edge. OpenRTiST is one such edge-native AR application. It utilizes a well-understood computer vision technique called style transfer to transform image frames in real-time that are sent from a device to a server at the edge with the transformed image then returned to the device for display. Empirical system tests have shown that attempting to run style transfer directly on a reasonably powered smartphone is impractical, even for images of fairly low resolution (240p) [113]. The deep neural networks needed to transform the images require computational resources beyond what the small integrated GPUs of today's smartphones can provide. At the other end of the spectrum, running the DNN (deep neural network) in the cloud results in increased latency and a degraded user experience that is immediately evident due to the interactive nature of the application [113].

In the domain of games, research has largely focused on how edge computing can improve current game systems. For example, the Edgegap platform offers low latency game hosting services by using edge computing [88]. Schmoll et al. demonstrated edge computing's capabilities with their multiplayer game that is playable on a VR headset and on a standard monitor [247]. Both of these examples are demonstrations of edge-enhanced games, as the core game functionality would still be possible without the use of edge computing. Jia et. al's work on multiplayer communication for mAR games deployed at the edge is another instance of research looking to optimize current games [156]. To our knowledge, no *edge-native games* yet exist, whether using mAR or otherwise. Our system seeks to fill this gap and to allow the rapid development and iteration of such games.

3.3 The V-Light System

In this section of the paper, we describe the interaction model used to underpin our system and game design decisions. Next, we share our game development pipeline, including how we extended existing work on edge computing to support synchronous gameplay. Finally, we describe our game *V-Light* and demonstrate that it is indeed an edge-native mAR game.

3.3.1 Interaction Model for Shared-World AR

To describe the features of *V-Light*, and to anchor our reflections on the process, we use Bhattacharyya et. al's model for designing synchronous colocated AR games [36]. Bhattacharyya et. al created this model to support the design of shared-world AR games, in which players can synchronously interact in the AR environment in real-time with others. The model consists of five categories of interaction in shared-world mAR:

1. **Single-player:** these interactions are between an individual player and augmented objects. Single-player interactions rely on the player's ability to utilize the augmented interface and can leverage known metaphors of mobile interaction, such as tap and hold or drag.

2. **Intrapersonal:** this form of interaction emerges from the player's experience in the augmented world, and refers to the individual player's internal thoughts, emotion, and cognition during gameplay.
3. **Multiplayer:** these interactions take place between multiple players and the augmented objects. Designers can also utilize multiplayer interaction to manipulate proximity and distance between players as they interact with the augmented world.
4. **Interpersonal:** this form of interaction refers to the experiences between players *that do not involve the augmented world*. This includes communication between players as well as physical contact.
5. **Environmental:** these interactions are between the players and the environment. This includes navigation, physical exertion, and collision avoidance.

Due to the rarity of synchronous shared-world mAR games, this model serves as the first formalization of best practices for game development in this area. To this end, this model is our foundation for the design and evaluation of *V-Light*. We show that the power of the edge has also allowed us to extend the design possibilities originally described in the model, which was built on a cooperative game. With *V-Light*, we created a *competitive* multiplayer mAR game with a greater emphasis on movement and use of the shared-world AR space. In using the shared-world interaction model, we are building on the foundation built by Bhattacharyya et al. to envision the future of shared-world mAR.

3.3.2 System Architecture

V-Light's basic architecture is a client-server model inspired by the OpenRTisT system [113]. Figure 3.1 shows a high-level overview of *V-Light*'s system design, including the current two-device implementation. In its current state, *V-Light* supports synchronous, multi-user mAR game experiences with two mobile client devices and a single edge server.

Our system architecture extends OpenRTisT in the following ways. First, OpenRTisT captures and manipulates only image data. We refer to this as the *image processing pipeline*. We create a second *gameplay pipeline* for player interactions such as shooting at their opponent, and integrate it with the image processing pipeline. Second, OpenRTisT assumes that all mobile devices are independent of one another. We synchronize interactions across two client devices, including maintaining game state and providing rapid feedback to players. Finally, we adjust the location of some computational processes (device vs. edge) to maximize the potential for real-time interaction.

Our *image processing pipeline* begins with image capture in Unity, which captures images on the client device at a rate of 50 FPS and a resolution of 426x240p. Images are compressed on the client device in the Gabriel client, then sent to the Gabriel server using a custom C# connection built around the Unity plugin BestHTTP/2 [33]. Gabriel [127] is an open-source offloading framework that can run on a cloudlet and adds negligible delay to the response times of interactive applications [56].

On the edge, images are processed by a computationally demanding subsystem. In the case of *V-Light* this subsystem is OpenPose [45] which is used to identify player body position (see game details below). The pose estimation is sent back to the client device in the form of key

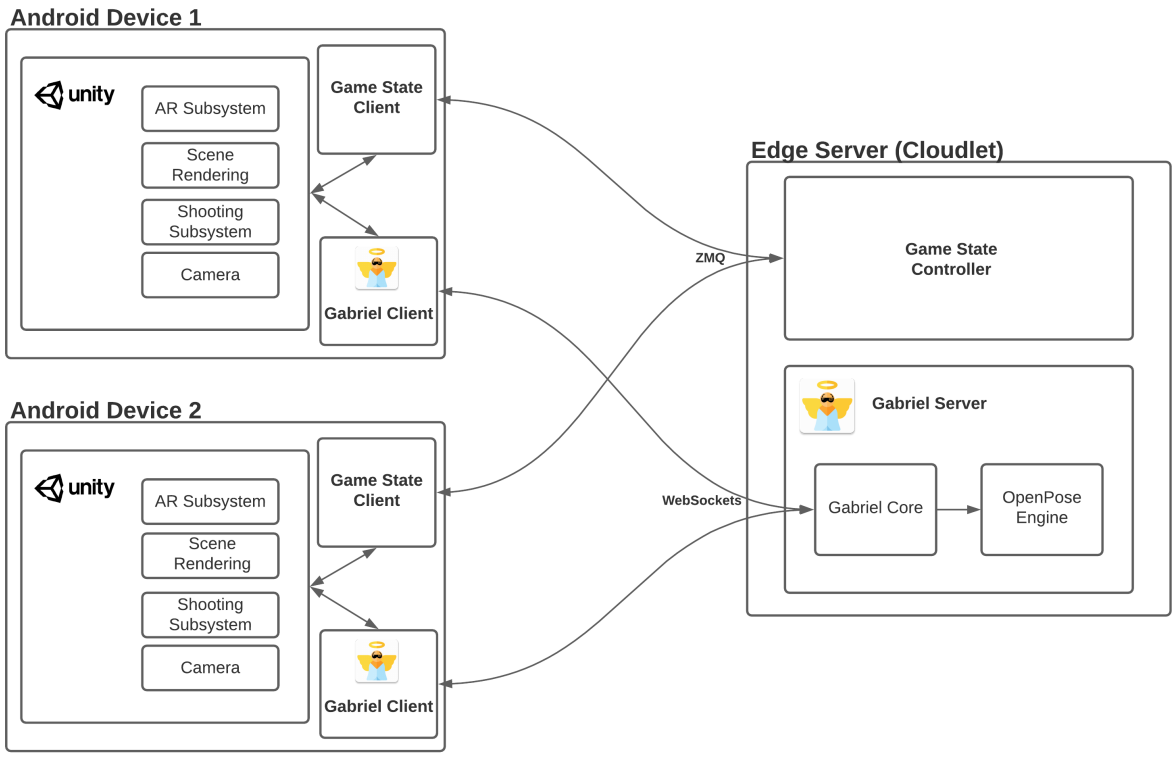


Figure 3.1: The V-Light tech stack

point coordinates. This process is different from OpenRTiST, which sends back a fully processed image. To save bandwidth and increase responsiveness, the original images are stored in an on-device image buffer and displayed once pose data is made available. Note that while *V-Light* integrates with OpenPose, other computationally demanding frameworks can also be swapped into the architecture.

Our *gameplay pipeline* captures gyroscopic and user interaction data in Unity. It sends this information to the game state controller client for encoding, then on to the server that synchronizes game state across players' devices. The controller tracks player shots, armor status, and health status, among other game variables. Using this data, it computes overall game state based on data from both clients, then sends the game state back to all client devices using our game state controller.

To maintain game synchronization between devices, and to broadcast game information regarding hit status, we created a game state controller using the embedded library ZMQ [332]. Similar to Gabriel, the ZMQ network server resides on the cloudlet, while the client resides on the mobile device. The game state server can synchronize data with an arbitrary number of client devices.



Figure 3.2: V-Light running on an Android device

Finally, our pipelines come together in a Unity rendered scene on the client device. Pose data from the image processing pipeline is used to locate the opponent in AR space. Combined with game state data (e.g. armor damage), it is then used to render on-body AR elements on the opponent (e.g., their armor see Figure 3.2). The Unity system also renders any room scale objects in the scene, updates game interface elements, and displays real-time feedback about game events, such as whether the player is currently being targeted.

Feature	Description	Interaction Type
Multiplayer shooter	Shooting is done by aiming with the mobile device and tapping/holding to shoot. Both players interact with virtual armor objects.	Single-player/multi-player
Collection of ammunition	Players must move around the room to find and collect ammunition.	Single-player/environmental
Competitive play	Players shoot the opponent's armor pieces to win the round. Timed rounds support competitive play.	Interpersonal/multi-player
Dodging and blocking	Players can avoid incoming shots by dodging or blocking.	Single-player/intrapersonal/ environmental

Table 3.1: Summary of features and interaction types of *V-Light*.

3.3.3 Game Design and Development

V-Light is a room-scale mAR shooting game for two co-located players. Each player takes the role of an astronaut attempting to capture resources for their home planet. They use an mAR interface to aim and shoot at their opponent, while simultaneously dodging and blocking their opponent's shots. Players must collect ammunition that is placed around the room in order to continue shooting. If one player eliminates the other within two minutes, they win the round and the resources. Otherwise, neither player receives resources. In both cases, the players can immediately start another round.

To explore the design space of edge-native mAR games, we needed to create a game that can currently only exist in edge-native form, while taking advantage of one or more of the key affordances of mobile play. In other words, we needed to design a game that was computationally demanding (beyond the capability of a mobile device), required low latency, and took advantage of co-located multiplayer AR features.

A competitive shooter like *V-Light* has a low tolerance for latency [67]. Players shoot at one another in real-time using an augmented interface. If the other player's body is present in the target area on their phone, then they score a hit. Rapid and accurate feedback is critical, as players need to know immediately whether they are being targeted and where their shots have hit.

For the computationally demanding component of the game, we chose OpenPose, a multi-person pose estimation system that can detect body, hand, face and foot key points from images [45]. OpenPose was a good candidate for edge-native gameplay because the computation of the key points is intensive, but the processed data is relatively lightweight and can be sent to client devices quickly. To integrate OpenPose with our core game loop, we created augmented reality armor. Each user has five pieces of armor: a torso piece, two arm pieces, and two leg pieces. OpenPose computes the location of the relevant key points (e.g. knees, elbows) and sends position information back to the client. The armor is then displayed on the mobile device in real-time, overlaid on the opposing player's body. Players can block shots with their armor, but each time a piece of armor is hit, it takes damage. When a player's armor has been completely destroyed, they lose the round. In other words, armor makes the OpenPose system a key piece of

the shooter gameplay.

Finally, our game needed to integrate one or more affordances of mAR specifically. We chose to design for player body motion within the space. To encourage players to move during gameplay, our game supports dodging and blocking. When a player is being shot at, they can move their body out of the target area (dodging) or move so that a piece of armor is in the way (blocking). To further support player movement, we added a room scale feature: ammunition supplies. Players can make a limited number of shots before running out of ammunition. To replenish their ammunition, they must search the space for virtual supplies. These ammunition containers are fixed relative to the physical space by using the device's gyroscope, and therefore players must move within the space to find them.

Taken together, these features of *V-Light* support the five categories of interaction outlined in [36]. The features and their categorization in the model are summarized in table 3.1, below.

3.4 Technical and Design Challenges

The development of *V-Light* as an edge-native mAR game yielded multiple technical and design challenges. The game underwent an iterative development process that was informed by our explorations of the chosen edge technologies and from using insights gathered from playtests. These playtests were conducted throughout the development process within the team and with external participants.

The challenges we discovered affected the afforded interactions, and by extension the play experience, across all five dimensions of the shared-world AR game model. The first challenge was *fostering player body motion*, which was encountered when designing for greater movement around the space with a mobile interface. The second challenge is *scaffolding real-time multi-player interaction*, which is critical for a successful multiplayer experience. Our final challenge was *preserving immersion*, which required particular attention to latency. Below, we detail these challenges, how we mitigated them, and the effect on play experience.

3.4.1 Fostering Player Body Motion

In consideration of the environmental interaction we wanted to support in *V-Light*, particularly with regards to player movement within the room, we made a number of design decisions to elicit roomscale movement.

3.4.1.1 Aiming

In our early designs for *V-Light*, we wanted to make sure that players could carefully target their shots at their opponent's armor. We hypothesized that players would have to move around the space to avoid these well-aimed missiles. Our early designs, therefore, allowed players to manually control the targeting reticle on the screen. By moving the reticle with a finger, players could home in on individual pieces of armor, such as their opponent's most damaged piece. The opponent would have to move fast to stay ahead of the damage.

In playtesting, we quickly found that finger-based aiming was counterproductive for these design goals. Players needed two hands to play the game: one to control the targeting reticle, and one to shoot. During the aiming process, players would slow or even stop moving around the space as they focused on small movements on their phone. Even when they were not actively aiming, they rarely let go of the phone with either hand, which constrained their movement and limited environmental interaction.

To address this issue, we fixed the reticle to the center of the AR screen, forcing players to aim with gross body movement (moving their arm or wrist) rather than fine body movement (moving the reticle with a finger). We observed that the new interface supported one-handed play, increasing the range of body motion available to players. Players also did not stop moving the rest of their bodies while aiming. Instead, the gross motor motion of aiming became a part of their body-use strategy.

3.4.1.2 Using Cover

As players began to move more freely around the space, we found that they were often only partially visible to the camera. In some cases, this strategy was used deliberately by players. Taking cover behind objects in the room protected them from being shot, at least until their opponent moved to see them.

Fortunately for us, the OpenPose system handles partially-visible bodies quite well. Despite the lack of z-axis data (see below), OpenPose could recognize bodies that were partly hidden behind other objects, or that were only partially in the camera view. Additionally, cover-seeking and cover-countering behavior encouraged players to maneuver around the space.

Unfortunately, player behavior around cover introduced a new challenge. To negate the benefits of cover, players would sometimes charge in close to their opponents. Not only did this make pose detection unreliable, it also meant that players could be trapped by their opponent behind an object or in the corner of the play space.

To address this issue, we added a "no go zone" between the players. Players' weapons would not work in this zone, which helped to enforce players staying far enough apart for both pose detection and safety. This decision also increased body movement in our playtests, as players at a greater distance have to move through a longer arc segment to counter an opponent's choice to use cover.

3.4.2 Scaffolding Real-Time Interaction

To create better single-player interaction with the mobile interface and multi-player interaction among players, we needed to ensure real-time interaction included effective in-game feedback mechanisms. An additional factor that affected real-time interaction was the exertion that came from the room-scale and competitive nature of *V-Light*.

3.4.2.1 Dodging and Blocking

We did not just want players to move around the space; we also wanted their in-game interactions within the space to be meaningful. We therefore wanted to design for real-time interactions

around being shot. *V-Light* allows players to dodge their opponent's energy blast, or to block it with an armored piece of their body. A direct hit to an un-armored body part means that player loses the game.

Because dodging and blocking are responses to the other player's behavior, we needed to give players advance notification that they were about to be hit. Originally we had planned to have bullets travel through the AR space toward the player. However, while OpenPose provides us with an efficient 2D pose estimation system, it does not provide z-axis data. Key points are defined on a two-dimensional plane. We therefore could not implement "bullets travel toward the player" with adequate reliability.

Instead, we created a charge-to-shoot interface. The attacking player needs to hold down the shoot button for a full second in order to generate enough energy for the blast. When they release the button, the shot fires. When a defending player is targeted during the charging process, they receive a "red alert" notification in their interface that gives them a chance to block or dodge. This design change affected single-player interaction due to the interface change, as well as multi-player and interpersonal interaction due to the time between shots. For example, multi-player interaction can be slower and more deliberate due to the charging time, which means players might use fewer bullets.

3.4.2.2 Resting

Early in our playtest process, we realized that we had inadvertently designed an aerobic exercise game. Players quickly became out of breath - and when they did so, they stopped talking. Silence meant the end of playful teasing, "trash talk," or light role-play around the game characters. Player speech also reinforced important game moments, such as calling out "I hit you!" Once play became silent, the interface had to carry extra interpersonal weight.

To address this, we introduced timed rounds. Players were given the role of battling astronauts, and their oxygen tanks would last for only two minutes. After each round of play, the players had a chance to rest and catch their breath before the next round began. We found that this sustained voice interactions between players for longer play sessions.

Adding rounds also fostered competition between players. First, players could win, lose, or draw individual rounds. Second, it gave players who were behind a chance to catch up, by having both players start each round with fresh armor and full ammunition stores. Finally, the rest breaks became moments when players would evaluate their progress, joke about who won the previous round, and insist on playing again to get another shot at winning.

3.4.3 Preserving Immersion

In order for players to enjoy *V-Light*, we need to ensure the play experience and immersion are not interrupted by the fidelity of the augmented objects or other technical difficulties.

3.4.3.1 Armor Design

During our game development process, we found that 3D armor was difficult to convincingly fit to a player's body with only 2D pose information. We reduced the complexity of the problem

by moving from armor with hard outlines to an ”energy field” design. Armor based on a particle system was more tolerant of momentary errors in tracking.

While the armor design change improved the situation, we still observed armor-fitting problems related to OpenPose's lack of z-axis. For example, when the player crosses one leg over the other, the system cannot always detect which leg is in front. Rather than add a general-purpose depth analysis algorithm and live with its computational costs, we captured a set of example movements where depth processing would be needed. We also made some strong assumptions based on our design. For example, armor rigidity means that distances between key points do not change over time. This in turn allowed us to create a custom armor rotation algorithm rather than solving the general-case problem.

3.4.3.2 Managing Network Stability

While our use of edge computing enables fast access to greater computing power, the stability of the wifi connection used by the Android phones to connect to that computing power is not always stable and not within our control. This would also be true for edge-based deployments outside of this work, as network stability is not always guaranteed. These network stability issues could have detrimental effects to the experience, and can introduce frustration in the interpersonal and intrapersonal dimensions. To address potential issues of local network stability we implemented a number of technical solutions.

First, we introduced a system for clients to automatically reestablish their connection to the cloudlet server in the case of network faults. To do this, clients constantly check the messages they receive from the game state controller for integrity. If the messages received appear abnormal and remain that way for a period of two seconds, the websocket is automatically terminated and reconnected to re-establish the state of the game. By doing so the robustness of the game is greatly enhanced.

Even though the game makes use of the performance benefits of an on-premises edge solution, we found that latency can still be a technical issue when players move too fast for the rendered armor to follow their bodies. To reduce the noticeable impacts of latency on the game experience in this situation, we implemented a buffering algorithm such that the client does not have to wait for the server to return pose data before sending a new image frame. This buffering approach allows the client to hold the image for a short period of time to wait for synchronization with the pose data returned from the server. Using this strategy, the fluency of the visualization is improved significantly and the latency becomes less noticeable.

3.5 Discussion

We set out to create a system that could support edge-native mAR games and provide a glimpse into the future of mAR game design. In doing so, we have learned more about the challenges in designing synchronous multiplayer mAR games at the room scale. Below, we elaborate on the design and technical implications emerging from our exploration.

3.5.1 Design Implications

3.5.1.1 Embodied Competition

We extend previous work on synchronous room-scale mAR games from a collaborative example [35] to a competitive one. Rather than having players collaborate in their goals, *V-Light* presents direct head-to-head competition between two players where the other player stands in the way of victory. This pivot to competitive play evolves the multi-player, interpersonal, and environmental interactions during play.

First, competitive play introduced *cycles of escalation*. When one player dodges, the other player chases. This can result in situations like players trapping one another, shoving physical objects, shoving each other, etc. Additionally, the interpersonal interactions are amplified by heckling and trash talk. While competitive play, by nature, can encourage these interactions, they may cause harm to the players. To address the environmental and interpersonal escalation, we can draw from physical games, such as sports, or even informal playground games (e.g. playground games) where rules are more flexible and are inferred from social norms [147].

One of the selling points of mAR is its portability; an mAR experience can be taken with the user anywhere their mobile device goes. This means a competitive shooter like *V-Light* can be initiated in any space. Games, be they formal physical games or informal ones, rely on the physical context to establish the rules of play. The rules for street hockey will vary slightly to those of ice hockey, for example. For the case of competitive roomscale mAR games, where the experience can be taken anywhere, how can we "inherit" existing norms around appropriate physical interaction during play from the chosen environment? In this case, the portability of mAR games presents a challenge, since we cannot simply rely on "arena rules" or informal "playground rules". Competitive mAR games will thus need to provide ways to determine when to invoke a common set of understandings around appropriate competitive behavior depending on the environment.

This can be done in several ways. First, designers could take an explicit rule-based approach that bans certain kinds of physicality. This could be problematic as there is a risk that the rigid rules might run orthogonal to the physical nature of the gameplay, incentivizing players to not follow those rules. Second, from a technical perspective, a no-go zone can be enforced, as was the case with *V-Light*. The same concept can be extended to detecting when players are moving too fast and acting accordingly to reduce their speed [94]. Third, a spatial perspective, an mAR game can be designed such that it is only playable in a particular kind of space, like a playground. This eliminates the need for multiple sets of norms and rules for each supported space, with the added benefit of potentially inheriting that space's pre-existing norms (e.g. a school hallway, where running is generally not allowed). Finally, designers could utilize interface design to their advantage. In our play tests, we saw that players' movement was limited when the interface required both hands to play. We evolved *V-Light* such that only one hand is needed to shoot; however, designers could instead intentionally limit player movement by requiring both hands on the device to play.

3.5.1.2 Augmented Bodies

One of our innovations with *V-Light* was the use of OpenPose to position augmented armor on the opposing player's body. This brings the players and augmentation closer together in the interaction dimensions; rather than interacting with other players and separate augmented objects in the environment, the players *themselves* are the augmented objects, and the body becomes the interface.

In our case, the computation to render augmented objects is camera-based. This means that the computational interface that you have access to as a player is not your own body, but rather the other player's body. To interact with this interface, the player will need to track their opponent with the camera to play the game. Compare this to an mAR game where the goal is to move around the space to collect more augmented coins than the opponent; the players' interactions with augmented objects is separate from the opponent, and they do not have to follow their body in space with the camera. This has additional implications for environment interaction; if players need to constantly track each other to play the game, how can the design ensure they do not bump into objects around the room that are outside of the camera's field of view? Moreover, how will the interaction with external augmented objects be affected? In *V-Light* we address some of these visibility issues by rendering ammunition packs within the field of view of the device so that players can quickly identify the location of the ammunition.

Finally, thinking forward, how does this augmented body interaction scale beyond a two-player experience? How might this approach extend not only to more players, but also observers who do not have devices? One possible scenario that builds on this concept is a team-based game scenario where players without devices serve as NPCs of sorts. For example, they might portray a horde of zombies that the players are trying to escape from.

3.5.1.3 Building a Time Machine

Edge computing is always going to live in the gap between mobile devices and cloud computation. At a larger scale, expectations for "low" latency and "high" computation are not constant. Today's high-end mobile devices, despite the hardware limitations we point out, have significant computational power - more than entire servers only a few years ago. In HCI literature, Hudson and Mankoff refer to investing in resources beyond what is currently feasible for the state of the art, with the hopes of envisioning the possibilities when it does become feasible, as *buying a time machine* [146]. Moore's law applies to smartphones, meaning that *V-Light* can conceivably run natively on a mobile device in the near future. Yet, when that time comes, and mobile devices can run pose detection like OpenPose, our edge computing system will still let us see what is beyond the state of the art.

Our time machine allows designers to pre-emptively explore complex designs and reflect on the implications. Within games, game designers might consider how edge-native experiences can allow innovation within established genres. For example, recent work on wearable technology in larp (live-action role-playing) focuses on social wearables to support physical connection [72, 253]. The authors found that the technology available was deeply woven into the game design process and social play. Edge-native mAR could provide larp designers with tools to create new gameplay experiences. For example, imagine a computationally intensive system that uses 3D

spatial audio data to predict social connections between players in a large (200+ person) larp. Such a system could be used to provide real-time data to players on who they are talking to, as the character may be expected to know things that the player does not. Connecting the system to a conversational agent could help struggling players figure out what to do next, while game organizers could view the social network map as a tool for planning their next event.

Our time machine also offers benefits for envisioning mAR futures in areas other than games. For example, work on novel in-car interfaces can use prototyping hardware embedded in the vehicle. However, prototyping vehicle-pedestrian interactions, for example with autonomous vehicles, is currently done in VR simulations [73, 74]. mAR prototyping tools based on *V-Light*'s architecture could support the rendering of simulations in live as opposed to virtual environments. In turn, this might help autonomous car researchers better support the experiences of pedestrians.

At the same time, our time machine can help catch users with older devices up to the present. Many global users are still using feature phones, while mobile devices for populations such as children and the elderly are often deliberately limited in their features and design. Access to bandwidth is, similarly, unevenly distributed. An edge-native experience that runs easily on a high-end device over 5G is unlikely to work at all on a much older phone using an unreliable network. However, edge computing can help that older phone run the same experiences that the high-end device/5G network has access to under ordinary circumstances.

3.5.2 Technical Implications

3.5.2.1 Managing Complexity

Our system relies on low round-trip latency between the device and the edge. That means that edge processing has to go fast, and the trip has to go fast.

There are different challenges for cases where you can add more edge compute resources (e.g. decreasing the processing time for the computationally complex element of the game), versus where it is easier to improve mobile bandwidth. However, designers typically will not know in advance where people are playing their mAR games - as they are able to travel with the player. We therefore need to choose computationally complex gameplay elements where there can be real-time optimization about what gets sent over the network versus processed at the edge. An example of this is our use of OpenPose, where we chose to send points rather than a processed image. While we implemented this optimization explicitly in *V-Light*, it could, in principle, be adaptive to network conditions. This would in turn shape the kinds of back-end complexity that can be added to mAR games with edge computing resources.

While computational complexity increases, it will become possible to render more augmented objects in the room. However, mobile devices' screen size, and thus field of view of the player, will likely not change greatly. If the complexity of the data processing increases, yet screen size stays constant, the player's ability to interpret the game's information may be compromised. Addressing this challenge will require innovations in data visualization and interface design to keep the experience playable.

3.5.2.2 Beyond Latency

Our system focuses on tradeoffs between latency and computing power in game design. However, there are also other benefits to mobile games that edge computing can provide.

Scalability is an additional benefit of utilizing edge computing. While edge nodes will never reach the level of scalability that is achieved by centralized cloud data centers, the increased computational power at the edge can enable many mobile devices to offload their onboard computation. Edge nodes are typically smaller in form factor and have lower power and cooling demands, making them more conducive to ephemeral deployments. One can imagine quickly provisioning a few edge nodes to support thousands of users at a gaming conference or a sporting event. It is thus possible to develop games like*V-Light* to support many players all sharing the GPU resources at the edge, while still maintaining the latency bounds needed for responsive gameplay.

Privacy is also a consideration for mobile games at the edge. Any data that is transmitted or stored will not leave the boundary of the edge. For games that process sensitive information about the user, for example by leveraging face recognition, the data will not be transmitted across the internet to a cloud data center; it only needs to traverse a short hop to the edge node. This may be particularly important for countries with strict privacy laws (such as GDPR in the EU) which do not allow some types of data to leave the country. Further, not only would locally sourced data not have to leave the edge environment but it could also be protected from external interactions associated with toxic player behavior in online games [161]. Edge developers or other edge operators could emphasize the privacy aspect of games deployed on their edge nodes as a way to foster community-driven game servers or services.

3.5.2.3 Service Selection and The Edge

From the user perspective, discovering and utilizing these services is another challenge. Users may wish to define criteria by which edge infrastructure is selected. These criteria may go beyond simply the cloudlet with the lowest latency, but include specific hardware (e.g. GPU) or service requirements (e.g pose estimation). For many users, simply finding the nearest (in terms of latency) server will be enough, but for advanced users, they may wish to describe their criteria in a more advanced way. This is analogous to the recommended video settings that many PC games come with, compared to an advanced user who may wish to tweak all the individual settings.

Developers will also face challenges in terms of deploying their games to edge environments. Edge platforms may or may not be owned by the developer. The resources available in each environment may vary greatly. Authentication/authorization and billing may be entirely different depending on who is operating the edge node. Additionally, developers will need to consider the challenges of deploying their games at the edge on heterogeneous edge infrastructure, be it a telco's infrastructure or a user's. Extending work on opportunistic device interactions may provide an avenue for addressing these limitations [79].

The resources of edge computing can be adapted to ensure that *V-Light* does not only work for those who already have the most resources. For example, a cloudlet might monitor local network conditions and serve a version of the software that supports higher packet loss, or take on more

processing for users with older devices. Cloudlet resource deployment techniques could even be used to increase the economic accessibility of applications built with architectures similar to *V-Light*, for example by having a user's search criteria for the "best" nearby cloudlet include application pricing.

3.6 Limitations and Future Work

Regarding technical limitations, *V-Light* is currently implemented for a single technical stack: Android devices, Unity as the game client, and OpenPose as the computationally demanding component. Our system architecture cleanly separates between client, messaging, and cloudlet operations, which should allow any of these components to be swapped out. For example, we could replace Android with iOS devices or OpenPose with a different computationally demanding library. Future work is needed to evaluate this system with other components, such as supporting iOS devices. Similarly, we tested only with wifi as our network component. Testing our game with other network infrastructure, such as 5G, might change the tradeoffs between on-device computation and network latency.

More broadly, our system currently makes some strong assumptions about the local network environment. We assume that anyone who deploys our game will also have access to a custom GPU-heavy edge server that they control. This assumption runs counter to the long-term vision of mAR as a fully portable experience. For a proof of concept that our system supports edge-native gaming, requiring server control is acceptable. In future work, we will have to consider strategies for registering with nearby cloudlets and discovering what games are supported "locally."

We also did not go as far as we could have with offloading computation to the edge. While our most computationally demanding operation (OpenPose) runs on the cloudlet, other computational demands remain on the client device, such as video rendering. Rendering on the client is well within the capabilities of the client devices that we used, and even with on-device rendering we demonstrate that *V-Light* provides edge-native gameplay. However, moving rendering and other on-client activities to a cloudlet would allow truly lightweight client experiences that could run on AR glasses or on legacy devices.

Another limitation of our system was choosing OpenPose as our body-detection system. The lack of z-axis data in the pose estimates affected our design choices. We chose OpenPose due to its efficiency and partial body-detection capabilities, at the expense of 3D pose estimation. Future iterations could consider alternative pose-estimation algorithms, such as BodySLAM [11], to continue to improve on pose detection. Additionally, we currently only use the edge infrastructure to perform pose estimation. Future work could explore offloading other functionality such as depth estimation [316] [34].

Finally, we must reckon with the environmental implications of our work. Deliberately seeking out a computationally intensive design space can be environmentally risky, as the case of blockchain currencies demonstrates - and the wider the adoption, the greater the negative impact on the environment. Mitigating these environmental risks will require commitments far beyond the scope of one research team. However, we can work with sustainability researchers to identify more-sustainable ways to explore this space, such as requiring cloudlets running our software to be powered by renewable energy.

Our long-term goal is for *V-Light* to enable a new generation of game (and other) mAR experiences, and more work is needed to truly explore this design space. Our game was created by a five-person student team, and was only internally tested. External playtests and user studies are required to further examine *V-Light*'s game experience. Following this, we will collaborate with game designers at a range of expertise levels, from novices to professional independent designers, to design and develop games built on *V-Light*'s architecture. Through this process, we will continue our use of the edge as a time machine and identify further design opportunities and challenges in the space.

3.7 Conclusion

Due to the limitations of mobile devices, current mAR experiences trade off on-device computation with remote-computing latency, which limits the design potential of mAR games. Our game, *V-Light*, leverages edge computing for its multi-player, room-scale mAR gameplay that has both high computational demands and low latency. This paper contributes a set of design insights for creating complex, roomscale mAR games. We believe that *V-Light* has paved the way for establishing best practices for roomscale, synchronous multiplayer mAR games. We hope to make the future afforded to us by the edge time machine a reality, with future iterations both in the domain of game design and beyond.

Within the arc of this dissertation, V-Light marks where my interest in data-driven creativity tools began. The project was framed around player data rather than creator data, but building it made clear that how data is captured, processed, and routed through a system shapes what creators are able to make. That lesson, that a tool's data capabilities can change the kind of work a creator can do, carried directly into my next project. Where V-Light built a pipeline around data about players' bodies, MARS turned to live data that creators themselves generate in the course of their work, asking what becomes possible when a streamer's live gameplay data is made available to their audience.

Chapter 4
MARS: Metadata-Augmented Real-Time Streaming for Audience Engagement

This chapter describes MARS (Metadata-Augmented Real-Time Streaming), a system that streams gameplay metadata to viewer interfaces in real time, enabling features like contextual overlays, asymmetric gameplay, and personalized viewer experiences. This chapter provides an overview of the MARS architecture as well as three example demos of MARS' capabilities. MARS highlights the potential of real-time creator data to support new forms of interaction and shift engagement responsibilities within the live streaming ecosystem.

This work was originally published: Noor Hammad, Erik Harpstead, and Jessica Hammer. 2023. The View from MARS: Empowering Game Stream Viewers with Metadata Augmented Real-time Streaming. In The*36th Annual ACM Symposium on User Interface Software and Technology* (UIST '23), October 29–November 01, 2023, San Francisco, CA, USA. ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3586183.3606753

4.1 Introduction

Live streaming platforms, particularly Twitch and YouTube Live [223], have seen an exponential growth in popularity in recent years. These platforms allow streamers to share content with viewers, creating a live experience where viewers and streamers can react and engage with the same content at the same time [252]. The growth of live streaming has seen it evolve and mature from niche entertainment into a medium with millions of daily viewers [5], streamers who are employed full-time via their streaming income [317], and a diversity of content that includes informal learning around topics such as art creation or programming [93, 102]. While Twitch is now home to many different genres of streams, video game streaming remains the most popular form of streamed content.

Current streaming interfaces support live viewer interaction with the streamer, e.g. through text chat. However, viewer interactions with content from the game being streamed are limited to watching the streamer's video. Anything the viewer wants to happen - such as turning on an accessibility feature or seeing a different part of the game by panning the game's camera - must go through the streamer. Both streamers and viewers appreciate interaction [171, 175], but viewers cannot count on getting the streamer's attention, particularly in larger streams [100]. Additionally, different viewers may want different things; one viewer might want to learn how to play the game while another wants a recap on the action that they missed, and the streamer may not be able to accommodate both. Finally, without the ability to indicate a specific referent (e.g. that game unit), viewers struggle to communicate what they want in text [116].

Previous systems have focused on aggregating viewer engagement and interactions with the streamer, rather than empowering individual viewers to independently augment their experience. For example, *Helpstone* is a streaming extension for the game *Hearthstone* that allows viewers to vote on what strategy they think an opponent should use, to draw lines on the screen that the

streamer can see, and to up-vote and down-vote comments for the streamer's attention [172]. The audience participation game (APG) *God of Socks and Spoons* gives an individualized role to up to ten viewers. Each viewer can independently choose which resource to produce for the streamer, but their participation is focused on increasing engagement with the streamer instead of giving them the capacity to control aspects of their experience without the streamer involved [116]. In both cases, the systems point to the value of *game-aware* interfaces for viewers, which, unlike typical *game-agnostic* interfaces, allow viewers to interact with game objects using video overlays in real time.

We build on the concept of game-aware streaming interfaces to empower game viewers with our system MARS: Metadata Augmented Real-Time Streaming. MARS is a system that enables any Unity game's real-time information to be rendered on a viewer's Twitch interface. We designed MARS such that a game developer can customize what metadata to send to a live stream, and a front-end developer can design and implement a game-aware overlay interface that augments the stream video. We use a thin viewer client, replay data, and screen coordinate mapping to render the overlays without needing to know anything about the game engine or screen layout. Game developers, fan communities, or researchers can create their own game-aware interfaces for any MARS-enabled game, and viewers can use the interface in whichever way they prefer without relying on the streamer's gameplay.

To accomplish this, our technical contribution addresses the following challenges:

- Tracking and transmitting relevant moment-to-moment metadata from games
- Spatially and temporally orienting transmitted metadata in terms of the streamed screen space video
- Exposing metadata in a clear, non-proprietary, and accessible way for use by a range of front-end developers.

We have identified three use cases for the MARS system that showcase different ways the system can empower users. First, a MARS interface can provide contextual information, such as allowing the viewer to click on a game unit and get more information about it. If a viewer is watching an unfamiliar game, they are now less dependent on the streamer to understand what is happening. Second, a MARS interface can add accessibility features, such as enlarging the game text, or translating it into a different language. Viewers with access needs can have those needs met without the streamer's intervention - even if the base game does not provide the accessibility feature in question. Finally, a MARS interface can support novel stream-based game designs, for example by providing viewers with information that is hidden from the streamer.

In this paper, we describe our development of the MARS system and identify its set of core technical features needed for the system's success. We then illustrate the system's capacities using three different game prototypes: a tower defense game that allows viewers to retrieve contextual information; a visual novel with added viewer-side accessibility features; and a puzzle game that relies on asymmetric information between viewers and the streamer. Taken together, our prototypes provide proof of concept that our system meets its technical goals, as well as demonstrating a range of possibilities for viewer empowerment using the system. We conclude with a discussion of additional designs that are made possible by this system. We thus contribute the following:

1. A scalable system architecture that supports metadata gathering from any Unity game
2. An approach that localizes the metadata to the correct screen space and accounts for stream delay
3. A platform that supports the development of game-aware overlays using any modern web framework
4. Three system prototypes that demonstrate the system's technical capabilities

4.2 Related Work

Live streaming platforms such Twitch and YouTube Live enable streamers to broadcast their content to a live audience. Across platforms, the live streaming interface setup that has become familiar consists of a camera view of the streamer, which is overlaid on a video of the streamer's content, and a chat panel where viewers interact with themselves and the streamer [223]. In the following, we review literature on augmenting this interface, and related literature from the field of live streaming.

4.2.1 Augmented Streamer Interfaces

4.2.1.1 Chat Augmentations

Stream chat provides an opportunity for streamers and viewers to connect in real-time, and both parties value that opportunity [64]. However, in large streams, high chat volume can make the chat effectively unusable, as messages scroll by at high speed [101]. Conversation Circles [202] uses "neighborhoods" to limit what chats the viewers sees, thus helping viewers communicate and make sense of the chat in streams with many viewers. When installed in a streaming channel, it improves the chat in real-time. TwitchViz [217], on the other hand, is a visualization tool designed to help surface relationships between chat behavior and moments of gameplay. Streamers, game designers, and other stakeholders can use TwitchViz to reflect on what moments of gameplay produced the type of chat interactions they desire.

Chat interfaces can be used for novel interactions. For example, early audience participation games (APGs) use the chat channel to participate in the game alongside the streamer. This began with viewers simply giving instructions to the streamer, but the technology for this has matured. "Twitch Plays"-games allow a crowd of viewers to influence a game by automatically aggregating their input in chat and connecting that to game behavior [227]. However, all commands are given in text. Further work in APGs build on this by creating arbitrary HTML interfaces that viewers can manipulate, which then encode viewer interactions in a special hidden chat channel that is connected to a Unity back-end [116].

Multiple studies have found that viewers switching back and forth between the video interface and the chat while simultaneously attempting to manage interactions is a challenging experience. In the context of APGs, viewers needed strong direction for where to put their attention at any given time, and game designers needed to work around this conflict [116]. Another study found that viewers explicitly requested supplementary interface elements to be overlaid on video because they found switching their attention distracting [171].

Table 4.1: Summary of Relevant Systems

System	Game-Aware	Real-time Augmentation	Overlaid on Video	Requires Additional Installation By Streamer
Twitch Plays Pokemon	N	N	N	Y
TwitchViz	N	N	N	N
Glickman et al APG	Y	Y	N	N
HelpStone	Y	Y	Y	Y
Conversation Chat Circles	N	Y	N	N
StreamSketch	N	Y	Y	Y
MARS	Y	Y	Y	N

4.2.1.2 Video Overlay Augmentations

Twitch extensions are a new addition to the Twitch ecosystem [128] and enable the rendering of overlays onto the interface using modern web frameworks. Twitch has few categories of extensions for viewer engagement, such as specific game information supports, general tools for scheduling and countdowns, and streamer tools such polling and voting. They are overlaid on the stream or around the stream. The streamer selects and installs them, and some of them are also viewer-interactive. For example, there are closed caption extensions that provide closed captioning for the streamer's voice. The streamer has to install it - but once they do, the text appears for all viewers as an overlay at the bottom of the stream. In many cases, the viewer can interact by opening and closing the closed caption panel. Twitch extensions are a core part of the MARS architecture, yet previous implementations of the extensions technology are usually in the form of polls and donation links. Although there are a limited set game-specific extensions available, such as *Dead Cells* or *Hearthstone* that are integrated with the original game and provide some game-aware capabilities [264, 280].

Most stream overlays are *game-agnostic*. They provide a feature, e.g. polling or a donation tracker, that does not interact with game data. A few integrate data from the streamer's gameplay, which we consider to be game-aware. However, existing game-aware overlays (and most others) often require the streamer to install additional software on their computer, such as the Overwolf framwork [1]. Moreover, extensions that have game-aware capabilities such as *Dead Cells* are few and far between, and are typically only developed for the most popular games and eSports.

Another approach is to allow viewers to draw on top of the stream video. *StreamSketch* is focused on allowing viewers to add open-ended sketches, similar to a whiteboard interface [191]. The system *StreamWiki* was created to support information archiving once a stream is over, and allows viewers to add information that can be viewed after the stream [189]. StreamSketch and StreamWiki offer novel interaction mechanisms for viewers; however, they were not built specifically for game streams and thus do not create game-aware interfaces.

Helpstone is an augmented viewer interface for the deck-building game *Hearthstone*. This system offers additional communication channels for the viewers, such as allowing them to draw lines on the screen to suggest moves to the streamer or voting on the strategy the streamer is

applying [172]. Helpstone is focused on fostering two-way communication between streamers and viewers, and as a game-aware system it was a major inspiration for the development of MARS. While Helpstone provides an initial example of the potential for a game-aware viewer experience, it was limited to turn-based games with fixed cameras, such as digital tabletop games. Additionally, the system required viewers to manually configure data synchronization in order to see data in real time. Our system addresses both of these shortcomings by enabling real-time metadata to be automatically synced to video in time and space, while adjusting for dynamic motion of game elements.

The above systems address the need for novel viewer interaction mechanisms in various ways. MARS integrates features from these systems into a single architecture to enable new functionality; MARS uses *video overlay* augmentations in *real-time*; MARS does not require the streamer to install any additional software; MARS is a *game-aware* system that can be generalized to many genres, including ones that rely on dynamic camera movement. Creating MARS to be a system for real-time, interactive game-aware overlays for arbitrary games required a substantial technical pivot from previous work in this space, which we will discuss in later sections of the paper.

4.2.2 Viewer Needs

A key affordance of live streaming is the "liveness" of the experience, where users can experience an event in real-time with shared social realities [69, 311]. In large-scale streams such as eSports tournaments, the liveness is akin to a digital stadium where the crowd expresses itself in the text chat [101, 129]. However, liveness also matters in smaller streams. Viewers are mutually aware of each other, and being able to experience a game together at the same time and the streamer's reactions can be rewarding for viewers [129, 237, 252]. Similarly, Diwanji et al found that Twitch viewers are able to create a feeling of digital co presence among themselves and the streamer [82]. Hamilton et al further define Twitch's live affordances as providing a "third place" where streamers and viewers can build relationships [129]. A "third place" refers to contexts outside of the home (the first place) and workplace (the second place), and Hamilton et al extend this by saying virtual third place is an online space where people can engage in social interaction and community building in a manner similar to traditional third places, such as coffee shops or community centers. Glickman et al argue that live streams can be a "distributed digital hearth", where participants gather around the stream for a shared activity [116].

At the same time, live streaming viewers have heterogeneous motivations and needs. In a study of spectators of *StarCraft 2*, Cheung et al defined nine viewer personas, ranging from pupils who seek to learn how to play to socializers who enjoy interacting with other spectators [58]. Viewers can be motivated by entertainment, information-seeking, socialization and community building, and/or tension release, and they may value different parts of the streaming experience [122, 139, 258]. For example, Schuck et al propose five different types of viewers: System Alterer, who prefers to influence the game being played; the Financial Sponsor, who is interested in financially supporting the streamer; the Content Observer, who wants to view the stream without using interactive elements; the Streamer-focused Observer, who watches because they are interested in the streamer; and the Social Player, who prefers to interact with the streamer and other viewers playfully [250]. Even when viewers share the same motivations and preferences,

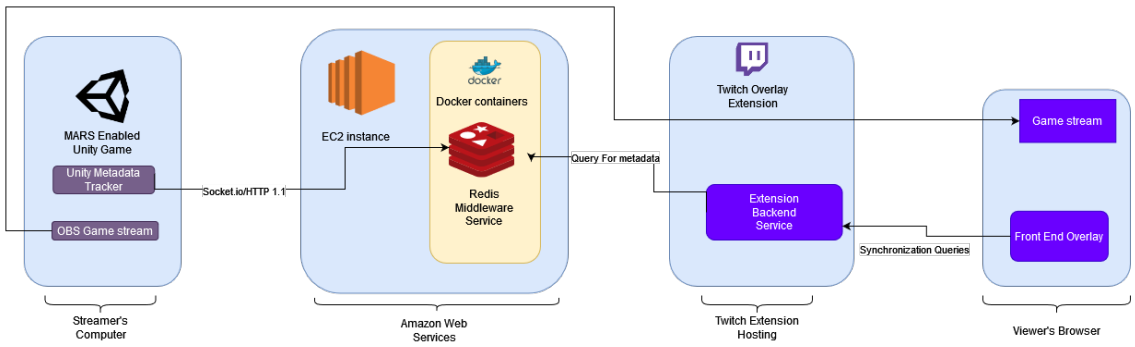


Figure 4.1: A schematic description of the MARS architecture

they can drop in and out of streams at any time [116, 325]. This means that individual viewers can also have different amounts of context for what is happening in the stream. However, an individual viewer's context might change based on circumstances, such as how long their current viewing session has lasted, or how much attention they're paying to the stream. The same viewer might have different access or control needs or wishes during different engagements. Not only do viewer needs vary from person to person, but can change for the same user. Despite these differences in motivations, viewers generally prefer the interactive elements of a live stream interface even if they choose to passively view the live stream [171, 175]

Taken together, these two bodies of literature suggest that viewers need support for both shared and personalized experiences. Most existing work focuses on improving the shared experience. Some of that is explicit, such as systems that support improvisational theater in the streaming context [192]. Some are implicit in the focus on systems that the streamer controls that are the same for all viewers, such as the design of Twitch extensions. Fortunately, personalization does not have to be at odds with having a shared stream experience, as our demos illustrate below.

4.3 The MARS System

MARS is a live streaming system that enables the creation of metadata augmented viewer-side streaming interfaces using real-time data. Below, we first describe the overall system architecture of MARS. We then give a brief overview of the metadata structure used throughout the system. Finally, we describe each of the components in detail and highlight the novel affordances that they enable.

Throughout the system individual frames are referenced by unique ids. A start frame is used to provide reference to initial settings for a given streaming session including the start time of the current game in the streamer's frame of reference, screen rendering dimensions, and the broadcast key and tween frame rates. Individual frames of metadata are stored with sequential ids. Finally, a latestFrame provides a description of the most up to date state of the game along with a frame id for the corresponding individual frame.

4.3.1 System Overview

The MARS System is composed of three main components (See Figure 4.1 for a schematic description):

1. **Unity Metadata Tracker:** A set of tools for the Unity game engine, which provides a system for tracking and serializing relevant GameObjects into frames of metadata that are sent to the Redis Middleware at definable frame rates.
2. **Redis Middleware Service:** A service hosted on Amazon Web Services (AWS) that receives and indexes metadata by frame to allow for asynchronous access by viewers with different latencies relative to the streamer.
3. **Twitch Overlay Extension:** A front end Twitch Extension that retrieves metadata from the Redis Middleware and renders it to Twitch viewers in sync with the live streamed game and localized to viewer screen space.

The MARS system architecture enables any number of Twitch streamers to play a MARS enabled Unity game on their channel while allowing any number of viewers to engage in the stream via a real-time metadata augmented interface. To start a MARS experience, a game developer must first implement the Unity Metadata Tracker within their game. The tracker must be configured to the correct middleware URI that is hosted on AWS. Then, a Twitch streamer must install the extension to be used on their channel, as is standard practice with all Twitch extensions. They can then stream the game to their Twitch channel using conventional streaming technologies such as OBS [4]. For the metadata to be rendered on the viewer side, a front-end developer (who may or may not be the original game developer) must implement a Twitch Overlay Extension using our front-end extension code to render a desired viewer experience. The front end developer must also ensure the metadata is received from the correct middleware URI that corresponds with the game being streamed.

4.3.2 Metadata Structure

The metadata structure that MARS adheres to is consistently used throughout the system. The metadata is organized as a series of frames that encode relevant game state information for rendering. We draw on the animation concepts of ”key frames” and ”tween frames” in our implementation [3]; Key frames define the start and end of an animation sequence, while tween frames are the set of frames between two key frames that allow for the smooth transition between one key frame and the next. In our case, a given frame of data contains both a keyframe, which encodes game state information for all actively tracked objects in the current scene, and a collection of tweens, which encode state information for volatile objects that changed more frequently than a single keyframe. The rate at which keyframes and tweens are captured is customizable and may vary from game to game. In our demos, we used a keyframe rate of 1 frame per second and a tween frame rate of 24 frames per second.

Within a single frame, the metadata representation for a given object contains both a screenRect that provides the x and y position, width and height dimensions, and z depth from the camera of a box that encompasses the object on screen, as well as a customizable JSON blob that provides additional game-specific information about the object. In addition to the metadata

```
{
  "game_time": 3022,
  "frame": 3,
  "key": {
    "RedWalker": {
    "screenRect": {"x": 728, "y": 378,
                   "w": 65, "h": 65, "z": 9.9},
    "secret_name": "Rocky",
    "color": "Red",
    "state": "Walking",
    "next_waypoint": {"x": 1073.25, "y": 556.75,
                      "z": 10.0}}
  },
  "tweens": [
    {"dt": 334,
     "game_time": 3356,
     "RedWalker": {
       "screenRect": {"x": 748, "y": 387,
                      "w": 65, "h": 65, "z": 9.9}}
    },
    {"dt": 668,
     "game_time": 3690,
     "RedWalker": {
       "screenRect": {"x": 768, "y": 396,
                      "w": 65, "h": 65, "z": 9.9}}
    }
  ]
}
```

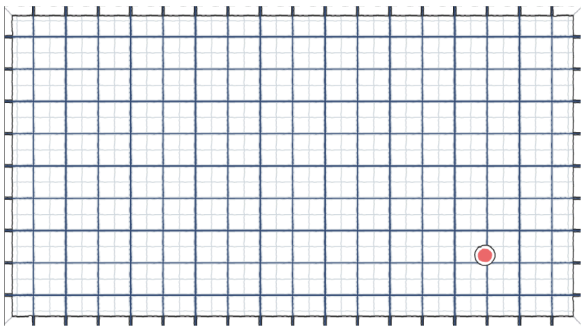



Figure 4.2: An example metadata frame and corresponding game state screenshot. This example involves a simple red character that walks around the screen following set waypoints. The walker has a randomly generated secretName, which is not visible to the player. Note that this example uses a key frame rate of 1 and a tween frame rate of 3.

of tracked objects, frames also contain the current game time (in seconds since the start of the game) and frame id. Figure 4.2 shows an example metadata frame taken from a debugging scene containing a single red character that walks along a set path of waypoints. Note that while this example uses a keyframe rate of 1 and a tween framerate of 3, only two tween frames are present as rounding in game times results in occasionally dropped tween frames.

The philosophy behind our metadata gives us a few benefits. First, we chose the metadata structure to tune the metadata's size by reducing the payload space of low volatility data. This structure also follows a "log-to-display" philosophy, which incorporates screen space information that reflects how the data will be used on the front end. This differs from existing systems' use of developer-provided game logs [172] which are designed to produce metrics for later analysis. While game logs are common practice in games, particularly among competitive games such as eSports, not all games that are streamed on Twitch provide public access to their game logs, nor are they created with the "log-to-display" intention. As such, in developing a generalizable system such as MARS, we could not assume that log data would be available nor adhere to the same structure across many game genres. Our system is thus neither aware of game mechanics nor makes game-specific assumptions about interactive zones; it derives this information directly from the metadata to be displayed on Twitch.

4.3.3 Unity Metadata Tracker

The Unity MetadataTracker is designed to harmonize with Unity's common GameObject / Component programming paradigm [291]. In order to track a given object a developer must register it to the MetaDataTracker. This can be done by extending a provided MetadataTrackable base class Component, which automates some aspects of registration and behaves similar to Unity's own MonoBehaviour structure, or adhering to an IMetadataTrackable interface, if the underlying game architecture requires more flexibility. Both options provide callback functions which allow implementers to specify custom metadata to be added to the JSON blob of a frame.

Once objects are registered, the MetaDataTracker martials registered objects into key and tween frames, at user definable rates. When a new keyframe is captured the previous keyframe and all tweenframes since are packaged and sent to the Redis Middleware Service over a web socket connection. This system makes it easy for developers to annotate and transmit relevant moment-to-moment metadata from games to viewers.

In addition to providing a system for tracking and recording metadata, the Unity package also provides tools for spatially orienting the transmitted metadata into viewer screen space. APIs are provided that calculate screen space rectangles relative to in-game cameras for a number of common Unity primitives such as MeshRenderers, Colliders, and Sprites, which correspond to viewer screen space used in the Twitch Overlay Extension. This means that all that is needed to localize the data in the extension is dividing the metadata screen rect values by the viewer's screen width and height. Using this information metadata can be localized on screen enabling novel viewer-side interactions such as hovering or clicking on elements to see additional information, or overlaying existing game elements with alternative visuals, which we demonstrate in our demos below.

4.3.4 Redis Middleware Service

The Redis Middleware Service is a Redis database hosted on an EC2 server provided by Amazon Web Services [17]. Along with providing a reliable and secure method to serve metadata to the Twitch Overlay Extension, Amazon EC2 servers give us the scalability that makes it possible to support any number of streamers or viewers. For example, if one Twitch streamer is playing a MARS-enabled game on their channel and experiences an unexpected influx of viewers, the EC2 server can scale to support the expanded audience.

For every EC2 instance that can be launched for a MARS-enabled stream, a database is hosted on that instance that receives metadata frames from the Unity Metadata Tracker and indexes them by frame id to be served by request to the Twitch Overlay Extension. The database that makes this possible is Redis [231], a database that provides fast and secure storage for key-value data. We chose Redis because of its speed and security, and due to the availability of APIs on both the Unity (.NET) and Twitch (Node.js) aspects of the system that allow us to interface with the Redis instance hosted on AWS.

4.3.5 Twitch Overlay Extension

The Twitch Overlay Extension makes use of Twitch's Extension tools, which allow third-party developers to augment the Twitch interface [128]. Twitch Extensions are front-end web applications that are hosted by Twitch. They are rendered within Twitch's interface and are composed of two main components: (1) and Extension Backend Service, which provides access to hosted assets and external data, and (2) a Front End Overlay, which renders visuals and drives viewer interactions. In our case, the Extension Backend Service acts as a pass-through to the Redis Middleware Service as a means for querying individual and collections of frames, while the Front End Overlay synchronizes data to the video stream and provides access to an up to date model of the game world.

One of the benefits of using Twitch-provided tools is that we can use stream information that Twitch offers, namely the viewer-side latency, to temporally synchronize the overlay with the game data automatically. This is done without the viewer needing to manually calibrate the system. In order to sync metadata to the live video stream the Twitch Overlay Extension makes use of two buffers of metadata, and initial buffer and a forward buffer. A fundamental assumption of this approach is that game streaming services like Twitch impose some amount of intentional broadcast delay (usually on the order several seconds or more). This means that the metadata being sent through the MARS system is guaranteed to be *ahead* of the content of the video stream from an individual viewer's frame of reference.

When first connecting to a stream the Twitch Overlay Extension queries the start and latest frames from the Redis Middleware Service. Using the streamer start time and frame rate settings from the start frame, and an estimate of stream delay provided by Twitches Extension tools [288], the Overlay Extension calculates how many metadata frames its current video stream is behind the current latest metadata frame. Using this estimation, the initial buffer is constructed by querying the Extension Backend Service for all metadata frames between the latest and estimated current frame. Once the system has been initially synced, a process is started which periodically queries the current latest frame from the Redis Middleware Service and adds it to the forward buffer, which is used once the initial buffer has been expended. This approach ensures that metadata remains temporally synchronized to the streamers video at all times.

With the initial and forward buffers maintained by the syncing algorithm, the Twitch Overlay Extension loops over the buffered data to update a persistent world model of the live game metadata. The world model is a hierarchical JSON object that contains all of the metadata from the most recent keyframe and any changes from ensuing tween frames. The world model is exposed by the extension code in an open and readable format to enable front end developers to use information to drive custom rendering or interactive experiences. The underlying Twitch Extension Overlay system is not opinionated about how rendering is performed and could in principle be used in conjunction with any number of front end frameworks such as React, or HTML5 game engines such as Phaser. In our own demos we manipulate SVG tags to enable lightweight interactions.



(a) *Tower Defense* game view that is streamed to viewers and played by the streamer



(b) The *Tower Defense* overlay that is visible to viewers, which reveals statistics about game units and information about upcoming enemies

Figure 4.3: The *Tower Defense* game, played by the streamer, and overlay, seen by the viewers

4.4 Demos

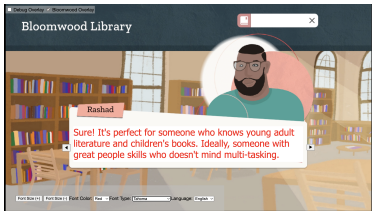
MARS is built because gameplay spectators have varying needs [58]. For instance, they have different levels of prior experience with the game being streamed, and thus recognize visual cues from the game differently. Moreover, viewers have different goals, such as improving their gameplay skills or socializing with the other viewers and streamer [129]. Viewers may also have different accessibility needs and require varying levels of support to fully enjoy the streaming experience. While streamers want to meet these needs of the viewers, it is already difficult for them to manage their audience while playing [101] and they have a limited ability to communicate non-verbally when it comes to addressing in-game objects [285, 321]. It is also impossible for the streamer to meet the access needs and conflicting goals of their individual viewers.

To showcase that MARS is able to meet these needs of viewers, we built three demonstration implementations that each use different Unity games and corresponding overlays using Twitch extensions. These demos showcase the key capabilities of MARS, namely enabling personalized information seeking for viewers, supporting accessibility needs, and affording novel live stream interactions using asymmetric game design. In addition, we chose the demos to highlight the complexity and breadth of games that are supported.

4.4.1 Contextual Information Cues - *Tower Defense*

Tower Defense is a 3D strategy game in which a player must stop waves of enemies from attacking their base. Enemies come in different classes, such as buggies, tanks and hovercopters, each with their own abilities and weaknesses. The player defends their base using towers and weapons which can be placed around the map. Each enemy that makes it through the defenses and into the base will result in a health deduction, and if enough enemies overcome the player's set up defenses, the game is over. If the player survives the specified number of waves, they will clear the level and be given a rating out of three stars. The fewer enemies that make it into the base, the better the star performance.

Tower defense games represent information rich environments where experienced players make use of complex strategies to get the most out of their tower placements [230], but less ex-



(a) *Bloomwood Stories* overlay is enabled, with the viewer adjusting the font type, size, and color.



(b) *Bloomwood Stories* overlay, shown here with translated in-game dialogue. The dialogue is now displayed in French.

Figure 4.4: *Bloomwood Stories* game view as seen by the streamer, and overlay features as enabled by MARS

perienced viewers may struggle to understand the details of a particular scene. From the viewers' point of view, a MARS extension can be enabled in their video that reveals contextual information about the units in the game. Viewers can hover over both towers and enemy units to reveal their current health, strengths, and weaknesses. Viewers can also see what enemies will be spawned in the next wave.

By giving viewers the ability to explore this additional information at any time, we can in turn enable personalized sensemaking. As the game progresses, the level is populated with many enemies and defense units which can overwhelm drop-in spectators and beginner viewers. Alternatively, viewers who are experts in the game can use the contextual information to provide hints and suggestions to the streamer. Without the overlay, viewers would have to rely on the streamer to hover over objects or to provide audio-visual commentary on the game. However, with MARS, viewers, regardless of their expertise or drop-in status can independently access information to support their own sensemaking.

Beyond the motivating use case of supporting viewer information needs, this demo was constructed to showcase technical capabilities. The Unity project is built using the tower defense template provided by Unity technologies [290], in order to show how our system can be integrated into a conventionally constructed Unity game. Additionally, the 3D art assets and dynamic camera serve to demonstrate the potential for our system to localize metadata to screen space while the player's perspective shifts. In other words, even in a 3D space, the front-end overlay will update in real time whenever the streamer moves the camera. Figure 4.3 shows the streamer's perspective of the game and overlay that is enabled on the viewer-side.

4.4.2 Accessibility Support - *Bloomwood Stories*

Bloomwood Stories: Block Party is an award-winning first-person visual novel where the player takes on the role of a community manager that completes quests [281]. To do this, the player engages in conversation with the non-playable community members and selects dialogue choices to affect the narrative. This narrative-driven game was designed to boost health literacy and self-efficacy among players.

Bloomwood Stories is a text-heavy game where conversing with community members and selecting dialogue options are core mechanics. This heavy reliance on text can be a barrier of entry for viewers with cognitive or vision impairments. *Bloomwood Stories* is also an English-language game, viewers who are not fluent in reading English may require support to understand the game's narrative. Accessibility barriers in video games have regularly excluded disabled individuals from enjoying the game to the fullest [31]. Fortunately, there has been an increasing number of mainstream digital games with robust accessibility features, such as *The Last of Us: Part 2* and *Celeste* [110, 150], however the widestream implementation of such features is not yet commonplace in all games. These accessibility barriers in games, particularly those related to vision impairments, translate into the live streaming viewing context as well. Consider the following example scenario:

In a live stream where the streamer is playing *Bloomwood Stories*, a viewer with low vision joins the stream and is interested in experiencing *Bloomwood Stories*' engaging narrative. However, since *Bloomwood Stories* does not use audio to supplement the text, the viewer struggles to read the text due to the small font size. Additionally, while attempting to read the text, the viewer is interrupted by the streamer moving onto the next dialogue option, leaving the viewer behind in the narrative experience.

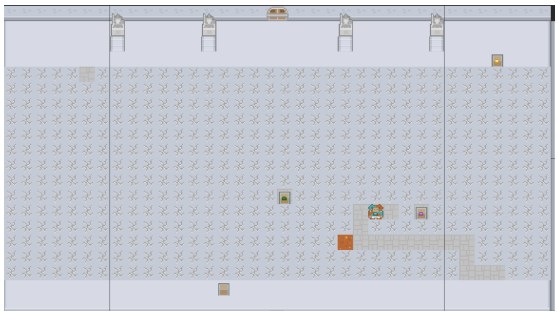
Joining in parallel in this stream is another viewer who is not a fluent English reader. This viewer is also a drop-in viewer that joined in the middle of the stream and has missed the earlier plot points in the game. This in turn means the viewer is not able to fully make sense of *Bloomwood Stories*' narrative, which reduces the viewer's level of enjoyment.

The MARS enabled *Bloomwood Stories* demo is built to support viewers such as these two personas. The *Bloomwood Stories* extension overlay on Twitch enables text manipulation features for the dialogue and text interfaces of the game. For example, viewers are able to modify the font type, size, and color to match their readability needs. To cognitively support viewers who want to read at their own speed, and to help drop-in viewers, the extension implements "next" and "previous" arrows that retrieve previous dialogue from the game. The above features were drawn from the cognitive and vision accessibility suggestions from the Game Accessibility Guidelines website from Able Gamers' Accessible Player Experience Design Patterns [2, 149]. Additionally, viewers can choose to have the displayed text translated into their own native language using the Google Translate API [121] to enable access to a larger global audience of speakers. Figure 4.4 demonstrates MARS in action and the implemented accessibility features.

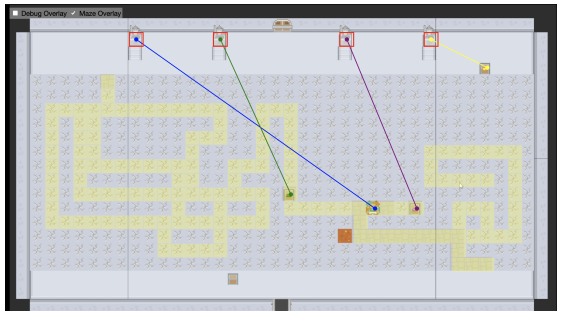
This demo was chosen to illustrate how multiple users could have personalized front-end experiences with the same game stream. From a technical standpoint, we wanted to explore how to integrate MARS into a previously released, complete game. We found that understanding the existing codebase, in particular to identify the objects that needed to be tracked, was the most challenging part of the implementation. This challenge is not specific to the MARS system, but does suggest that teams implementing MARS would benefit from prior familiarity with the game's codebase, even if they are not doing it in parallel with other development tasks.

4.4.3 Asymmetric Information - *Invisible Maze*

Invisible Maze is a 2D game where a player must escape a dungeon by placing four gems into their corresponding pedestals. To do this, the player must navigate through a room of treacherous



(a) *Invisible Maze* game view, with the player having partially navigated through the maze and approaching a gem



(b) *Invisible Maze* overlay that marks the safe tiles in yellow and draws lines to highlight where the gems should be places

Figure 4.5: The *Invisible Maze* game and overlay as enabled by MARS. With both sets of information, the streamer and players can successfully complete the level

tiles, where some are safe to walk on and others will break, falling into lava. Stepping onto a lava tile is a game over condition which causes the player to go back to the start of the maze and navigate back to where they left off. *Invisible Maze* is inspired by 2D dungeon crawlers and the early *Legend of Zelda* games.

In a Twitch live stream, the streamer playing Invisible Maze has no indication of which floor tiles are safe to walk on, and must rely on trial and error to determine the correct gem placements. Viewers, however, can be provided with information that the streamer does not possess. For example, they can see the correct path for the maze and where the gems must be placed to clear the room. Enabling the *Invisible Maze* MARS extension in their video player gives viewers access to this information (See figure 4.5). The tiles marked in yellow indicate the safe path, and the red lines indicate which pedestal the four gems belong to.

To help the streamer, viewers can share information in the text chat, and together, the streamer and their viewers can clear the dungeon successfully. For example, if a streamer is playing *Invisible Maze* and has two of their viewers (V1, V2) enable the extension, V1 can help the streamer get started in the maze by typing:

V1: “The way into the maze is six in from the right. You should be able to go up three tiles from there then turn left”

With the gems scattered across the room, V2 sees which gem is closest to the streamer:

V2: “I think after you turn right you should try to get that blue gem over there. That way you don’t have to backtrack.”

V1: “I agree, you’ll have to move seven tiles to the left and then it will be right above you, it goes on the first pedestal on the far size of the room.”

Following their suggestions, the streamer is able to pick up the blue gem and can proceed to place it in the pedestal that is across the maze. If they step onto a danger tile, they will drop the gem and will be teleported back to the start of the maze. However, the safe tiles they walked on already will be visible to them, thus allowing them to quickly overcome the areas that they figured out with their viewers.

Invisible Maze’ core interaction is based on the concept of asymmetric cooperative games

[134, 135]. In the definition of the conceptual framework for asymmetric gameplay, Harris describes this play experience as one where players are presented with "sharply contrasting aesthetic experiences in the same shared play space". In comparison to symmetric cooperative games, where all players have the same abilities and game mechanics available to them, asymmetric cooperative games are deliberately designed such that players have different information and abilities [134]. Limiting players' available actions means that no single player is able to succeed in the game on their own. In doing so, asymmetric cooperative play fosters greater social connectedness and interdependence amongst the players [133].

We extend this concept of asymmetric cooperative play to the live streaming context using MARS. In *Invisible Maze*, the streamer has more actions available to them than the viewers, however the viewers are given greater agency to influence the game because they have the knowledge needed to complement the streamer's actions. This interaction shifts some power to the viewer and creates interdependence between the streamer and viewers.

Invisible Maze serves as a probe into this new design space, and with MARS' flexibility, the game could have been designed differently. For example, rather than giving viewers access to the full solution to the maze puzzle, the overlay could instead display only some of the safe tiles in the streamer's immediate vicinity, which would require the streamer and viewers to piece together how to get from one safe tile to the next. Alternatively, metadata displayed to individual viewers could be systematically omitted such that no individual viewer could see the entire maze creating asymmetry not just between the streamer and viewers but within the viewers as well.

4.5 Discussion

In developing MARS, our aim was to explore new ways of empowering stream viewers. This process revealed lessons and broader implications of using this technology. Below, we elaborate on this, first discussing the relatively unexplored design space of supporting viewer needs in live streams, followed by discussing MARS' role in forming a distributed development ecosystem. We conclude by briefly discussing the hypothetical application of MARS beyond games.

4.5.1 From Capabilities To User Needs

Our demos were created to showcase key capacities of our system: supporting contextual information cues, enabling different accommodations for viewers available on the same stream, and providing information to the viewer without a stream referent.

In creating our demos, we were able to draw on known user needs of viewers to choose interesting and appropriate examples. For instance, we know from the literature that some viewers watch games they are unfamiliar with, so they can decide whether they want to play [58]. Augmenting the stream with contextual information supports novice players in understanding the game they are watching, even if the streamer is aiming their commentary at more knowledgeable players. Moreover, games without accessibility features add barriers of entry to disabled individuals [31].

The breadth of what is possible with MARS, and the potential to support the live stream experience that is unique to each viewer, contributes to our excitement for MARS application

beyond our three demos. For instance, Cheung et al's nine spectator personas includes "Curious" viewers, or those who are interested in the technical side of *Startcraft II*, and "Commentators" who want to critique gameplay and socialize by providing commentary [58]. Our demos do not directly address the needs of these two personas, however MARS' capabilities are such that the needs of such viewers can potentially be met within the same live stream setting. For instance, the "Curious" can have their needs met by enabling overlays that provide deeper insights into the frame rate and visual fidelity of the game as it runs. "Commentators" can be supported by drawing from the eSports spectatorship literature [49] to create overlays that support their analysis and their desire to provide game commentary in the chat.

As we continued developing MARS, we recognized that our work only begins to explore the possibilities for viewer empowerment. Furthermore, continuing to iterate MARS prototypes may reveal additional viewer needs not discussed in the literature. While our demos provide a glimpse into the potential of MARS, they were created to highlight MARS capabilities and demonstrate that these features are now made possible. We do not know the extent to which our demos *successfully* support sensemaking, accessibility needs, and asymmetric cooperative play. To this end, our demos act as design probes into new ways to support viewer empowerment. These probes spark new design avenues and future iterations that could better support sensemaking, accessibility, novel game genres, while also potentially identifying new viewer needs.

4.5.2 A Distributed Development Ecosystem

MARS is a system that we envisioned to improve the live stream viewing experience, yet as we developed the system to be more generalizable and applicable to any Unity games or front-end web frameworks, we found that our system design decisions affected other stakeholders. The needs of developers, both game and front-end, became central to MARS core features.

Similarly, by exposing metadata, we allow anyone to create their own overlays and have them work live on stream. All the streamer needs to do is play the game that shares metadata. Fans of the streamer and/or the game they are playing can thus customize their MARS overlays and share them with others. This is similar to the culture and process found in the long-standing video game modding communities, where fans become invested in the technical development of modifications and share those modifications online [260]. Alternatively, a different game company from the one that made the original game, such as a company that specializes in accessibility, can use MARS to build a custom viewer experience. Finally, viewers and streamers can also request metadata to be added on the Unity game side, and developers can respond to this request while knowing that adding more metadata will not break existing overlays but could enable new ones.

The envisioned dynamics described above reveal that MARS does not only enable viewer-side overlays, but also facilitates a larger *distributed development ecosystem*. Such an ecosystem is more than a place where viewers can enable overlays for a custom viewing experience, but a place where dedicated viewers, streamers, and players improve the play and viewing experiences by interacting with developers. This distributed development ecosystem that MARS can enable excites us for multiple reasons. First, the breadth of the community means the formulation of best practices for empowering viewers and meeting needs, as well as the development of novel designs that go far beyond what a small research team can offer. Second, an ecosystem such as this has the potential to foster deeper layers of connection between the various stakeholders. For

example, having dedicated stream viewers that build overlays for a specific streamer's community may strengthen the bonds of that community and provide added "social status" that some viewers value [129]. Finally, the distributed development ecosystem has implications for the discoverability of MARS-enabled games and overlays. Viewers can find whatever overlay they want, and being part of the ecosystem may help indie game developers bring attention to their game as well as having live stream communities build overlays if a small indie team is stretched too thin.

4.5.3 Streaming in non-game contexts

While MARS was built specifically for game streaming, our work poses implications for non-game streaming as well. Over the past decade, live stream platforms have matured and are now the hub for streams that cover various genres. For example, streamers build informal learning communities around topics such as art [102], programming [93], and cultural heritage education [190]. Streaming has also been adopted in formal learning contexts, especially in light of the COVID-19 pandemic [55].

When considering the similarity of the viewing interface across the various streaming genres, we argue that an exploration of how MARS can be extended to meet the needs of those viewers is an avenue of interest. The clear first step towards this extension beyond games is to consider other Unity experiences, such as virtual reality simulations. However, the underlying concepts for MARS metadata extraction may be applied to other technology contexts as well, such as an IDE or programming environment. Taking these concepts further, one avenue of interest that does not have a clear solution is creative streams where streamers are not streaming a digital practice but rather share a camera view of a physical artifact-related process, such as those found in art streams [102]. What these non-game applications reveal is that the metadata-to-server-to-viewer workflow may be applied beyond the architecture we created, and future exploration of what we can learn from MARS to apply to these contexts is warranted.

4.6 Limitations & Future Work

While we have demonstrated the potential for the MARS system to empower viewers of game streams to engage with streams in new ways, a number of promising directions for future work exist, both to address limitations of the current system as well as explore is potential for use and adoption by a broad audience of both viewers and game developers.

From a technical perspective some additional engineering enactments could be made to the system. For example, the current system for calculating screen rectangles does not make use of information to determine whether or not an object is occluded from the camera. This is not a fundamental issue with the structure of the broader architecture, and Unity provides its own set of tools to facilitate occlusion culling [292], however developing a suitably general solution for occluded metadata will require additional engineering. Similarly, the screen space position logic assumes that a game is being broadcast in full screen from the perspective of a single in-game camera at a time. Allowing for relative position from multiple simultaneous viewpoints should be possible but would require expansions to the current metadata protocol.

While some technical limitations are a matter of additional engineering efforts, other may prove more fundamental to the approach in the long term. For example, the nature of viewer enhancements being based on *overlays* imposes a constraint that an manipulations to viewable information must be additive as they do not have access to manipulate the original rendering the game image. For instance, consider a hypothetical extension intending to allow a viewer to remove unwanted imagery such as excessive gore or sources of phobias. As a MARS extension the only means to provide for such content control would be to cover it with some opaque shape, however, this would occlude any relevant aspects of the scene that may happen to occur in the same visual space. An intervention done within the game itself could simply not render the unwanted visuals [112] leaving the rest of the scene intact but such an intervention would be difficult to convey in metadata alone.

A final technical limitation that we foresee lies in the assumption of our syncing algorithm that the metadata will always be ahead of the streamed video. This assumption relies on there being an imposed broadcast delay between streamers and viewers, allowing the MARS system time to transmit data faster than the video. Continued advancements in low latency streaming will add limits on the metadata system's capacity to "win the race". However, Twitch does provide a toggle to streamers to increase imposed latency if they desire, which could reduce this limitation somewhat.

Outside of continued technical developments we also plan to explore future work in the usability and adaptability of MARS as a tool and of the individual demo experiences.

While we have endeavored to make both the Unity Metadata Tracker and Twitch Overlay Extension systems easily adaptable to developers with experience in their respective environments it remains an empirical question whether our current tooling is sufficient to allow adaption by others. Moreover, our system does rely on the game developers to implement the metadata tracking, and for front-end developers to design the corresponding overlay. We believe that the distributed developer ecosystem is a benefit of MARS, and plan to conduct developer studies to better understand the technical needs of both game developers and viewers to create their own MARS enabled experiences. To understand developer need we will explore game jams in which we provide our tools to a group of developers and analyze the prototype experiences they create. We will also work with designers to improve our existing APIs. To explore viewer needs we will instrument a single target game with the MARS system and challenge developers to create their own MARS enabled overlays that can interact with the base of metadata to see what a community driven creative process would look like.

We also plan to explore expanded user studies inspired by each of our individual demos. The *Tower Defense* demo shows the potential for individualized sense-making of game content but it remains unclear how best to design this kind of support to maximize viewer understanding. Additionally, the accessibility features shown in the *Bloomwood Stories* were primarily chosen to show a breadth of potential rather than target specific user driven needs. Future participatory design work could engage with real users who face accessibility needs to develop MARS enabled experiences that address the challenges of existing streaming interfaces. Finally, the *Invisible Maze* opens up an unexplored design of space of interdependent asymmetric games in the streaming ecosystem but also many interesting research questions. For example, which kinds of interdependent mechanics work well in the unique time delayed space of game streaming and what strategies to players invent to over come challenges?

4.7 Conclusion

The MARS system represents a significant step towards empowering viewers of live game streaming by providing them with independent control over their viewing experience. While current streaming platforms provide a means for viewer interaction with streamers, they do not enable direct interaction with game content without the streamer as an intermediary. Our system addresses this limitation by enabling game-aware overlay interfaces that augment the stream video and allows viewers to interact with game objects in real-time using moment-to-moment metadata.

Our system's technical contribution addresses several challenges, including annotating and transmitting relevant metadata, spatially orienting transmitted metadata, temporally synchronizing data to the streamer's video, and exposing metadata in a clear, non-proprietary, and accessible way for use by a range of front-end developers. Additionally, we have identified three use cases for the MARS system that showcase different ways the system can empower users: providing contextual information, adding accessibility features, and supporting novel stream-based game designs.

We have illustrated the system's capabilities using three different game prototypes and demonstrated that our system meets its technical goals. These prototypes showcase a range of possibilities for viewer empowerment using the system, and we believe that our system's scalability and flexibility will enable developers to create additional designs that are made possible by this system.

MARS extended the lesson of V-Light from infrastructure to using live creator data. The gameplay metadata MARS transmits was already being produced every time a streamer played, and making it available to viewers showed how data generated in the course of creative work can be put to new purposes. In this case, creator data was being used for reshaping the relationship between a streamer and their audience. Building the system also revealed that this data did not belong to streamers and viewers alone, which is what gave rise to the distributed development ecosystem. The next chapter investigates that ecosystem more explicitly, asking how the stakeholders implicated in data-driven game streaming understand and relate to the data that flows between them.

Chapter 5
MARS: Towards a Design Framework for Data-Driven Game Streaming

This chapter expands the scope of MARS from a single-system investigation into a broader study of the data-driven live-streaming ecosystem. Building on the prior chapter's focus on audience engagement, this work interrogates the relationship of streaming stakeholders with data as it moves across the ecosystem. These stakeholders include streamers, moderators, viewers, and game developers. Through a two-phase design study using the MARS architecture as a design probe, this project surfaces the expectations, tensions, and design goals of the streaming community, culminating in a framework for data-driven game streaming. In studying these streaming stakeholders, I found that data is never neutral, but a dynamic material that acts simultaneously as a resource and a constraint, depending on who engages with it. This realization helped me understand how stakeholders' relationships to data evolve through practice, informing my later work on how creative systems might more deliberately engage with creator data.

This work was originally published: Noor Hammad, Erik Harpstead, and Jessica Hammer. 2024. Towards a Design Framework for Data-Driven Game Streaming: A Multi-Stakeholder Approach. Proc. ACM Hum.-Comput. Interact. 8, CHI PLAY, Article 342 (October 2024), 28 pages. https://doi.org/10.1145/3677107

5.1 Introduction

Game live stream platforms such as Twitch [16] can be understood as data-driven ecosystems. Games, produced by game developers, use player interaction data to change the game state. Streamers share live video of themselves and their gameplay. Viewers and moderators engage the streamer by typing or sharing emojis in a simultaneous text chat channel. However, by default these stakeholders have very limited capacity to interact with one another, and participants' capabilities are primarily determined by their role in the ecosystem. Viewers, for example, are often restricted from modifying game state data.

Systems that leverage *data-driven game streaming* can offer novel interaction and communication mechanisms through the collection, propagation, and rendering of *live* data. Data-driven game streaming systems use data to power novel live streaming features, which can be collected from multiple sources, namely the game being streamed (game data), the streamer playing the game (streamer data), or the viewers and moderators watching the live stream (viewer data). Using these diverse sources of data, data-driven game streaming systems can enable features such as participatory mechanics for viewers, rendered overlays for streamers, or communication mechanisms for streamers, viewers, and moderators.

Some existing systems explore approaches to augmenting game streams with live data. Audience Participation Games, or APGs, offer the viewers the ability to affect the game being played by the streamer, often through the use of voting or aggregation mechanisms [116, 252]. Lessel et al, through their HelpStone system for the game *Hearthstone* [37], offer additional commu-

nication and interaction avenues for viewers via an overlay rendered onto the video stream of a game by utilizing game log data generated by the streamer's gameplay [173]. Similarly, Hammad et al's MARS (Metadata Augmented Real-Time Streaming) enables real-time rendering of metadata from Unity games [130] to support personalized viewing interfaces across many genres of games. Even outside of the HCI research community, systems such as Twitch's own extension tools are attempts to provide additional avenues for viewers and streamers to interact with each other and the game being played [128]. These systems vary in their approaches to data, how they define stakeholder capacities, and what types of features they support. However, they collectively demonstrate a hunger on the part of developers, streamers, viewers, researchers, and other stakeholders for richer game streaming experiences.

Although this hunger continues to drive system development, we do not yet have a framework to talk about the relationship between different data-driven game streaming systems, nor do we have a design language to talk about what kinds of experiences they make possible. To continue the exploration of data-driven game streaming experience established by these previous works, and to legitimize the data-driven game streaming paradigm, it is important to formalize the shared design space that unites all data-driven game streaming experiences.

Stakeholders who produce live stream data are also stakeholders in *how* that data is used to augment the streaming experience, and their buy-in is needed to make systems that work in practice. Game developers need to implement data-driven features in their games; streamers need to enable the data-driven functionalities on their channel, and leverage them to build relationships with their audience; streaming interface developers need to render incoming data; viewers need to engage with the new functionalities using those interfaces; and moderators need to maintain community relationships and guidelines in the presence of these new tools. Because these stakeholders are tightly coupled, we must work with all stakeholders *together* to understand what challenges they face, how their needs might be met, and what design opportunities exist.

Towards the goal of formulating the design space of data-driven game streaming, this paper shares the results of a two-phase design study conducted with target stakeholders in the distributed data-driven game streaming ecosystem, using an existing data-driven game streaming system to support participant ideation [130]. Stakeholders including streamers, developers, designers, and viewers served as our participants, with the goals of 1.) understanding their needs from a data-driven game streaming ecosystem, and 2.) utilizing their unique perspectives as our target stakeholders to understand how they think about the design space. The results from this study offered insight into how to build data-driven game streaming systems with the target users in mind. We propose a design framework that integrates the findings of our study with previous insights from other design space explorations of data-driven game streaming instances [116, 252, 271]. This framework provides a structured approach for designers to create, analyze, and improve data-driven game streaming designs; our framework asks designers to identify design objectives and target user (Experience), define abstract actions that users can take (Abstract Actions), describe how users can take those actions across available interaction affordances (User Interaction), and to determine the stakeholders' relationship to the data needed to power the design (Data Behavior). In summary, our work here contributes the following:

1. A set of design goals that stakeholders desire from data-driven game streaming systems
2. A series of general design considerations to maintain the technical, ethical, and social

integrity of these systems

3. A framework that incorporates the above findings to help generate, evaluate and improve data-driven game streaming designs

5.2 Related Work

Prior work shows that players already use their own game data to improve performance [236, 300]. However, game data has potential to play a greater role in the play experience if it is used as a "creative asset" [299] that serves both players and other stakeholders in a streaming context. In framing live streaming as a data-driven experience, we build on the concept of data as a creative asset that shapes the experience of all streaming stakeholders. To this end, we situate our study in the areas of data-driven streaming research and in research on streaming stakeholders.

5.2.1 Data-Driven Game Streaming Systems

Existing work on data-driven game streaming systems uses different data channels for different stakeholders and purposes. One common system design leverages or manipulates chat data. In particular, a series of systems have used chat data to enable sensemaking of conversations at scale. Conversation Chat Circles [202] is a tool to limit the number of chat messages the viewers sees, thus helping viewers communicate and make sense of the chat in streams with many viewers. The system dynamically adjusts the chat volume by using "neighborhoods" that group viewer chat data based on a number of criteria. Chung et al's VisPoll system serves a similar sensemaking purpose for a different audience, helping streamers make sense of visual viewer input data at scale [64]. This visual viewer interaction data is categorized using a visual input management framework (VIMF), which allows streamers to specify, aggregate, visualize, and make sense of this data. Another system called TwitchViz [217] uses chat data for visualization to help improve the streamer's understanding of the relationship between chat behavior and moments of gameplay on stream. Another use for real-time chat data is to mitigate toxicity, as is the case with Yen et al's StoryChat system [328]. This is a system built to encourage prosocial behavior on stream by generating graphical narratives from viewer chat messages, an approach which particularly supports moderators who would otherwise have to intervene.

Stream viewer chat data can also be used as a means of viewer participation in game streams. *Twitch Plays Pokemon*, for instance, aggregated chat data to drive the game mechanics in *Pokemon Red* [227]. *Choice Chamber*, another early audience participation game, allows viewers to choose what obstacles the streamer faces through chat [30]. Lessel et al's Crowdchess system used aggregated live stream chat data from viewers to allow them to play Chess against an AI on stream [174]. Seering et al also leveraged chat data in their first formal exploration of live streamed APGs [252]. This was followed by Glickman et al's toolkit that enabled the creation of arbitrary HTML interfaces for APGs, this time using the chat channel as a hidden means of encoding the viewers' interaction data for gameplay [116]. Both explorations of APGs revealed critical insights about the design of such experiences, paving the way for the exploration of the design space and its associated challenges. A new form of chat-based audience participation game known as Danmaku Participation Games (DPG) emerged on the live streaming platform

BiliBili which shows promise in innovating aggregation-based APG mechanics [307]. Wang et al shed light on this new form of stream-based gameplay, finding key design attributes that differentiate the independent viewer control of gameplay in DPGs from previous iterations of APGs. APGs such as the ones discussed here are a well-established form of data-driven game streaming system, however APGs are merely one type of of data-driven game streaming system which focus on offering viewers the ability to impact gameplay. We therefore argue that APGs are a subset of data-driven game streaming, and that the overall design space includes other types of experiences.

A second form of data-driven system involves incorporating multiple data sources, including game data, and rendering that data onto the video feed in an overlay format. The HelpStone system made use of game log data from the deck building game *Hearthstone* [37], combined with viewer data from an interactive overlay, to enhance streamer-viewer communication in real-time. Building upon this work, the MARS system (Metadata Augmented Real-time Streaming) formally introduced the concept of game-aware interfaces, or interfaces that allowed audiences to interact with game objects from a streamed game. MARS is a generalizable architecture which can be integrated into any Unity game, and makes use of Twitch's Extension tools to enable personalized data-driven interfaces for viewers. Twitch extensions are a recent addition to Twitch which enable overlays for viewers of streams [128]. Extensions can be developed using any modern web framework, and as of now there are hundreds of extensions available on the platform. In a data-driven context, extensions can offer many capacities for viewers and streamers. This includes game-specific overlays for popular games such as *Destiny 2*, which let viewers visualize game data in real-time. Extensions can also enable broader data-driven capabilities using an "Extension Backend Service", which allows an extension to communicate with an external data source. Nowadays, extensions power many functionalities for viewers, including iterations on the "Twitch Plays" experiences that kickstarted the APG phenomenon, such as the "Crowd Control" extension which can be used to activate audience participation in over one hundred games.

Finally, some systems integrate data sources outside the streaming platform itself. For example, Robinson et al incorporate real-time biometric data from the streamer to convey their emotional state to their viewers using a Heads Up Display [237].

We note that many interfaces are designed specifically to enhance the eSports spectatorship experience, given the popularity of the genre. For example, tools like Charleer et al.'s real-time dashboards visualize real-time data from the popular eSports titles *Counter Strike: Global Offensive* and *League of Legends* [49]. Stahlke et al also explored spectator interfaces for eSports, particularly focusing on the design of the spectator-player experience and identifying mechanics to support spectators [264]. Relatedly, there are a number of data-driven streaming systems which, while relevant, do not apply strictly to games [190, 192, 324, 325]. While eSports remain an integral part of the modern live streaming landscape, the unique qualities of eSports, such as the nature of viewer engagement being more akin to a "digital stadium" as opposed to a "digital hearth" [252] and the use of eSports casters to provide commentary, do not apply to the broader game streaming ecosystem. As such, we chose to focus our efforts on game streaming experiences where a streamer is playing a game on their channel for viewers to engage with. While we are focused on games *outside* the context of eSports, we found that the data-driven game streaming systems we are interested in rely on similar data sources as those eSports: chat

data, game log data, overlay data, and data sources from outside the core stream.

Many of the above papers identify design implications of their respective systems in the form of design goals and considerations [116, 202, 252]. Taken separately, these design insights lay the foundation for the design spaces for different types of data-driven game streaming systems. Taken together however, these systems lack a design language that unites the overall space of data-driven game streaming. Our goal then, is to move towards establishing that design language to pave the way for best practices in the field.

5.2.2 Streaming Stakeholders

Streaming stakeholders include streamers, viewers, moderators, game developers, and more [130]. These stakeholder groups have different needs and priorities, even as they work together to create engaging live-streamed game experiences.

A substantial body of work explores the needs of individual stakeholder groups, including how those needs might inform design. For example, Wohn et al conducted interviews with streamers to understand their audience management practices. Their findings yielded design recommendations to create streamer tools for audience management to support knowledge discovery and growth management [317]. Mallari et al interviewed game streamers to understand their data analytics needs [195]. They found that streamers have a range of information needs ranging from content, marketing, and community data which are not easily met by existing tools, and have demonstrated the importance of designing effective analytics interfaces for game streamers. In terms of viewers, Wang et al's exploration of the recent Danmaku Participation Game (DPG) phenomena involved semi-structured interviews with viewers on the BiliBili live streaming platform to understand their practices and experiences [307]. Also working with viewers, Lessel et al surveyed over 400 viewers to understand the elements they value in a live stream setting [175]. Their findings ranked 58 elements of a live streaming experience, with interactive components being the most valued by viewers.

While need-finding with streamers and viewers is relatively common, investigations of other stakeholders' needs are rare. For example, Li et al's study of indie game development communities on Twitch interviews game developers in a streaming context, revealing how members of the indie game developer community leveraged live streaming for a distributed software development setting that encouraged a democratic model of game development [177]. However, the paper effectively treats the developers as streamers and viewers, using live streaming as a means to build community, rather than focusing on their underlying game development needs. Similarly, the diversity of streaming moderator motivations and experiences is only now beginning to be understood in ways that can inform design [43, 251].

Although our focus is on game streaming, we note that studies of live streaming outside games often engage multiple stakeholder groups at the same time, and illuminate how their needs can be complementary. Interviews with streamers and viewers of intangible cultural heritage streaming found that streamers had altruistic intentions, and viewers were motivated to learn and engage with the arts [190]. Similar work was conducted with streamers, viewers, and moderators of educational streams to understand their current practices for knowledge sharing, with the findings being used to build a knowledge sharing system for viewers [189]. In a similar educational context, interviews were conducted with streamers and viewers in a classroom setting, where

instructors took on the role of streamers to continue teaching during the COVID-19 pandemic [54]. In the context of creative streams, streamers and viewers were interviewed and surveyed to identify how to best support the goals of streamers in their creative process [102].

While many of the papers discussed in the previous section worked with viewers and streamers for formative or evaluative purposes [173, 202, 217], streaming stakeholders are typically not directly involved in the design process. A key exception is Glickman et al's work with APGs [116], where researchers worked with game developers for ten months to create audience participation games. This process moved the design space of APGs forward by yielding six design challenges for APGs, such as latency and sharing the screen, along with worked examples by game professionals. Striner et al extended this work by engaging six student game developer teams, which resulted in three sensitizing concepts for APG design: agency, pacing, and community [271]. In working with developers, the above moved the field of APG design forward, underlying the importance of incorporating developer feedback in data-driven game streaming design.

Taken together, these studies involving stakeholders highlight the importance of understanding the data practices of individual stakeholder groups, such as streamers' needs around data analytics [195]; learning about synergies between groups so that designs can address multiple stakeholder needs simultaneously [190]; and engaging stakeholders in design, particularly game developers and others who are already designing systems for streaming, to leverage their insights and expertise [116]. We designed our study to directly engage stakeholders *together* to explore the potentials of data-driven game streaming systems, arguing that their perspectives and interdisciplinary collaboration through our study can help contribute to our understanding of their varying, and often conflicting needs. We thus work with streaming stakeholders to understand what features they want to see in these systems, what concerns they have with this technology, and how we can collectively move towards defining a shared language for the design of data-driven game streaming systems.

5.3 Ideation and Design Study

In order to better understand the design space for data-driven game streaming, we executed a multi-stakeholder study structured around the following research questions:

1. What features do streaming stakeholders want in a data-driven game stream system?
2. What challenges and concerns do stakeholders have when it comes to novel data-driven game streaming features?
3. How can stakeholders contribute to the design of data-driven game streaming systems?

To address these research questions, we designed a two-phase study where game developers, live streamers, viewers, moderators, designers, and researchers were invited to collaboratively envision the future of data-driven game stream experiences. In the first phase of our study (Phase One), participants brainstormed desired features (Design Goals) and ways to mitigate challenges (Design Considerations) for data-driven game streaming, generating *affinity diagrams* from their ideas [222]. The research team then reviewed Phase One data to synthesize themes and insights. We used our analysis to create prompts for the second phase of the study (Phase Two), a partici-

patory design workshop where participants envisioned sample data-driven game interfaces. Our analysis of the Phase Two data allowed us to see how stakeholders understand data being created, maintained, and rendered.

Below, we provide a detailed breakdown of our study, including the methodology and results of each of the study phases.

5.3.1 Participants

To ensure our study included the diverse perspectives within the data-driven game streaming ecosystem, we targeted a range of participants: game developers, front-end developers, live streamers, viewers, moderators, designers, and live stream researchers. More specifically, our inclusion criteria specified that participants be residents of the United States, be aged 18 or older, have a familiarity with the Twitch live stream platform, and have experience in one of the roles we defined.

We utilized multiple channels to reach potential participants, including social media platforms (X, Facebook, Reddit) and informal networks within the Twitch streaming community. We also recruited heavily through specialized Discord servers, namely servers for game developers (IGDA, Global Game Jam, Unity Developer Community), user researchers (IGDA Games User Research and UX SIG), Games Related Academic conferences (CHI PLAY, Foundations of Digital Games), and Twitch extension developers (TwitchDev). We did not utilize eSports community channels in our recruitment, as our scope was on other forms of game streaming. Interested parties first filled out an online screener survey, and selected participants were emailed the consent form and scheduling information.

In Phase One of our study, we recruited a total of 26 participants, who were distributed across six study sessions. For Phase 2, we retained 21 out of the initial 26 participants, distributed across four study sessions. While all of our participants were representative of the stakeholders in the distributed developer ecosystem, many of them identified as having multiple roles; for example, a participant who was a professional game developer also live streamed on Twitch as a hobby, while another participant was a professional streamer who also had experience as a moderator. Fourteen of our participants identified as male, eleven identified as female, and one participant identified as non-binary. In total, ten participants identified themselves as *streamers*, ranging from full-time professional streamers who stream at least three times a week with 50-200 viewers per stream, hobbyists who stream once or twice a month with 10-20 viewers per stream, to individuals who did professionally stream in the past but no longer do so. Nine participants identified themselves as *game developers*, four of whom are current graduate students in a game design program. A single participant identified themselves as a professional *UX designer*. Three participants identified themselves as *moderators*, all of whom had experience moderating multiple channels. Two participants were academic *researchers* interested in live streaming. Finally, six participants identified themselves as *viewers*, all of whom regularly watch Twitch streams on a weekly basis.

5.3.2 Phase One: Collaborative Ideation

Phase One of our study was primarily focused on addressing the first two research questions; What features streaming stakeholders want in a data-driven game stream? What challenges and concerns do stakeholders have when it comes to novel data-driven game streaming features? To accomplish this, we used affinity diagramming [222], a collaborative ideation and organization method which can be conducted with target stakeholders to generate information and categorize that information into related groups.

To help support participants' ideation, and to establish a shared meaning of what data-driven game streaming is capable of, Hammad et al's MARS system was used as an example model of data-driven game streaming. MARS was selected for the study for three reasons; it is a recent system that demonstrates the current cutting edge capabilities of data-driven game streaming; its architecture offers generalizeability to many genres of games; and it is designed specifically with a "distributed developer ecosystem" in mind [130].

Each affinity diagramming session consisted of 3-6 participants, and was conducted on Zoom and Figma. The placement of participants into sessions was largely decided based on the availability of the participants, however the research team ensured no session consisted of all participants who identified as the same stakeholder type. Sessions lasted approximately two hours, and were facilitated by a member of the research team. Two additional research team members supported the session lead by addressing technical issues as they occurred. To maintain participant privacy in the group setting, each participant was given a pseudonym as soon as they entered the Zoom waiting room. Participants introduced themselves using their pseudonym, and all study data was recorded using this identifier.

Next, the facilitator set the context of the session by describing the MARS system. The overview of MARS included a discussion of the system architecture as well as a video demo to show examples of the system's capabilities. We provided time for questions to ensure that all participants understood the concept of data-driven game streaming and how it might be instantiated in a system.

After context setting, we moved to a brainstorming phase. Participant brainstorming was broken into *individual brainstorming* followed by *collaborative brainstorming* [86]. This strategy maximizes the diversity of participant ideas, while providing shared context during the group discussion.

As individuals, participants were given fifteen minutes to generate ideas on sticky notes on Figma, in response to three guiding questions:

1. How do you envision MARS being used by streaming stakeholders such as viewers, streamers, moderators, and developers?
2. How can people benefit from this technology?
3. What features would you like to see in MARS supporting?

Participants were then instructed to look at and annotate all sticky notes on the board. If they liked an idea, they could drop a "kudos" sticker on it. If the note was not clear, participants could leave a question mark sticker. Facilitating natural discussions in a workshop among strangers was difficult when done remotely [99], and the kudos and question mark approach provided a segue for participants to start discussing with one another. Once participants were done annotating, the

facilitator read the notes marked with the question marks out loud, which gave the participants an opportunity to clarify and discuss their ideas with the whole group. At this point, participants were given five minutes to collectively answer one final guiding question:

1. What ethical, social, personal, or technical challenges can you foresee occurring when using this technology?

The final hour of the session focused on sorting and grouping the generated notes through affinity diagramming. Following standard affinity diagramming protocol, participants took turns placing one single note into an existing group; if the note did not fit within that group, participants created a new note grouping. Sorting and grouping of the notes was done based on the similarities or connections among the underlying ideas, as participants were regularly reminded not to group notes based on keywords. Throughout this grouping process, participants were engaging in discussions to negotiate the placement and refine the clusters.

When all notes were grouped, and the participants were satisfied with their groupings, the session concluded by generating theme sentences for each note cluster. These had to be short, descriptive sentences that captured the underlying relationship between a group of notes. To conclude the grouping activity, participants were instructed to take a final pass of the groups of notes and their associated themes to discuss any final changes.

5.3.3 Inter-Phase Analysis and Phase Two Preparation

In total, we conducted six affinity diagramming sessions in Phase One involving 26 participants, which generated approximately 370 notes in 58 groupings. Our next task was to consolidate knowledge across the six different affinity diagrams (one from each session). Following our established affinity diagramming protocol, we categorized the themes and their corresponding groups into higher-level groupings. This process mirrored the affinity diagramming procedure employed by participants, with each researcher taking a note group and systematically placing these groups into larger categories. This approach took multiple iterations before all research team members were satisfied with the final resulting affinity diagram.

In generating the finalized affinity diagram, two distinct higher level categories emerged; feature-driven categories that focus on an experience the design should be be built around, and categories which should apply to all data-driven game streaming experiences. We named these two categories *Design Goals* and *Design Considerations*, respectively. Design Goals encompassed specific functionalities. Categories labeled as Design Goals used the "As a [role], I want to [action]" format. For example, "As a viewer, I want to meaningfully contribute to the game being played by the streamer" denotes a Design Goal category. A related, albeit distinct Design Goal category that emerged was "As a viewer, I want to be able to play a game with/against the streamer." Such Design Goals could potentially be realized through a data-driven game stream experience found in pre-existing audience participation games [252]. However, not all data-driven game stream experiences need to support these gameplay-oriented features; an interface for visually impaired viewers, for example, need not enable game contributions from the viewer for it to be successful. On the other hand, Design Considerations, unlike Design Goals, emerged as a category that describes guidelines applicable to *all* data-driven game streaming experiences. Categories labeled as Design Considerations were largely focused on mitigating challenges and

concerns that participants had about data-driven game streaming, and are framed as questions. For example, "How does the system enable voluntary interaction/opt-in mechanisms for viewers?," is a Design Consideration that emphasizes that no viewer should be forced to participate if they do not wish to, reflecting the stakeholders' emphasis on respecting viewer autonomy.

In total, we identified eight Design Goal categories and seven Design Consideration categories, listed in Table 5.1 and 5.2 respectively. It is worth noting that of the eight Design Goals, only one is focused on streamers and the rest are viewer-facing goals. This is to be expected, as viewers are often the main recipients of novel features to help bolster their engagement, especially since viewers often have the least amount of agency to influence live gameplay [130, 252]. This in turn could have influenced our participants, who may already have familiarity with existing data-driven game streaming systems that focus on viewer-facing features. Besides viewers, the needs of the other stakeholders in our study largely manifested as Design Considerations, most of which related to ethical challenges and managing the workloads of streamers, moderators, and developers.

We then used these findings to prepare for the second phase of the study, a design workshop in which returning participants would create sample data-driven interfaces. Due to the use-case-specific nature of the Design Goal category, we matched each Design Goal with specific games (for our final choices, see Table 5.1). To ensure we were selecting games our participants could use as design material, we sent a survey to all participants before Phase Two commenced. In the survey, participants were asked to rate their familiarity with each of 29 possible games on a scale of 1-5, with 5 meaning they are confident they can comfortably design for that game. We then used this data to create groups of 2-3 participants who were all familiar with the same game, but who had diverse perspectives based on their Twitch stakeholder type.

With this comprehensive understanding, we transitioned to the next phase, the design workshop, leveraging these insights to explore the design space of data-driven game streaming interfaces with stakeholders.

5.3.4 Phase Two: Design Workshop

After the affinity diagramming from Phase One, we were able to address RQ1 and 2 through a list of Design Goals and Design Considerations essential for data-driven game streaming interfaces. With these Design Goals and Considerations in hand, Phase Two shifted focus to address the last research question: How can stakeholders contribute to the design of data-driven game streaming systems? Thus, the overarching goal of Phase Two was to address this research question by exploring the insights from Phase One through a design workshop with stakeholders.

We conducted four Phase Two sessions with a total of 21 returning participants. While Phase Two had availability constraints which affected the session groupings, such as needing to ensure all participants were familiar with a game if they were assigned a design for it, more attention was paid in Phase Two to place participants into groups for brainstorming purposes. Each group needed to have multiple perspectives of stakeholders to ensure fruitful discussion, meaning no group was made up entirely of viewers, for example. Instead, participants were placed into groups with other participants who identified as a different stakeholder type. Each session started with a recap of Phase One, followed by an overview and discussion of our results, the derived Design Goals, and the Design Considerations. We then divided participants into their as-

Design Goal	Example Game
As a viewer, I want to personalize and control the information I see during the stream.	*The Witcher 3: Wild Hunt*
As a viewer, I want the system to give me contextual information that is needed to understand the game stream.	*Baldur's Gate 3*
As a viewer, I want to meaningfully contribute to the game being played by the streamer.	*Portal 2*
As a viewer, I want to be able to play a game with/against the streamer.	*Among Us*
As a viewer, I want to use the system to engage with other viewers.	*Animal Crossing: New Horizons*
As a streamer, I want to use data from the system to better understand my audience and/or my gameplay during or after the stream.	*Sid Meier's Civilization VI*
As a viewer with visual impairments, I want to use the system to make the game stream accessible to me.	*The Elder Scrolls V: Skyrim*
As a viewer who is not fluent in English, I want the system to translate the in-game text to a language of my choice.	*Cyberpunk 2077*

Table 5.1: Design goals derived from Phase One and the games they were matched with in Phase Two.

signed groups. Each participant group was assigned a specific game and Design Goal for which they were to design data-driven game streaming prototypes. Figma boards were set up for each group before the session, containing their assigned Design Goal, relevant game information, and designated areas for designing. Unlike Phase One, where participants used Figma largely for writing on sticky notes and dragging them across the board, Phase Two emphasized Figma's design features for remote collaboration and low-fidelity prototyping.

Before starting group-based design work, participants were first given 15 minutes for individual brainstorming of possible prototypes to create. This was followed by participants being placed in breakout room sessions with their group members for collaborative brainstorming and design refinement. Each group was asked to create two separate designs, with 15 minutes dedicated to each, ensuring equal time allocation for both ideas.

Similar to Phase One, guiding questions were once again provided to participants to aid their design process. These questions were asked when group design activities commenced; What kind of data would you need in order to make the overlays for the game? Would the data need to be bi-directional? Can the viewers send data back to the game or to the viewer? How would the front-end display/utilize this data? How would the user/viewer interact with the overlay?

After the collaborative prototyping period, participants were instructed to refine both of their designs by reflecting on the Design Considerations from Phase One. The full list of Design

No.	Design Consideration
1	How can the system enable features which excite streamers without adding too much to their workload?
2	How can the system incorporate features to help moderators do their job?
3	How does the system mitigate trolling/cheating behavior?
4	How does the system implement industry-standard practices such as privacy and security measures?
5	How does the system streamline the development process to minimize the added work developers need to do to get the system working for them?
6	How does the system foster communication between all stakeholders such as viewers, streamers, and developers?
7	How does the system enable voluntary interaction/opt-in mechanisms for viewers?

Table 5.2: Summary of design considerations found in Phase One.

Considerations from Table 5.2 was provided to participants to encourage them to think about if their designs were not meeting concerns such as enabling voluntary interaction mechanisms for viewers. They were given ten minutes to collaboratively iterate their low-fidelity prototypes with these Design Considerations in mind.

Upon returning from breakout rooms, participants presented their designs to the whole group, facilitating discussions on design intent and allowing researchers and other participants to ask clarifying questions.

In total, participants generated sixteen design prototypes for eight different games and their corresponding Design Goals. To analyze these designs, we employed an iterative coding approach inspired by qualitative coding. Initially, the lead researcher annotated each prototype using sticky notes, capturing key features as described by participants. The goal of this initial annotation was to get a feel for the characteristics of each prototype. Examples of annotations include descriptions of audience participation features in the *Portal 2* designs. Following this initial annotation, the lead researcher engaged in discussions with the research team to review the observed patterns and themes.

Subsequently, a more comprehensive analysis was conducted. The lead researcher categorized each design, utilizing a spreadsheet to delineate more specific features. These attributes, including factors such as chat implementation, data usage, and target user (e.g. streamer or viewer) were continuously refined with each iterative pass of the data. As patterns emerged, additional columns were added to the spreadsheet to accommodate the evolving understanding of the designs. The lead researcher iteratively added columns until every prototype was sufficiently described, leading to a total of 19 columns.

The goal of the previous prototype analysis was to be able to thoroughly capture the facets of each prototype. The next step in our analysis was to explore the underlying relationships between the designs.In addition, we also studied the *process* through which the participants gen-

erated their ideas, collected in our observational notes. We collated behaviors across participants to define common steps that participants initiated in their ideation and analysis, such as their approaches for thinking about what data is needed for their designs and how to render that data. We then integrated the process data with our spreadsheet analysis of the designs, as well as with our data from Phase One of the study, to understand the underlying patterns and relationships between prototypes. This relationship exploration was an iterative process of the lead researcher reviewing the data, sketching possible models, and discussing the meaning of these models with the group.

This process facilitated more robust models that organized the shared attributes we found in the data, ultimately revealing insights which contributed to the formulation of an early-stage design framework for data-driven game streaming experiences. This framework provided a structured approach to understanding and categorizing the diverse array of design elements observed across the study, offering valuable insights into the potential implementation and implications of data-driven game streaming interfaces.

5.4 Design Framework For Data-Driven Game Streaming

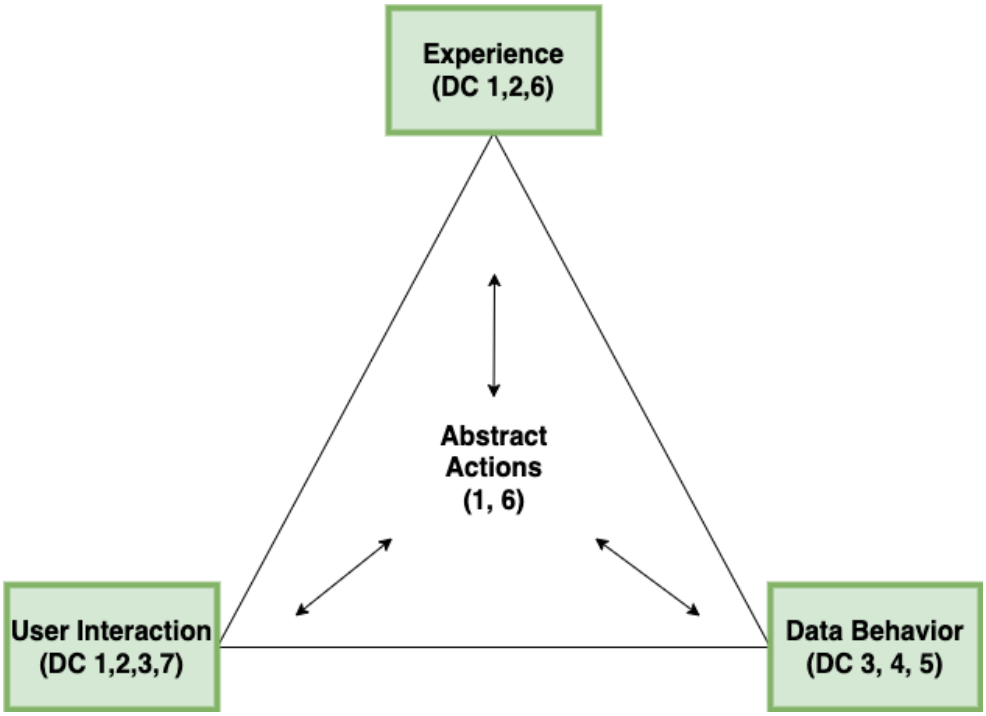


Figure 5.1: The design framework for data-driven game streaming

As part of the sense-making process for our multi-phase study, we created a design framework that incorporates Design Goals and Design Considerations from Phase One, the analysis of the design process from Phase Two, and theories from existing literature on APG design [252, 271]. This framework is illustrated in Figure 5.1, which shows the relationship between the core *components* of our framework. Each of the outer components of the triangle represent a necessary

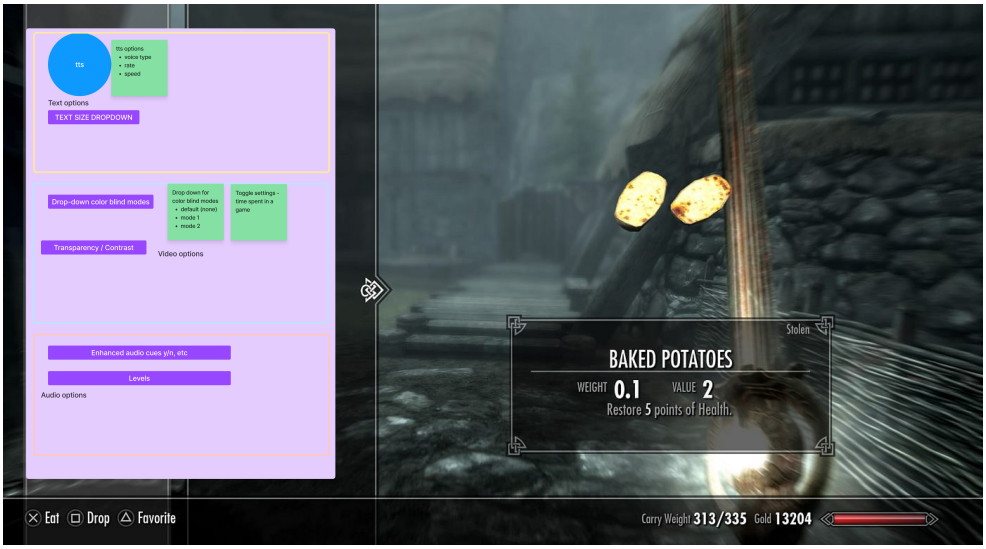


Figure 5.2: A design for visually impaired users using the game *The Elder Scrolls V: Skyrim*

facet of the design: establishing the desired experience (*Experience*), creating an effective interface (*User Interaction*), and powering the user needs and the interface with data (*Data Behavior*). In Phase Two, participants approached their design in a sequential order (Experience, Abstract Actions, User Interaction, and Data Behavior) and were not able to revisit previous steps in their process due to time constraints and the nature of the study procedure. Part of our decision making in structuring the framework in this way is because we noticed early on in our analysis that there was an interconnectedness between the different decisions that a designer would have to make. Decisions related to abstract actions in particular clearly had greater implications for the other aspects of the design. We adapted the framework to reflect that interconnectedness such that design teams, which are comprised of streaming stakeholders, can iterate within each of these components, or move back and forth between them, as part of their iterative design process.

Central to the framework and linking the other components we place *Abstract Actions*, which serve as an *alignment mechanism* between these three perspectives in the design process [53]. These abstract actions help creators consider each of the components within relation to one another, helping them instantiate the experience in interfaces that respect the technical capabilities of a data-driven game streaming environment. The bi-directional arrows to and from Abstract Actions represent this, and highlight that movement between the outer components requires alignment with the abstract actions first. As an alignment mechanism, abstract actions also serve to help interdisciplinary teams stay on the same page, even when they are working on different components of the project.

While Abstract Actions are the core of the framework, design teams can begin ideation from any of the outer components as well. For example, a design team might start from a set of Design Goals based on user research (*Experience*) and use these goals to then consider what interaction mechanisms (*User Interaction*) would be necessary to create them and what in-game metadata (*Data Behavior*) is required. Alternatively, the design team can take a data-first approach, starting from a sense of what metadata can be sent from the game (*Data Behavior*) and from there



Figure 5.3: A design for streamers interested in visualizing data generated from their stream of the game *Sid Meier's Civilization VI*

align with Abstract Actions to ideate on possible interfaces (*User Interaction*) and what novel experiences (*Experience*) they can create. As designers move through these outer components, they can make use of the idea of Abstract Actions as a lens to help coordinate relationships between them.

To illustrate each of the elements of our framework, we make use of three designs created by participants from Phase Two. These include a design for the game *The Elder Scrolls V: Skyrim* [272], which offers accessibility features to visually impaired viewers (figure 5.2); a design for streamers to make sense of data generated while playing the game *Sid Meier's Civilization VI* (figure 5.3) [109]; and a design that enables viewers watching a stream of *Portal 2* [296] to contribute to gameplay (figure 5.4). We then demonstrate how the framework can be used to better understand existing systems by using it to analyze *Helpstone* and how the interaction opportunities afforded by the system relate to its target experience and available data [173].

5.4.1 Abstract Actions

The central component of the framework is about understanding abstract user actions by asking the fundamental questions: What will the user do? What actions are they expected to take? These abstract actions are *interface-agnostic*, meaning they are not concerned with the visual appearance or specific implementation of the design. The focus of this component is to identify *what* actions need to be performed rather than dwell on *how* these actions can be taken.

In our Phase Two study, we extracted abstract actions from the process we observed from our participants in Phase Two, where they only brainstormed the user actions after discussing the Experience and never iterated upon these actions. However, in a more realistic situation, design teams would repeatedly return to the Abstract Actions component to ensure that their design vision, the interactions they create, and their technical and data behavior decisions are

aligned. For example, a team might revisit their abstract actions after modifying their target user persona, after testing a particular interaction, and/or after identifying and implementing a new data pipeline they can integrate. Any resulting changes to the abstract actions then inform the iteration at the outer components of our diagram.

As part of Abstract Actions, design teams are invited to consider how their work enables exciting features without overwhelming streamers and how the actions can foster communication among stakeholders (Design Considerations 1 and 6) The abstract actions for the *Skyrim* design are:

1. Manipulate game text properties
2. Manipulate visual properties of the game such as contrast
3. Enhance in-game audio volume
4. Filter out audio channels



Figure 5.4: A design for viewers to contribute to gameplay in *Portal 2*

The abstract actions for the *Civilization VI* design are:

1. View notable chat messages and keywords sent by viewers
2. View region-based statistcs about viewers
3. View metrics across the timeline of the stream such as viewer retention data and key moments. This timeline feature allows streamers to see trends across the whole streaming session and can be revisted for post-hoc analysis after the stream ends

Finally, the abstract actions viewers can take in *Portal 2* are:

1. Select team (with streamer, against streamer) to join
2. Contribute to the decision of what objects to add to the environment to help/hinder the streamer
3. Contribute to the decision of what objects to freeze in the environment to help/hinder the streamer
4. View the game level or environment

Mode	Target	Overlay	Game Environment	Game Mechanics	Communication
Individual		View notable chat messages/key-words, view region-based metrics, view timeline-based metrics	None	None	None
Collective		None	None	None	None

Table 5.3: Example of the interaction dimensions with the *Sid Meier's Civilization VI* design.

In particular, note that the *Civilization VI* actions are focused on viewing and sense-making, instead of being limited to activity verbs as one might expect in a game environment [21]. Similarly, note that the actions in *Portal 2* are interdependent with those taken by other players, which we address in the User Interaction component of the framework.

5.4.2 Experience

The primary focus of the Experience component is on establishing a clear design goal and target user. In our study, participants began their prototyping in Phase Two at this component since they were assigned a Design Goal at the start of the study session. The Experience component is representative of the Design Goals derived from Phase One, and the key questions addressed in this component are: What is the overall goal of the design? Who are the intended users? While our sample primarily targeted viewer needs, design teams can extend beyond this stakeholder group; additionally, even within these groups they can identify specific user subtypes as well, such as visually impaired viewers. This component is also where design teams consider how they can accommodate streamers without overloading them with additional tasks that hinder their experience, how their goals impact moderators, and how their goals can foster communication across stakeholder groups (Design Considerations 1,2, and 6).

Our examples demonstrate some of these possibilities. For the *Skyrim* design, the design objective was to create a system to make a game stream accessible to a viewer with visual impairments. For *Civilization VI*, the objective was to enable a streamer to visualize real-time data from a stream to better understand their audience or gameplay. For *Portal 2*, the objective was enable gameplay motivated viewers [252] to meaningfully participate in the streamer's game.

5.4.3 User Interaction

While abstract actions asks design teams to identify *what* abstract actions the user can take, the User Interaction component focuses on *how* the user completes those actions. Our approach to this component is designed to help teams navigate the specific challenges that come with pro-

Mode	Target	Overlay	Game Environment	Game Mechanics	Communication
Individual		Select team to play with or against streamer using overlay buttons; select "Freecam" button to explore game environment	"Freecam" feature to view game environment independently from other viewers and streamer	None	None
Collective		Select what objects to place or freeze from available pop-up	None	Click to vote on objects to freeze or place	None

Table 5.4: Example of interaction dimensions with the *Portal 2* design.

totyping interfaces for streaming (e.g. [116]). In our study, the participants began sketching low-fidelity prototypes after discussing the goals and actions they wanted to explore (Experience and Abstract Actions components), and we derived the elements of this component when analyzing their resulting designs. Design teams could start at this component if they have particular features or user interface features they wanted to prototype first. They can then return to this component after re-aligning with the abstract actions of the design.

Along with prototyping, this component comprises of two stages where design teams must first establish the *interaction dimensions* followed by the *responsiveness* of the target user's actions.

5.4.3.1 Interaction Dimensions

Interaction dimensions can be broken down into the *modes* and *targets* where they take place. Here, the emphasis is on describing and classifying interaction opportunities available to the user. Our use of modes and targets builds on Striner et al's design space of *agency* [271], and on individual and social agency as defined by Seering et al [252]. We expand this prior work to generalize the desire for "personalized control" beyond just play and APG-based mechanics, and to capture a broader array of multi-user interactions in the data-driven game streaming design space that do not necessarily have to be "social".

The two modes of interaction are *individual* and *collective*, each having the possible targets of *overlay*, *game environment*, *game mechanics*, and *communication*.

In an individual interaction mode, the result of an interaction is, as the name suggests, personalized to a single user. The interaction does not impact nor rely on the experience of other users of the system. For example, an interface that allows viewers to independently hover over

game elements to view their attributes is considered an individual interaction, as viewers are free to hover on any element they want without the hover interaction affecting other viewers. Interactions classified under the collective mode, on the other hand, offer capacities that impact and/or rely on other users' actions, resulting in a shared interaction experience among users. For example, an interface that allows viewers to click on a point in space and add a sticker to the game's environment is a collective interaction, as that sticker is now visible by other viewers. In our study, we observed individual and collective interactions manifesting in four main ways.

We capture these as four types of interaction, which offer different types of *agency* for the user to impact the experience. These targets are: overlay, game environment (which includes visuals and audio), game mechanics, and communication. The overlay target pertains to interactions that directly manipulate the rendered overlay. The game environment target grants users control over what they see and hear within the game, while the game mechanics target allows users to engage with game mechanics to progress gameplay, such as helping or hindering mechanics in APGs [116, 252]. Lastly, the communication target encompasses the channels dedicated specifically to facilitating communication among stakeholders like viewers, streamers, and developers.

Taken together, the interaction modes and targets allow designers to break down the user experience into interactable components, which can help design teams evaluate the patterns and tradeoffs in their design. Tables 5.3 and 5.4 show example task breakdowns of the *Civilization VI* and *Portal 2* designs respectively. The *Civilization VI* example shows how multiple types of interaction may target the same mode (individual) and target (overlay). While there are differences between the actions, they all involve the individual viewer interacting with overlay data.

The *Portal 2* example, on the other hand, demonstrates how a single design can include actions in multiple modes (individual, collective) and with multiple targets (overlay, environment, mechanics). More specifically, this demonstrates how a single abstract action such as "Contribute to the decision of where to place added object" can be broken down into multiple target interactions. Here, the overlay target describes the UI components that users interact with, such as using a pop-up which appears when voting for object placement is available. Along with this overlay target, the underlying mechanism powering the decisionmaking of where to place the added object is a game mechanics target, which is controlled using a voting system.

5.4.3.2 Responsiveness

In addition to interaction modes and targets, the next aspect to consider is the *responsiveness* of the design. This is the frequency of interaction and updates, which can be inferred by the interaction dimensions. For example, actions in the collective mode often involve coordinating with other users, which can slow the pace of updates. and result in lower responsivness.

Given that accurate response times are impossible to generate at the low fidelity prototyping stage, designers can estimate how long the user needs to wait to see the results of their actions as "low", "medium" or "high." For example, a low responsiveness action in the *Portal 2* design is when a viewer places a vote for their decision to freeze an object to help or hinder the streamer using the voting pop-up. After a viewer places their vote, they have to wait until the voting system aggregates all votes to see the outcome of that action. Conversely, in the *Skyrim* example, the interaction to manipulate the text properties of the game manifests as highly responsive in this design. This is due to the design's use of individual modes of interaction that do not depend

Interaction	Mode	Target	Responsiveness
Select team to play with/against the streamer using overlay buttons	Individual	Overlay	High
Select what objects to place/freeze using available pop-up window	Collective	Overlay	Low
Click to vote on objects to place or freeze	Collective	Game Mechanics	Low
Use freecam feature to view game environment	Individual	Overlay, Game Environment	High

Table 5.5: Example of responsiveness with the *Portal 2* design.

on other stakeholders.

Table 5.5 shows the responsiveness of the *Portal 2* design. The table lists the interactions from the interaction dimensions table for that design to illustrate the impact those decisions have on responsiveness. Here, the individual interactions of the design result in high responsiveness, whereas the collective interactions yielded low responsiveness.

In terms of Design Considerations, the design team is again asked to consider how their evolving design is accommodating the streamers' workloads, impacting moderators' workflow, mitigate cheating, and enable opt-in participatory mechanisms for viewers (Design Considerations 1,2, 3,7).

5.4.4 Data Behavior

The Data Behavior component is concerned with one of the critical pillars of a data-driven game streaming system; the data itself. In our study, we invited our participants to consider the data needed to power their designs towards the end of the study sessions after they had completed two iterations of prototyping. However, design teams can start here if they already have existing data pipelines they want to leverage or if they want to design around the technical constraints of their systems. Design teams can return to the Data Behavior component if further iterations of the design require a re-evaluation of the required data behavior.

To capture the essential data operations needed from each stakeholder, we borrow from file system terminology to define the *data privileges* of each stakeholder. In considering data operations, four primary data privileges emerge: reading, writing, deleting, and manipulating data. While reading, writing, and deleting data are fairly self-explanatory, the concept of data manipulation warrants further exploration. Data manipulation refers to taking existing data and altering its properties. Filtering menu data or altering the color of video data are examples of manipulations. The framework uses these four data operations to prompt design teams to consider what data is needed to power their design and what the stakeholders' relationship to that data is.

Along with data privileges, we define the concepts of data *synchrony* and *throughput*, which capture how often data needs to be sent and the volume of data that needs to processed. Similar to responsiveness, throughput can be estimated by design teams using "high", "medium", and "low" to denote volume.

By comprehensively analyzing data behavior, the framework empowers design teams to tai-

Data	Viewer Privileges	Streamer Privileges	Developer Privileges	Synchrony	Throughput
Audio channel properties	Read, Manipulate	None	Write	Synchronous	High
Game visual properties	Read, Manipulate	None	Write	Synchronous	High
Game text	Read, Manipulate	None	Write	Synchronous	High

Table 5.6: An example of the data behavior component with *Skyrim*.

Data	Viewer Privileges	Streamer Privileges	Developer Privileges	Synchrony	Throughput
Camera location	Read, Manipulate	None	Write	Synchronous	High
Freezeable and placeable objects list	Read	None	Write	Asynchronous	Low
Voting data	Read, Write	None	Read, Write	Asynchronous	Low

Table 5.7: An example of the data behavior component with *Portal 2*.

lor data operations to suit the diverse needs of stakeholders and, to understand the constraints and opportunities the available data place on the design, and to optimize the overall user experience. The data behavior of the *Skyrim* and *Portal 2* designs can be found in Tables 5.6 and 5.7 respectively. The *Skyrim* design enables viewers to personalize their game viewing experience for accessibility purposes, which means viewers *read* incoming data and can *manipulate* its properties. The nature of this design requires synchronous data because viewers expect to see their personalizations happen in real-time; if the data pipeline is disrupted, the user experience is disrupted. The data required is classified as high throughput, since audio, visual, and text data can often be large in size.

Conversely, the voting features of the *Portal 2* designs means that viewers need to *write* that data for developers to *read* in the game. Developers then need to *write* data in response to those voting actions. Additionally The voting data needed for the *Portal 2* design is asynchronous in nature, since delays are necessary to wait for all viewers to finish voting. This data in this case is also classified as low throughput, since vote tallying does not require large volumes of data to be written by each viewer. Though it is theoretically possible that a stream may have hundreds of thousands or millions of viewers attempting to vote, we still classified this as a low throughput data operation since it is unlikely that a stream will have enough viewers to throttle the data pipeline with viewer votes.

With respect to the Design Considerations at this component, design teams are asked to consider the aspects of the data that can mitigate trolling/cheating, implement privacy and security measures, and reduce the workload on developers (Design Considerations 3,4, and 5).

Mode	Target	Overlay	Game Environment	Game Mechanics	Communication
Individual		Use clock widget to synchronize with the streamer, hover over cards to browse historical info, use widget to browse through previous turns and see what cards were played	None	None	None
Collective		Draw lines between zones of interest, use archetype widget to vote on play style	None	None	Use thumbs up or thumbs down to convey feedback to the streamer, write comments to the streamer

Table 5.8: A breakdown of the interaction dimensions in *Helpstone*.

5.5 Analysis of An Existing Data-Driven Game Streaming System

Thus far, we have illustrated our framework using three design examples taken from Phase Two of our study. These three designs were created by our participants through a procedure that started with the Experience component. To demonstrate the framework's flexibility and its applicability to existing data-driven game streaming systems [173], we use it to analyze *HelpStone*. The *HelpStone* system improves communication between the streamer and their viewers, and it does so by offering various interactions for viewers to engage in the gameplay. For this reason, we begin our analysis from the User Interaction component; for reasons of scope, we demonstrate our analysis using the viewer interface rather than also including the corresponding streamer interfaces. Table 5.8 classifies the interaction dimensions of *Helpstone* into their respective modes and targets. It offers communication channels that viewers could use to quickly convey information via the thumb up/down feature, or to provide lengthier feedback in the form of text comments. It also allows viewers to interact with the overlay both individually and collectively.

While *HelpStone* excels at providing a variety of interaction opportunities for viewers, most interactions happen at the overlay target, and there are no interactions that occur at the game environment or mechanics targets. This is due to the fact that the available interactions for the viewer provide suggestions for the streamer that the streamer can take up or ignore, rather than taking actions in the game space directly. Taken together, these insights allow us to see possible future design opportunities for *Helpstone* by allowing viewers to manipulate the game environment properties, or to directly affect gameplay mechanics. For example, a future iteration

Interaction	Mode	Target	Responsiveness
Use widget to synchronize time with streamer	Individual	Overlay	High
Draw lines between zones of interest	Collective	Overlay	High
Use widget to browse cards played in a previous turn	Individual	Overlay	High
Use archetype widget to vote on playstyle	Collective	Overlay	Low

Table 5.9: *Helpstone's* interaction responsiveness.

of *HelpStone* could allow users to customize the game's sound effects (game environment) or add cards to the streamer's deck (gameplay mechanics).

With regards to the responsiveness of the interactions, Table 5.9 summarizes the responsiveness of the available interactions. One notable aspect about the design of *HelpStone* is that its interaction design of features such as drawing zones of interest allow for high responsiveness even in a collective mode. Enabling collective interactions that have high responsiveness was something our participants struggled with, yet *HelpStone* demonstrates that it is possible to create collective interactions that are highly responsive for viewers. In parallel, *HelpStone* effectively uses voting mechanisms to achieve its goals, particularly in the play style interaction where viewers vote on the play style they would like the streamer to take on. This is an example of how low- and high-responsiveness interactions can co-exist in the same experience.

We can identify the abstract actions of *HelpStone* from the User Interaction component analysis. The abstract actions for viewers in *HelpStone* are:

1. Synchronize to streamer view
2. Provide gameplay suggestions to the streamer
3. View historical information about previously played cards
4. Provide real-time feedback to the streamer's actions
5. Comment on gameplay to communicate with the streamer or other viewers

Using Abstract Actions, we are able to see the data needed to power these interactions. Table 5.10 summarizes a subset of the data needed to power the interface. In this case, our table shows that game developers are not highly involved in reading data, which is due to all of the data needed to power these interactions being *retrospective* as it is obtained from game logs [173]. In other words, the data is generated by the game, but then is read-only by all other stakeholders. This constraint provides technical reasons why *Helpstone* as originally designed does not interact with the game environment or the mechanics.

In the case of *Helpstone*, the designers had clear Experience goals and shared them in their research paper [173] (Fostering viewer-viewer engagement, contextual information, and meaningful game contribution for viewers). It is perhaps disingenuous of us to pretend that we are not aware of these goals during this analysis. However, we can see that the goals as discussed in the paper are linked directly to the abstract actions that we describe. Because these actions are then connected to both the user interface and to the data layer, we can see how abstract actions provide a way to connect these three design approaches regardless of where one begins the process.

Data	Viewer Privileges	Streamer Privileges	Developer Privileges	Synchrony	Throughput
Streamer Clock Data	Read	None	Write	Synchronous	Low
Viewer Clock Synchronization Input	Write	None	Write	Asynchronous	Low
Viewer line drawing data	Write	Read	None	Synchronous	Low
Viewer voting data	Write	Read	None	Synchronous	Low

Table 5.10: The data behavior breakdown of *Helpstone*.

5.6 Discussion

In conducting this study, our aim was to move the design space of data-driven game streaming forward by working with target stakeholders. We approached this by asking what features streaming stakeholders wanted to see in data-driven game streaming systems (RQ1), what challenges and concerns stakeholders have when it comes to this technology (RQ2), and how stakeholders can contribute to the design of such systems (RQ3). Our work yielded Design Goals and Design Considerations that stakeholders believed were important, and we synthesized these findings into our design framework to contribute the stakeholders' perspectives in the forming of this design space.

Our pursuit of these research questions revealed further lessons and broader implications for future work in this area. We elaborate on these lessons by first discussing in-depth the insights derived from analyzing the Phase Two designs using the framework. We then discuss the methodological insights we derived from working with many types of stakeholders together with regards to managing incentives. Finally, we discuss the implications and limitations of our work in developing the design space to include other forms of data-driven streaming.

5.6.1 Insights From Designs

While analyzing our data and creating the framework, we noted how interconnected the components of our framework were. A notable pattern that demonstrated this interconnection within the framework was the *tradeoffs* introduced when iterating between or within components . Most notably, choosing between individual or collective modes has implications for which targets, responsiveness levels, or data behaviors are easiest or most salient to envision interactions for.

Across the designs our participants envisioned, concepts that used an individual mode never used a game mechanics target. This means there were no capabilities for users to influence gameplay individually, only as a collective. Further, when game mechanics targets were included in collective mode concepts, they exclusively made use of voting mechanisms to deal with aggregation of inputs. This is demonstrated in the *Portal 2* design, and all other designs in our data used similar voting mechanisms. This is a known trade off between enabling individual capacities to

affect gameplay and accounting for scale [252]. However, there are demonstrated ways to afford individual capacities [307] as well as different strategies of aggregation beyond the standard majority rule [174]. This highlights the potential for a broader set of interactions with gameplay mechanic targets which afford individual capacities or deeper collective capacities, as seen in our *Helpstone* analysis.

Similarly, we saw a tradeoff at the responsiveness level. Individual interactive modes, such as the ones found in the *Skyrim* design, tended to be highly responsive while collective mode interactions, such as *Portal 2*'s, were lower in their responsiveness. This is partly a consequence of technical constraints imposed by collective modes' need to coordinate data between multiple parties such as the game, steamers, and other views. Individual model interactions, on the other hand, are not reliant on all parties coming to collective agreement.

From applying the framework to the concepts our participants created, we observed that data behaviors were often emergent from the prior choices rather than being explicit choices in themselves. For example, the collective designs that relied on voting/input for game mechanics tended to be asynchronous and low throughput, as demonstrated by *Portal 2*'s data behavior table (Table 5.7). Conversely, individual designs that have lots of interactions to enable a personalized experience tend to rely on synchronous, high throughput data, as seen in *Skyrim*'s design and Table 5.6. Moreover, because individual modes did not tend to offer direct impacts on the game, users were more often consuming or manipulating data rather than acting as data producers.

An additional notable pattern that we observed was specific to the Design Goal "as a streamer, I want to use data from the stream to better understand my audience and/or my gameplay during or after the stream", which was the design objective for *Civilization VI*. In Phase One, participants commonly expressed a desire for analytics solutions for streamers, which is corroborated in the literature as a common need [195]. However, when asked to envision new concepts to address this need, it was challenging for participants to come up with novel concepts, instead focusing only on using only an overlay target for interaction as opposed to a broader mix of targets.

We hypothesize this might be because of a few different reasons. First, analytics systems are complex and require effective integration and visualization of data, which might be hard to imagine for our participants. Second, participants might be anchored to their prior experience with analytics interfaces, particularly Twitch's built-in data dashboards. Third, there may be a limited data literacy in this population which makes it difficult to understand how different sources of data can serve them and their needs.

5.6.2 Managing Stakeholder Incentives

As a research community, we endeavor to design and develop solutions to support streaming stakeholders, thereby advancing the design space of data-driven game streaming. Creating data-driven streaming ecosystems that are successful in the long run will need the buy-in from all stakeholders involved. However, it is crucial to acknowledge that stakeholders across the streaming ecosystem may have conflicting needs and incentives, and trying to meet a particular stakeholder's needs may impact those of another stakeholder. For example, much has been written about the community and entertainment motivations of viewers and streamers [58, 129], however streamers, particularly those whose primary employment is streaming, have financial incentives that their audience does not [305]. Similarly, game developers and publishers have their own

financial concerns [157].

In our study, these conflicting motivations and needs manifested in the Design Considerations from Phase One, particularly relating to maintaining autonomy and reducing workload. All stakeholders, be they viewers, streamers, moderators, or developers, valued autonomy in their data-driven game stream engagement. Where they differed, however, was how that autonomy is conceptualized. Viewers in our study were eager to design interactions so that they can personalize and impact the stream, while streamers and moderators sought to maintain autonomy in how they manage their communities. As a result, streamers and moderators at times expressed concern that by giving viewers individualized interfaces, it would make it easier for bad actors to disrupt the stream. Maintaining viewer autonomy while allowing streamers and moderators to maintain control of their communities is one possible conflict that can emerge in this context.

Concerns about workload were particularly pronounced among streamers, moderators, and developers, who expressed a desire to innovate without significantly increasing their workload. Streamers are tasked with playing, socializing, entertaining, and managing while live [317], and game developers are asked to enable features and data collection in their games. In our study, both parties were eager to design novel data-driven game streaming features that viewers wanted, however a major concern was how enabling those novel features could add to their workloads. Our findings underscore the importance of designing systems and interventions that alleviate these concerns, ensuring that the development of data-driven game streaming experiences does not unduly burden stakeholders.

Another source of conflict that poses additional challenges relates to the financial incentives of streamers and developers [305]. In the standard Twitch ecosystem, developers would need to ask streamers permission for advertisements. In our study, a professional streamer expressed apprehensions about how enabling data-driven game streaming features at the game-level may result in game developers bypassing them in monetization efforts. Developers on the other hand, may be more inclined to contribute features to a game being streamed if there are financial incentives at play. Balancing these incentives is crucial for fostering a healthy and sustainable ecosystem, necessitating nuanced approaches that account for the diverse motivations of stakeholders.

Our study highlights the significance of understanding the roles and concerns of each stakeholder in data-driven game streaming ecosystems. As researchers, we must integrate these considerations into our methodologies, creating spaces for stakeholders to voice their concerns and designing interventions that address their needs. Moving forward, future research should continue to explore the nuances of these systems, treating stakeholders as shared members of an ecosystem rather than individual groups with isolated needs. By doing so, we can better capture the complexities of data-driven game streaming experiences and foster collaborative innovation within the ecosystem.

5.6.3 Toward a Unified Design Language

In our framework, we capture the desired features of stakeholders in the form of Design Goals along with the Design Considerations to mitigate their concerns. We also provide a structured approach to explore design across various dimensions, including abstract actions, interactions, responsiveness, and data behavior. However, this framework represents just the beginning of a

broader exploration within the research community. To advance the design space further, we propose two key areas of focus: incorporating a broader set of data-driven streaming use cases and considering additional data behaviors.

One use case that warrants further attention is eSports. While our framework applies in some ways to eSports spectatorship, there are nuances specific to eSports that we have yet to account for. For instance, eSports has a competitive motivation akin to traditional sports, with viewers often seeking to analyze gameplay strategies rather than directly impacting gameplay. Moreover, eSports may involve higher production values and additional stakeholders such as color casters, professional eSports teams, and post-match analysts. Financial concerns in eSports, including revenue generation channels, also differ from other forms of game streaming and merit consideration. Seen through the lens of data-driven streaming, the attributes of eSports introduces additional stakeholders while also placing different responsibilities on others. Analysts and commentators, for instance, are eSports-specific stakeholders whose role in the data pipeline remains unclear.

Additionally, it is worth considering how a how data-driven streaming may differ in contexts that do not involve traditional video games, such as coding streams or educational content. These streams have their own norms and constraints, and the absence of a "game" as a digital data source raises similar questions about the nature of the stakeholders' relationship to the data. In a coding stream context, for example, data may come from official technical documentation in addition to the direct content of the stream, suggesting the existence of additional interaction targets that would be context or use case dependent. Expanding the framework to capture additional data channels presents challenges, as it introduces complexity to the data landscape and may affect stakeholder relationships with the data. For example, in an educational context there may be additional privacy concerns related to the use data from students. Stakeholders' perceptions of data may not align with their perceptions of desired features, highlighting the need for a clearer understanding of their existing relationships with the data. Furthermore, the inclusion of new data sources and channels may impact stakeholders in unforeseen ways, necessitating careful consideration and exploration within the design space. By providing a common language for framing these concerns our framework creates a foothold to explore these issues.

5.7 Limitations and Future Work

Our study encountered several limitations that point to avenues for future research and development in the domain of data-driven streaming interfaces. One notable limitation pertained to the low-fidelity nature of the designs generated during our study, coupled with incomplete details regarding data flow processes. This aspect suggests a need for refining the designs and elucidating the intricacies of data flow to enhance the practicality and comprehensiveness of proposed solutions. Additionally, while our framework provides a structured approach to analyzing data-driven game streaming interfaces, its efficacy in guiding design decisions warrants examination in later stages of the development process. Longitudinal studies could offer insights into the framework's performance throughout the design lifecycle, offering opportunities for refinement and optimization.

Another limitation we encountered is that there may blind spots in our findings as a conse-

quence of how we structured the brainstorming and affinity diagramming activities. For instance, while the Design Considerations in our study offered a more balanced view of the concerns of all stakeholders involved, the Design Goals notably focused predominantly on viewers. Future work is necessary to develop design goals that address the experience needs of the broader ecosystem beyond viewers.

Another limitation related to the structuring of the study is that observed that participants primarily defaulted to the existing text chat rather than propose new communication channels, which is why the communication target was often not accounted for in interactions. This may be because participants were not explicitly prompted to think of new ways to communicate with stakeholders.

While our study included a diverse set of stakeholders across the data-driven game streaming ecosystem, our study was constrained by the limited number of moderators involved. Like streamers and developers, the desires of the moderators in our participant pool only manifested in the Design Considerations. Future research endeavors should prioritize recruiting a wider pool of stakeholders to ensure comprehensive coverage of Design Goals that reflect the needs of the diverse stakeholder ecosystem. More specifically, further collaboration with moderators is needed to establish effective Design Goals that moderation practices which are tailored to the unique dynamics of data-driven game streaming ecosystems.

Additionally, the exclusion of eSports and non-gaming streaming contexts from our study's scope limits the generalizability of our findings. An immediate next step for this work is to expand the design space to encompass these domains, exploring how data-driven streaming interfaces can adapt to suit their specific requirements and user behaviors.

Finally, while we identified key Design Considerations, integrating them into the framework remains a challenge. As a result, the Design Considerations mainly come into play as a final reflection in each component of the framework that encourages designers to incorporate the considerations in their design. Future iterations of the framework should prioritize seamless integration, ensuring that Design Considerations are not treated as an afterthought but rather as integral components at every stage of the design process. Addressing these limitations and pursuing future research directions will contribute to the advancement of data-driven game streaming interfaces, facilitating the development of more inclusive, engaging, and ethically grounded streaming experiences for all stakeholders involved.

5.8 Conclusion

In conclusion, our research on data-driven game streaming systems unveils critical insights into the needs, expectations, and experiences of stakeholders within the streaming ecosystem. By conducting a comprehensive two-phase design study involving streamers, developers, designers, moderators, and viewers, we have identified key Design Goals desired from data-driven streaming systems and outlined essential Design Considerations to ensure the technical, ethical, and social integrity of these systems. Building upon these insights, we propose a novel design framework aimed at analyzing and generating data-driven game streaming interfaces, thus paving the way for a more formalized approach to the design and development of such systems. Our framework provides a structured approach for designers to create, analyze, or improve data-driven

game streaming experiences, emphasizing the importance of Experience, Abstract Actions, User Interaction, and Data. By providing a framework for future research, our work contributes to advancing the field of HCI and facilitating the development of more inclusive, engaging, and ethically sound data-driven streaming experiences. In doing so, we push the streaming experience to move past its humble video-and-text origins and into a new generation of live interaction and shared engagement.

This study shifted my attention from what live gameplay data can do feature-wise, to interrogating the relationships that impacted stakeholders have with it. My work here showed that live stream data acts as both a resource and a constraint, and that which of the two it becomes depends on who is engaging with it. Data, in other words, is never neutral, and designing with it means attending both to the people it comes from and to the system behaviors it impacts. In my next project, rather than asking how data circulates among many stakeholders, VibeClip asked how the data held within a creator's own in-progress work might help a tool align with what that creator is trying to express.

Chapter 6
VibeClip: Supporting Expressive Alignment in Generative Music Tools

This chapter presents VibeClip, a creative assistant that helps video creators generate emotionally aligned background music using vibe-based recommendations. Through a formative study and system evaluation, VibeClip illustrates how generative tools can align with creators' intent by integrating trace-like video data into the music generation process. Of the systems in this dissertation, VibeClip is the one that most directly prompted my turn toward creative activity traces, since its design depended on treating the creator's in-progress video data as an input to the tool to support creator alignment.

This chapter has been adapted from Noor Hammad, C. Ailie Fraser, Erik Harpstead, Jessica Hammer, and Mira Dontcheva. 2025. "It's more of a vibe I'm going for": Designing Text-to-Music Generation Interfaces for Video Creators . In *Designing Interactive Systems Conference* (DIS '25), July 5–9, 2025, Funchal, Portugal. ACM, New York, NY, USA, 17 pages. https://doi.org/10.1145/3715336.3735814

6.1 Introduction

Music is an essential component of storytelling in videos. From epic feature-length films to social media shorts, the strategic use of music can profoundly influence a video's emotional impact and memorability [199, 314]. On social media platforms, the vast majority of videos include music (in 2023, 85% of videos on TikTok and 84% of videos on YouTube contained music [71]), though the style and prominence of music in social videos varies widely.

Most "narrative-style" videos such as vlogs, explainer/how-to videos, interviews, and podcasts feature dialogue, voiceover, and/or other audio. For these videos, background music can help underscore a narrative, add energy or emotion, emphasize key moments, and attract viewers [314].

While some narrative-style videos are made by professional teams, most are made by hobbyist or professional-aspiring individuals who handle every aspect of video production themselves [25, 140, 239]. These creators typically do not have expertise in composing or producing music, so they rely on existing songs as background music for their videos. However, finding the right song for a given video is a challenging and tedious process [106]. Creators must be careful to avoid using copyrighted music without permission, so they often rely on popular royalty-free stock music libraries such as Epidemic Sound [261] and Artlist.io [22]. As prior work has found [106, 219, 279], searching these libraries is hard because users have to describe their desired music in words and/or filter by a limited selection of facets (e.g., genre, mood). Creators must also ensure that the song they choose fits the duration and narrative arc of their video, which requires careful previewing of potential options. This can quickly become time-consuming and frustrating.

Recent breakthroughs in AI music generation present new opportunities for creators to make their own songs using only a text prompt [10, 52, 68, 92, 210, 248]. While this technology is promising, prior work has not yet explored how it might be leveraged in the context of video editing and how to best help users overcome the challenges in prompting generative models [75, 330]. This paper provides an initial investigation into this burgeoning design space, exploring the potential and challenges of incorporating generative music into video editing workflows. First, we conducted a formative study with eleven social narrative video creators to understand the role music plays in their videos, their current workflows and challenges, and their experience with existing music generation technologies. Our findings revealed that while many creators lack formal musical training, they often have strong opinions about the kind of music they want. However, challenges arise when they try to articulate these preferences as queries on music search platforms, as creators primarily describe music in terms of its emotional impact or "vibe", rather than technical aspects like instruments, genre, or tempo. We also found that participants valued creative control as well as the ability to explore options and iterate, but current music search interfaces make exploration feel aimless and unproductive.

Based on our formative study as well as prior work, we distill a set of design goals for interfaces that bring text-to-music generation into the hands of video creators. Through an iterative design process, we designed and implemented a creative assistant that embodies these goals with a focus on balancing automated guidance with user control. This assistant automatically recommends an editable "vibe" based on the user's video, translates this vibe text into generated music, and provides a structured interface for iterating on the generated music. Using this prototype as a design probe, we conducted a user study with seven video creators who generated music with and without the creative assistant to understand how this interaction paradigm compares to directly prompting a music generation model. Our findings suggest that vibe-based assistance is useful when creators are open to new ideas, while direct prompting is valuable in situations where creators already have specific song attributes in mind.

In this work, we make the following contributions:

- Findings from a formative study with social video creators that underscore the importance of emotional impact or "vibe" in describing music for videos
- The design and implementation of a vibe-based creative assistant that integrates text-to-music generation into video editing workflows
- Design recommendations for generative music interfaces for video creators, including offering initial guidance based on users' videos, surfacing intermediate structures, and allowing navigation between layers of control

6.2 Related Work

Despite the importance of music in videos, there is limited existing HCI research on the needs and workflows of video creators when it comes to music [106]. In this section, we first discuss the general challenges of human-AI co-creation, then discuss common challenges and potential solutions from prior work regarding music search and recommendation, and AI music generation.

6.2.1 Human-AI Co-Creation

Iteration is a key part of the creative process [75, 338, 339], but when it comes to prompting large models, iteration can often feel unproductive due to their opaque nature [75, 184, 320, 330]. Recent work has explored how AI tools can better support productive iteration by designing human-AI co-creation tools across creative domains including writing [47, 65], image creation [184], video editing [303, 309], and music creation [106, 187, 339].

One common challenge with interacting with large AI models is developing an effective prompt. Prompting a model in natural language might sound intuitive, but in reality, writing effective prompts can be challenging for non-experts due to the "black box" nature of AI models [23, 75, 320, 330] Prior work has shown that effective prompts often conform to specific structures [23, 75, 320]. Human-AI co-creation tools therefore often provide the user with prompt templates [47] or transform user input to improve the quality of a prompt [303, 309, 339].

Prior work has also shown that breaking complex tasks down into smaller, more concrete subtasks can improve model transparency, feelings of control, and output quality [320]. Human-AI co-creation systems break the creative process into concrete stages to help users more easily guide AI output and iterate within individual stages [65, 303, 309, 339]. Our design process was informed by this prior work, and we demonstrate through a design probe how similar strategies might be applied to the creative task of generating music for an edited video in order to give users creative control and enable productive iteration.

6.2.2 Music Search and Recommendation

Frid et al. [106] identified common workflows and challenges in adding music to videos through two formative questionnaires with social video creators. They found that most creators source music from free or paid online music libraries, and the main factor they consider when searching for music is its emotion or mood. Most popular stock music libraries rely on keyword-based search, but constructing a search query with keywords can be difficult, especially for non-music experts who don't have a specific song in mind [219, 279]. Filtering songs by attributes limits the user's expressivity and control to a small set of platform-defined attributes (usually genre, instruments, mood, and theme [22, 261]). Research on multimodal text-audio embeddings shows promise for more effective natural language music search [144, 196], but would still require the user to articulate a concrete description of their goal. Some music platforms also support searching for "similar" music to an input song [9], but this is only useful if the user has an example song in mind.

Prior work has proposed methods to automatically recommend music for a given video, typically by training a model on an existing dataset of music-video pairs [85, 180, 200, 225, 226, 279, 329]. This can be helpful for situations where the user does not have a specific idea in mind. However, most of these models either consider only the visual attributes of the video, not existing audio such as dialogue [200, 226, 279], or they explicitly focus on the domain of music videos, where music is at the forefront and there is typically no other audio [85, 180, 225]. Approaches also exist for recommending music based only on the textual content of an audio story [318], but for narrative-style videos, both the visuals and the dialogue/voiceover are important [314]. Inspired by this prior work, our work demonstrates how both a video's transcript and visual data

can be leveraged to recommend styles of music to generate.

Most of the above methods are fully automatic pipelines that do not allow the user to specify their intent or iterate on the recommended music [180, 225, 226, 279, 329]. There are a few exceptions; McKee et al. [200] allow the user to include a natural language text description of their desired music, and MuseChat by Dong et al. [85] allows the user to iterate once on the recommended song with a natural-language response. However, these systems were not evaluated with users, so it is unclear how they would fit into a video creator's workflow.

6.2.3 AI Music Generation

Research and development of text-to-music generation models has exploded in popularity recently, both in academic research [10, 52, 68, 91, 92, 208, 209, 210, 248] and in commercial platforms (e.g., Stable Audio [13], MusicFX [164], Suno [278], Udio [289], and Riffusion [12]). Most commercial interfaces provide an open-ended text box as the main user input, often with suggested example prompts and keywords the user can try. These models present a new opportunity for non-music experts to quickly create and iterate on their own personalized songs. While AI-generated music has been an active area of research in the Computer Music research community for decades [66], the more recent text-to-music models are an exciting development for two main reasons: 1) they generate high-fidelity raw audio files, as opposed to symbolic (MIDI) music which consists of a limited number of prescribed instrument tracks [106]; and 2) they take natural language as input, as opposed to lower-level music parameters such as time signature and pitch [106]. However, these benefits also have trade-offs. First, while high-fidelity raw audio provides richer and more realistic soundscapes than MIDI music, it is hard to control the output of these models at a fine-grained level with a single text prompt [331, 339]. Second, although these models can take any text as input, like other generative models [23, 75, 320] they tend to perform better when the input has more structure and mentions specific musical properties such as instruments and tempo [14, 331] (see Stable Audio's user guide [14] and the examples typically shown in commercial products [13, 164, 278, 289]).

Prior work has proposed interactions to help users express their intent and control music generation models such as steering along specific structured dimensions [188], conversational dialogue for iteration [337, 339], and drawing notes as input [187]. Zhang et al. [339] demonstrated how an LLM can help interpret the user's intention in order to select the best fitting AI model for the type of edit they want to make to a song (e.g., add or remove tracks, re-generate sections, transpose key). Recent work has also begun to explore how generative models themselves might directly support more user control by providing additional inputs such as images [61], audio [84, 319], and other time-varying attributes like dynamics or rhythm [208, 209, 319]. However, none of this work has considered how music generation might apply to video creators, who have more constraints around the music they want based on their video, such as duration and timing [106].

6.2.4 Music Generation for Videos

Some recent work has developed AI models that can directly generate music given a video as input, by learning a correspondence between music and video representations [81, 273]. Similar to

the aforementioned research on automatic music recommendation, these methods show promise for helping users get started without needing to explicitly describe their intent, but they do not address how users might control and iterate on this initial output.

To our knowledge, the only work that has specifically explored interactions for video editors to generate music is Frid et al.'s Music Creation by Example [106]. This paper proposed an interaction paradigm for non-music-expert video creators to generate symbolic (MIDI) music based on a reference song, and iterate on the output by combining different instrument tracks from different generated songs. Our work is largely inspired by this paper and its focus on bringing powerful AI systems into the hands of non-expert users with a careful balance of automation and control. We build on this work by exploring how interfaces for high-fidelity text-to-music generation can strike a similar balance, to bridge the gap between how video creators conceptualize music and the language generative models expect. Because symbolic music generation systems require lower-level input parameters, a main contribution of Music Creation by Example was to extract low-level features from a reference song and use those as input parameters to generate new music. In contrast, since modern music generation models use text prompts as input, our approach focuses on helping the user express their intent in natural language and translating that intent into a structured prompt that the model can understand. We believe these two approaches can be complementary, as both text and reference songs are likely to be important input modalities for video creators.

6.3 Formative Study

To design music generation interfaces for video creators, we first needed to understand their current experiences with AI music generation, the role music plays in their videos, their workflows, and the challenges they face. To this end, we conducted a formative study with eleven experienced social video creators. Our study focused on the following questions:

- What workflows and strategies do video creators employ when adding music to their videos?
- What challenges do video creators face when adding music to their videos?
- How do video creators conceptualize music?
- What are video creators' perceptions of AI music generation?

6.3.1 Participants

We targeted social media video creators who produce narrative-style videos that include background music, such as vloggers, how-to YouTubers, and travel content creators. Our inclusion criteria required participants to be at least 18 years old, manage their own production and editing processes, and have publicly available content on a platform such as Youtube, Instagram, or TikTok. To reach this audience, we posted a recruitment message on X as well as corporate and university Slack channels, engaged with specialized content creator communities such as Discord servers for YouTubers, and sent cold emails to YouTubers whose work aligned with our study criteria. Prospective participants completed a screener survey to ensure eligibility, which

included questions about their video editing experience, links to their video channels, frequency of music use in videos, and their preferred platforms for finding music.

In total, eleven participants took part in the study. Table 6.1 summarizes the diverse range of content types and platforms used by our participants.

Participant	Content Type	Content Platform(s)	Music Platform(s)
P1	Get Ready With Me	TikTok, Instagram	CapCut stock music
P2	Travel, vlog	YouTube	YouTube, SoundCloud
P3	Sport highlights	YouTube, Instagram	YouTube
P4	Short film, vlog	YouTube, Instagram	Spotify, Artlist.io
P5	Vlog, TV show reviews	YouTube	Vllo stock music, SoundCloud
P6	DIY crafts	TikTok, Instagram	Instagram
P7	Vlog, travel, short film	YouTube	Epidemic Sound
P8	Vlog	YouTube	Epidemic Sound
P9	Vlog, travel	YouTube	YouTube
P10	Vlog, video diary	YouTube	Epidemic Sound, YouTube
P11	Travel, documentary	YouTube, Instagram	YouTube

Table 6.1: Information about our formative study participants, including types of videos they make, the platforms they post their videos on, and the platforms they use for music.

6.3.2 Procedure

Each study session was conducted remotely via Microsoft Teams, consisting of a 25-minute semi-structured interview, a 25-minute task walkthrough, and concluding with a ten-minute semi-structured interview. Prior to the study, participants were asked to prepare an in-progress project to share during the task walkthrough. In total, sessions lasted approximately 60 minutes and were recorded and transcribed for analysis. Participants received a $30 gift card as compensation.

The first semi-structured interview focused on participants' existing workflows and challenges. Questions covered the role of music in their videos, when in the editing process they typically add music, their current strategies for finding and adding music to their videos, what challenges they face during this process, and what an ideal workflow might look like.

Next, participants completed a task walkthrough and think-aloud, adding music to the in-progress video project they had prepared. We interspersed the walkthrough with additional questions about the challenges in music selection, strategies for resolving issues, and their approach to finalizing a song choice. The task walkthrough concluded with a "music supervisor" question: we asked participants to describe their desired music as if they were delegating the music search and selection process to an expert.

We closed the session with additional semi-structured questions focusing on participants' experiences with AI music generation systems. These questions included whether they had used music generation technologies before, their impressions of such tools, and their requested features for future music-generation systems.

6.3.3 Results

Two members of the research team transcribed and cleaned session transcripts. We then employed an inductive coding approach to identify key themes [39].

Overall, music played a critical role in participants' videos in several ways: establishing the creator's brand identity (P6, P11); "captur[ing] the essence" of the content (P1), particularly by augmenting storytelling (P3); and promoting viewer engagement (P3, P5, P9). Our interviews revealed the challenges that participants face when attempting to achieve these musical goals. We elaborate on these challenges as well as the remainder of the study's findings below.

6.3.3.1 Rough-cut first workflow is commonly used for narrative-style videos

In line with prior work [106], most participants used a *rough-cut first* approach, where they edited raw footage into a draft video before integrating music (P1, P2, P3, P5, P6, P7, P8, P9, P10, P11). While some participants were thinking about the music they wanted as early as filming (P1, P4), most began thinking about music only after completing their rough cut, allowing the video's emerging structure to guide their music selection process.

6.3.3.2 Participants conceptualize music in terms of "vibe"

As prior work suggests [106], participants focused on the mood they wanted the music to evoke, rather than referring to musical attributes like instruments or genre. Notably, all participants used the word "vibe" when describing their goals for the emotional resonance of a piece of music. While the term is somewhat ambiguous, it generally refers to the "mood or character of a place, situation, or piece of music and the way they make you feel" [44] and is a common colloquialism in modern English. Participants needed the vibes of the music to align with the "overall feeling" (P6) of the video. For example, P4 described their desired music as "beach sunset summer vibes", which illustrates the flexibility of the term "vibe" when layering multiple moods or atmospheres into a single description. When asked about future interfaces, participants imagined getting music recommendations based on a video's vibe, which they felt would streamline their process (P1, P2, P5, P7, P8).

Some participants also referenced other media to help articulate their vision, whether by pointing to existing artists (e.g. Taylor Swift), soundtracks (e.g., Interstellar), or broader categories like movie styles and genres (e.g., Disney). For instance, P5 described her desired music as "Disney meets Romcom," remixing familiar genres to communicate the tone she wanted. This form of musical reference differs from what previous research has identified [106], as participants are not simply pointing to specific songs but invoking broader cultural touchstones, such as styles, moods, or iconic associations (e.g., "Interstellar" or "Disney") that immediately convey a shared understanding of the desired tone.

6.3.3.3 Music must align with the video

When asked to describe their desired music to a music supervisor, nine out of eleven participants began by providing an overview of the video's narrative to offer context, including both verbal descriptions and scrubbing through their footage while screensharing. Some participants went a

step further, offering a detailed breakdown of how they wanted the music to evolve throughout the video, describing specific points where the music should crescendo or transition to a different mood (P1, P8). In other words, narrative video creators needed music to align with the video's *structure* and *story*, in addition to producing the right vibe.

6.3.3.4 Music search is messy and inefficient

In line with previous findings [106], participants attempted to develop search strategies that addressed known challenges including finding royalty-free music, finding music of the right length and mood, and finding music that would need minimal editing. Strategies included browsing well-known music platforms such as Spotify or Epidemic Sound (P4, P7, P8), using stock libraries integrated into their video editing apps such as CapCut or Vllo (P1, P5), and exploring music from other creators' videos (P2, P4, P6, P10, P11).

Overall, participants were dissatisfied with the search strategies available to them, feeling that the process was time-consuming and aimless. They could not predict whether their search was "going in the right way" (P10), and often felt that finding the right music was a matter of luck rather than skill. Although they wanted to search for music based on vibes, they hesitated to use keywords for searching (P1, P2, P4, P5, P6, P8, P9, P10, P11). It was hard for them to translate "something so abstract like a vibe into something that assists you to find it" (P1) and some believed they were "bad at figuring out [...] keywords" (P2), in part because existing search tools often failed to interpret their keywords accurately. For example, the keywords "Spanish Romantic" returned results fitting only one of the terms, but not both (P2). Even when they found a song that matched their desired vibe, it often turned out to be too short or too long for their video, making it unusable (P1, P2, P8, P10). These challenges led many participants to reuse music to save time (P1, P2, P5, P9. P10), at the cost of settling for less-than-perfect music (P10).

6.3.3.5 Music results can fail when validated with the video

After searching, participants validated their song choice by listening to the music while watching their video (P2, P3, P4, P5, P7, P8, P9). Rather than listening to songs all the way through, participants typically sampled short segments—ranging from two to thirty seconds—to determine whether a track matched the desired vibe (P1, P7, P8, P9). Some participants scrubbed through songs, jumping to the middle or end for brief evaluations (P4, P5). If the "vibes aren't vibing" (P2), the music would be rejected.

Because participants often evaluated multiple songs, they did not want to go to the effort of importing all their music options into their video timeline, so instead played them from their original source. However, the context switching required to validate songs against the video can also be chaotic. P2, for example, described how relying on YouTube often left her with numerous tabs open that "usually have no relation to one another" as she tried to validate potential songs against her rough cut (P2). Participants desired a more streamlined approach to comparing multiple songs against their videos, which was seen as an important part of the creative process (P1, P2, P6, P7, P8, P10, P11). As P7 put it, "I'm not going to be sure [of my song choice]

because I haven't kissed enough frogs to get my prince. You know, there's actually a lot of value in kissing frogs."

6.3.3.6 Participants see potential in music generation, but want to maintain creative control

Most participants were unfamiliar with AI music generation technology (P1, P2, P5, P6, P7, P8, P10, P11), and none had used it in their videos. Nonetheless, participants believed that music generation could address key challenges in their search and evaluation processes, such as helping them create music that matches their desired vibe (P1, P8), enabling quick comparisons between multiple options (P5, P7), and creating personalized instrumentals that alleviate the need to reuse the same songs in every video (P2, P10). Participants also expressed interest in using AI for manipulating musical structure, such as producing variable-length songs tailored to their needs (P1, P8, P9, P10) or aligning the instruments or sections with their video's narrative (P1, P4, P5, P6).

Despite the interest in music generation to help with their process, particularly the potential of natural-language interfaces (P2, P5, P7, P10), participants also wanted to maintain creative control over features such as song length, instruments, and the overall vibe (P1, P3, P4). P4 argued that relinquishing creative control is "what a lot of creatives are scared of," because the process of engaging with the musical material is a valuable creative act. Instead, P4 envisioned music generation tools "that can enhance your ability to be creative, instead of just doing it all for you."

6.4 Design and Prototyping

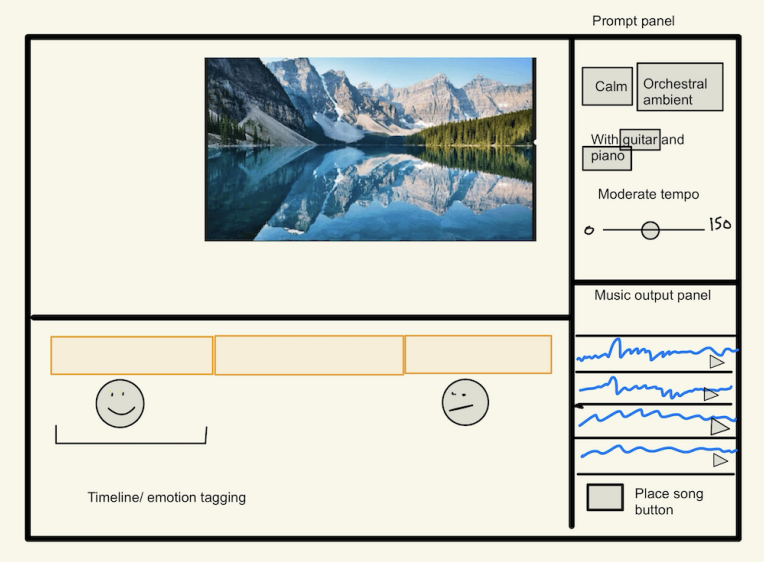
Based on our formative study findings as well as prior work, we identified several key design goals to guide our exploration of integrating music generation into video editing workflows. With these goals in mind, we designed and implemented a creative assistant as a design probe. This section details the goals and process that went into this design, followed by a description of the prototype we built.

6.4.1 Design Goals

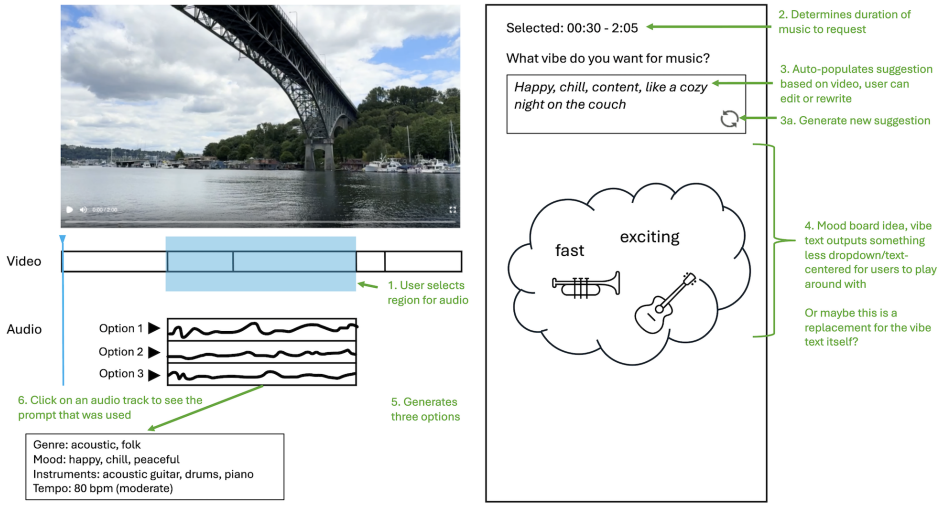
(DG1) **Support a rough cut-first approach:** For narrative-style videos, the most prevalent workflow among participants was to add background music to the video after it has been edited.

(DG2) **Center on vibe and emotion:** Participants consistently described music in terms of its "vibe," corroborating related work [106, 219]. Thus, interfaces should abstract lower-level musical attributes (e.g., genre, tempo, instruments) in favor of higher-level descriptors like vibe and emotion.

(DG3) **Consider video context:** Music must account for the narrative and content of the video [106, 314]. Participants in our formative study framed their music choices by describing their video's content, emphasizing the need for music that aligns seamlessly with the story being told in their videos.



(a) An interface mock-up exploring how users could annotate the video timeline with emotive markers to illustrate the mood of desired music.



(b) An interface mock-up exploring how a mood board could help users illustrate desired characteristics of music.

Figure 6.1: Early prototype sketches created during our iterative design process

(DG4) **Enable music validation with video:** Validating candidate songs with the video before adding them to the timeline was a critical step for participants. This process includes ensuring the music matches the video's vibe and aligns with its duration [106].

(DG5) **Support structured exploration of multiple music options:** Participants emphasized the importance of exploring multiple song options, but their current strategies for doing so were overwhelming and lacked direction. Interfaces that add structure to the exploration process can help it feel more productive.

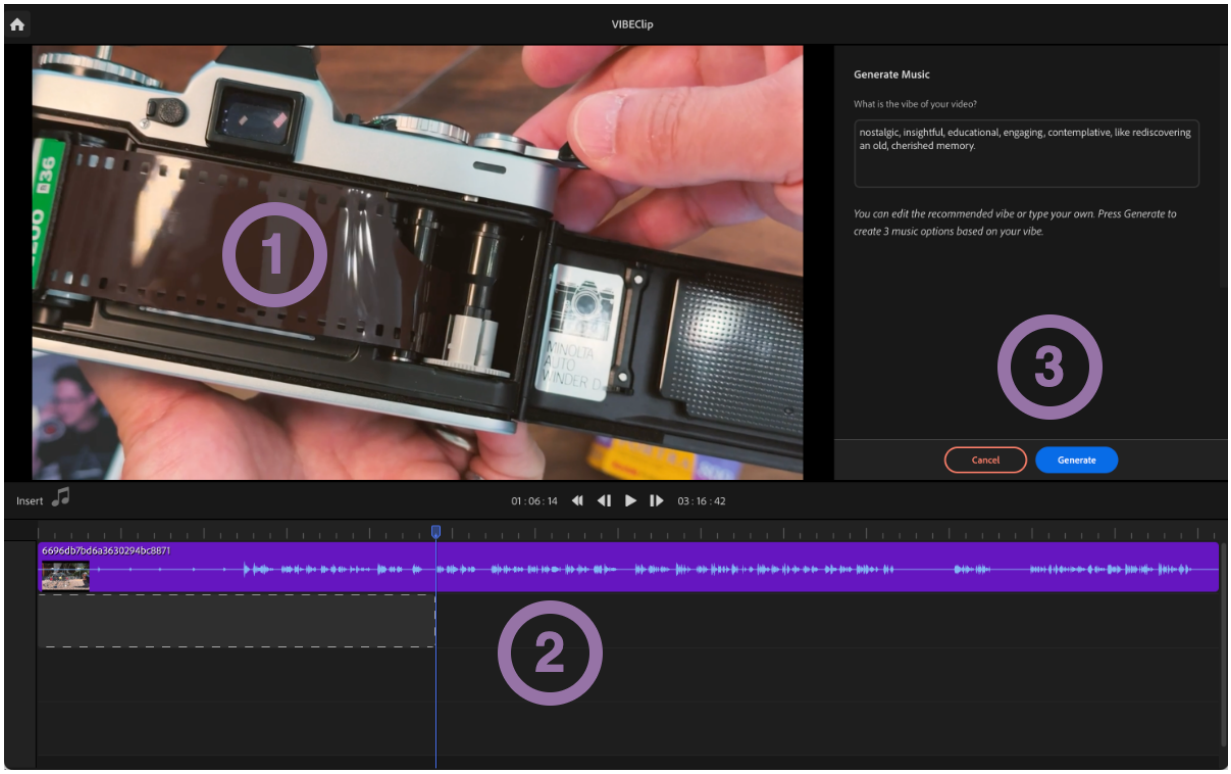


Figure 6.2: The creative assistant prototype featuring a (1) video player, (2) a multi-track timeline, and (3) a music generation panel.

6.4.2 Design Process

With these design goals in mind, we conducted an iterative design process to brainstorm how an interface for text-to-music generation could support video creators. Central to our exploration was determining the inputs the system should process. These fell into two categories: *implicit system inputs for automated assistance* and *explicit user inputs to articulate intent*.

For implicit system inputs, we explored leveraging video data to inform music guidance (DG3) such as:

- Transcript and Audio: Emotion, topic, and narrative cues
- Visual Information: Scene descriptions, prominent objects, people, colors, and actions

- Editing Features: Cut points, cut pacing, text effects, and sound effects

For explicit user inputs, we explored modalities that allow users to express their intent or refine system-generated suggestions. These included:

- Natural language text: E.g., "uplifting, cinematic"
- Visual sketches: E.g., Graphing emotional curves over time
- Audio-based inputs: Humming or uploading reference songs
- Structured input to specify musical attributes: E.g., drop downs to select genre, instruments, tempo, etc.

One early sketch introduced emotion tagging directly on the timeline, where emoji-like characters represented the changing vibe of the video (Figure 6.1a). Though this visual approach helped convey temporal changes, it lacked the expressivity needed for the types of nuanced, verbal vibe descriptions our formative study participants used. We excluded the modalities of humming and sketching because none of our participants mentioned or used them, and we did not want to overwhelm users with novelty given that AI music generation was already a novel technology for most participants. In a later mock-up, we tried supplementing natural language text input with additional "mood board" features, allowing users to drag and drop musical elements (e.g., instruments or themes) (Figure 6.1b). In the final design, we replaced the mood board with structured text boxes to give users the flexibility to type their own text while also producing structured prompts that the music generation model can understand [331]. In other words, although we explored many options in our design process, we chose natural language text as the primary input modality due to its approachability and prevalence in current generative AI interfaces, and because it aligned effectively with participants' existing practices for expressing musical intent.

Eventually, we converged on an interaction paradigm where users can upload a rough-cut video (DG1) and select a section of the video timeline to define the music's duration (DG4). The system automatically recommends a vibe based on the video content in that section (DG3). Users can then edit the textual representation of the vibe before submitting it as input for music generation (DG2). The system generates multiple different songs based on this vibe, and users can then refine finer details of each song in a more structured interface (DG5), which is visible alongside the video (DG4). Like most commercial music generation interfaces today [13, 164, 278, 289], our approach provides the user with suggestions to get them started, but unlike commercial interfaces, these suggestions are tailored to their specific video.

This approach provides a higher-level abstraction of music at the start of the process to support the user's mental model, while providing more structure and control over musical attributes during the refinement stage. By structuring the music generation process around multiple intermediate stages, our design supports a human-in-the-loop paradigm, striking a balance between automated guidance and user control, and aligning with principles from human-AI collaboration research [320]. Importantly, the user is always able to modify or rewrite any AI-generated text to ensure it captures their intent.

6.4.3 Creative Assistant Prototype

We implemented a prototype that embodies our design in a creative assistant powered by a generative AI pipeline (Figure 6.2). The prototype comprises a video player, a music generation panel, and a multi-track timeline. These interface elements and the core interactions are drawn from existing video editor paradigms (e.g., Premiere Pro, Final Cut Pro, CapCut) but are limited in functionality to only the music-related tasks described below. The timeline displays the user's video on the first track, and all generated music is rendered on audio tracks below (DG4).

6.4.3.1 Video understanding

As an offline pre-processing step, we compute two metadata features from the user's video: a transcript of all spoken dialogue in the video, and visual scene captions comprising a textual description of the visual contents of each salient scene in the video. This metadata is used in the following two steps (DG1, DG3).

6.4.3.2 Vibe recommendation

The user starts by selecting a region on the timeline where they would like to add music. This determines the duration of the music that will be generated and its position on the timeline (DG4). Once a region is selected, a recommended "vibe description" will automatically appear in the music generation panel (DG2). This recommendation is generated by prompting a large language model (LLM) to describe the overall vibe/emotion of the selected section of the video. The prompt first includes the transcript and visual captions from the selected portion of the video to target the user's selection, followed by the full transcript and visual captions of the entire video. We include the full transcript and captions to ensure that the LLM has access to the broader narrative of the video with the goal of increasing the relevance of the vibe descriptions (DG3). Since "vibe" is an inherently ambiguous and subjective concept, we also allow the user to edit the recommended vibe description to ensure it captures their intent before moving to the next step.

6.4.3.3 Translating vibe into structured music prompts

When the user is satisfied with their vibe description, they can press the "Generate" button, which translates the vibe description into three distinct music prompts and submits each prompt to a text-to-music generation model. To achieve this translation, the system prompts an LLM to generate three music-generation prompts that incorporate information about the selected clip and the overall vibe (DG2, DG4). We generate three prompts at once to provide the user with multiple options for exploration (DG5) that capture different potential interpretations of the vibe, since there is no one correct way to map an abstract vibe to a concrete music description.

The structure we ask the LLM to use for each prompt is as follows:

- "genre": "1-2 genre descriptors",
- "instruments": "1-2 instrument descriptors",
- "mood": "1-2 words describing the mood of the video",

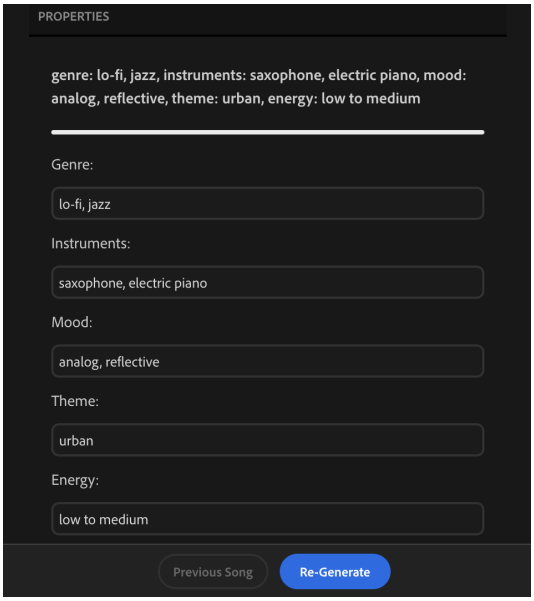


Figure 6.3: Interface shown in the music generation panel when the user selects a song for refinement. They can optionally edit any of the prompt's properties before clicking "Re-Generate," after which the "Previous Song" button will be enabled. If the user clicks "Previous Song," the UI will update to show the previous prompt, and the "Re-Generate" button will change to "Next Song," allowing them to navigate between versions.

- "theme": "1-2 words describing the video theme",
- "energy": "energy level of the video"

This prompt structure is based on publicly available prompt guidance for the popular music generator Stable Audio [14], common music attributes used by popular stock music libraries [22, 261, 269], as well as trial-and-error testing during development.

6.4.3.4 Generating and previewing music

The system submits three requests to the text-to-music service, each with a prompt as per the previous section, along with a randomly-generated seed. The seed is a random number that serves as a starting point for AI generation. We use a different seed for each prompt, following the typical default behavior for generative AI services [8]. Once they are ready, the three generated music tracks are placed on the timeline in the region the user had selected, in three vertically-aligned tracks below the video track. Users can mute and unmute each music item, drag and drop it in the timeline, and delete it using the Delete key. Finally, they can use the playhead to seek and the play/pause button to play the video, as they would in a regular video editor. The video plays in sync with any unmuted music items (DG4).

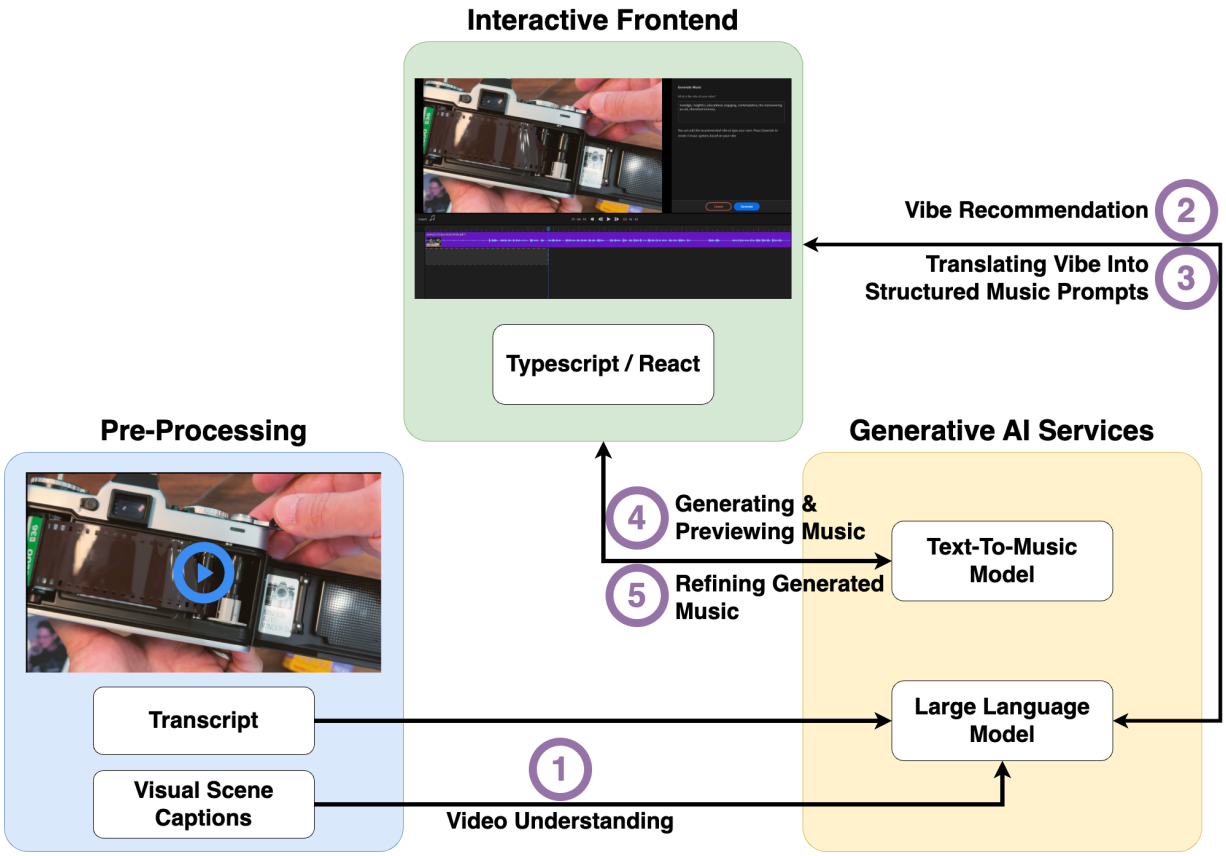


Figure 6.4: Overview of our prototype implementation, starting from the (1) Video Understanding step that generates transcripts and captions for (2) the vibe recommendation. Once the user submits the vibe (3) the system translates the vibe into structured music prompts for (4) the text-to-music model to generate three songs. To iterate on the songs, the user can (5) refine the structured prompt.

6.4.3.5 Refining generated music

The user can select a music item on the timeline to see the prompt that generated it in the music generation panel. Since the prompt conforms to a specific structure (see section 6.4.3.3), the interface breaks the prompt into independently editable text boxes for genre, instruments, mood, theme, and energy (DG5) (Figure 6.3). If the user clicks "Re-Generate" without modifying the prompt at all, the system uses a new random seed in its request to the text-to-music service; otherwise the same song would be generated again. If they did modify the prompt, it uses the same seed as before. This can help maintain consistency across iterations when making minor tweaks to the prompt (e.g., changing one instrument), as it means the generative model will start from the same initial number, making it more likely to produce a similar song. The generated result replaces the selected music item on the timeline, and the user can toggle between iterations using the "Previous song" and "Next song" buttons to hear how their prompt changes affected the resulting music (DG4, DG5).

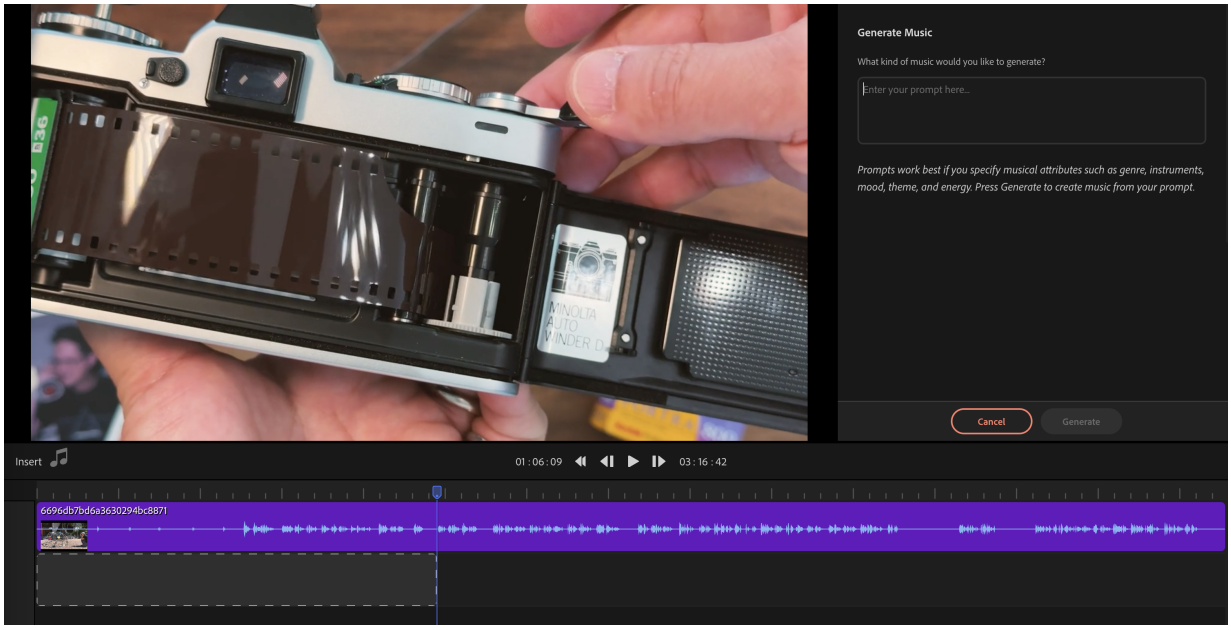


Figure 6.5: The baseline prototype used in the user study. The music generation panel features a single text box where the user enters their prompt.

6.4.3.6 Prototype Implementation

Figure 6.4 illustrates the core components of our prototype's implementation. First, the user uploads a rough cut video. Our offline pre-processing step computes a transcript of the video using Speechmatics [262] and visual scene captions using CLIP [213] and LLaVA [234]. We used GPT-4-Turbo as the LLM that generates the vibe recommendation (section 6.4.3.2) and the structured music prompts (section 6.4.3.3). To generate music, the prototype uses a custom text-to-music generation model, trained on licensed instrumental-only music. The model takes a text prompt and desired duration as input, and generates up to five minutes of instrumental music.

Finally, all user interactions are done through our web interface, developed using Typescript, React, and MobX.

We note that the implementation details of these features are not part of this paper's contribution. In practice, each element could be replaced by other technical solutions, e.g., a different text-to-music model, as long as the components of the system can communicate as per Figure 6.4.

6.5 User Study

Our design and prototyping phase yielded a high-fidelity design probe which allowed us to explore different aspects of the emerging design space, given limited current understanding of how users interact with music generation tools in the context of video editing. We invited seven video creators to use our probe along with a similar prototype with a standard text-prompt interface. We introduced the second prototype as a baseline to compare how our probe, whose design enables assisted human-in-the-loop interactions, compares to the type of interface music generation tools use today. We examined how participants experienced the features of both prototypes and how they compare to their current workflows for sourcing music.

Below we discuss the study details, including the baseline prototype, and the study results.

6.5.1 Baseline Prototype

Since none of our formative study participants had any previous exposure to existing AI music generation systems, we wanted to explore how directly prompting a text-to-music model compares to the assisted workflow found in the probe. We thus created a prototype that does not include any video analysis or automated assistance (Figure 6.5). As in the probe, the user selects a region of the timeline where they want music. Once a region is selected, a text box appears in the music generation panel, where the user can enter a natural-language prompt describing the music they want. When the user clicks "Generate", their prompt is sent to the music generation service. Unlike the probe, only one song is generated at a time.

Participant	Content Type	Content Platform(s)	Music Platform(s)
P1	DIY Crafts	TikTok, Instagram	Instagram
P2	How-To	YouTube	Epidemic Sound
P3	Vlog	YouTube	Epidemic Sound
P4	Vlog	YouTube	Adobe Stock
P5	Get Ready With Me	TikTok, Instagram	CapCut stock music
P6	Vlog	Instagram	Adobe Stock
P7	How-To	YouTube/Internal	YouTube

Table 6.2: Information about our user study participants, including types of videos they make, the platforms they post their videos on, and the platforms they use for music.

When the user selects a music item on the timeline, they can see and modify the prompt that generated it in the music generation panel. As in the probe, clicking "Re-Generate" will submit

the new prompt to the music generation service. We chose to keep the "Previous Song" and "Next Song" buttons from the probe since existing music generation interfaces also allow users to explore their previous prompts and results [13, 164, 278, 289].

6.5.2 Participants

As with the formative study, we targeted social video creators who produce narrative-style videos with background music, with one added constraint: participants were recruited internally from a large tech company due to the prototypes' use of an internally deployed text-to-music model. Interested individuals filled out a screener survey, where we asked about the types of videos they create, how they source their music, and their general video production practices. Selected participants were emailed consent forms and scheduling information. In total, we recruited seven participants, three of whom had also participated in the formative study. Importantly, the formative study did not include exposure to generative music or mention the prototypes, ensuring that repeat participants approached this user study without prior bias or familiarity with the systems being tested.

Table 6.2 summarizes the content types and platforms used by our participants in the user study.

6.5.3 Procedure

Study sessions were conducted remotely via Microsoft Teams, with participants sharing their screens and audio to allow for real-time observation. In total, sessions lasted approximately 80 minutes and were recorded and transcribed for analysis. Participants received a $50 gift card as compensation.

Since our prototype assumes a rough-cut-first workflow (DG1), participants were asked to submit two in-progress videos in rough-cut form, simulating the point in their workflow just before they begin searching for and adding music. They submitted videos ahead of time so we could pre-process them and add them to our internal database.

Each session began with two 20-minute think-aloud tasks, one with each prototype (presented in random order). In each task, participants had 15 minutes using the prototype to add music to one of their submitted videos, followed by 5 minutes to complete a post-task survey. The post-task survey comprised twenty questions on a seven-point Likert Scale. We designed the survey by adapting relevant elements from Cherry & Latulipe's Creativity Support Index (CSI) [57], Amershi et al.'s Guidelines for Human-AI Interaction [18], and similar post-task surveys from related work [106, 303]. See the Supplementary Materials for the full list of survey questions.

After completing both think-aloud tasks, the session concluded with a 40-minute semi-structured interview, which complemented the survey by providing deeper insights into participants' experiences. We asked about their overall impressions of the systems, their preferences, and how the tools compared to their current workflows. Observational notes were also revisited during the interview.

6.5.4 Data Analysis

While the small sample size makes statistical testing inappropriate, subtle variations emerged across specific survey questions, which warranted further investigation to understand participants' nuanced perceptions. To investigate these trends, we used exploratory data visualizations including divergent stacked bar charts and slope graphs. To derive qualitative insights from the interview data, we conducted a thematic analysis of the transcripts using an inductive coding approach.

6.5.5 Results

While both prototypes were well-received, participants had conflicting opinions on which system offered better creative control. The main takeaways from our user study are as follows:

- The design probe was helpful for getting started and exploring possibilities, mainly due to its vibe recommendations and multiple output suggestions.
- The baseline prototype was most helpful when participants had a specific musical idea in mind, and was preferred by participants with musical expertise.
- Participants disagreed on which prototype gave them more agency and control; some found the design probe's structured prompt easier to control, while others found it overly constraining and preferred the baseline's direct prompting approach.

6.5.5.1 Vibe recommendations and structured prompts enable productive exploration

All participants found the vibe recommendations in the design probe to be useful (Figure 6.6), describing the system as a great starting point for the music process that helped them generate ideas and explore possibilities (P2, P4, P7). Several participants likened the probe to a conversational assistant, offering a helping hand when needed (P2, P4, P5, P7). One participant highlighted the vibe recommendation as an excellent way to establish "a solid enough playground to mess around with things" (P5). This participant also found the suggested prompts helpful: "It was good to have a bit of training wheels, I think, because I don't know the words to necessarily use when it comes to describing instruments, for example" (P5). Overall, exploration emerged as one of the probe's key strengths (Figures 6.7 and 6.8): "having the three [songs] laid out is beneficial if you're trying to explore totally different themes" (P1). Participants pointed out that the three music suggestions allowed them to consider genres they might not have explored otherwise (P2, P3, P5) while making it easy to track different iterations (P6, P7).

6.5.5.2 System preference depends on user expertise and situation

Four out of seven participants preferred the design probe for their workflows (P2, P3, P4, P7), citing its flexibility and ability to suggest ideas they might not have considered. Two participants enjoyed using both systems but noted that the baseline was especially useful in scenarios where they had a specific musical idea in mind and needed a more direct approach to realize it (P1, P5). P6, who has more musical expertise than the other participants, was critical of both systems due to the inability to define musical attributes directly. Despite this, they acknowledged being

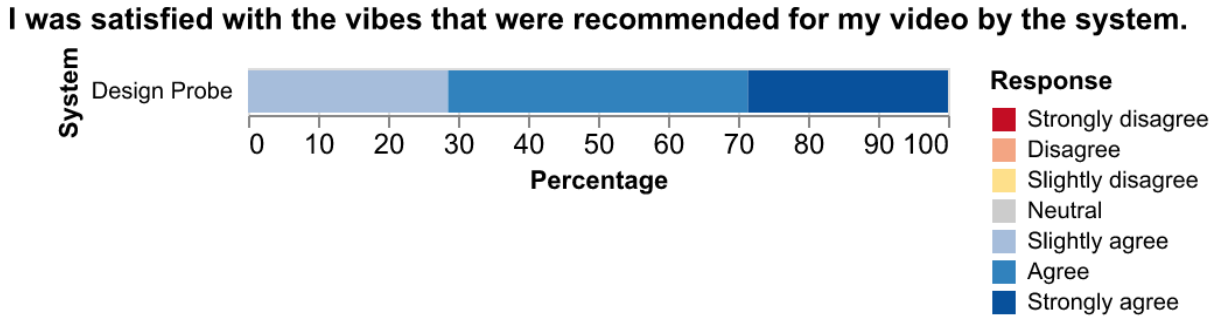


Figure 6.6: Participants' responses when asked about the recommended vibe for their videos

impressed by the design probe's capacity to accurately recommend music that matched their intended vibe.

Participants emphasized that their choice of system would likely depend on situational factors (P1, P3, P5, P6, P7). For example, "if [their goal] was exploration, the [probe] is more ideal" (P1). In contrast, the baseline would be preferable in situations where they already have "a very clear idea of the type of music" (P3) in mind or even "already [have] a track to edit" (P1). However, the ability to describe a specific musical direction using the baseline "would be limited by how musical the user is" (P7). Accordingly, one of the most commonly requested improvements was for the system to "progressively [disclose] more and more controls" (P4), and to allow users to move between different levels of abstraction depending on their use case. P7 echoed this, stating that "it would be neat if there could be an open mode [i.e., the baseline] and guided mode [i.e., the design probe]", so they could choose when they want to use the vibe recommendations versus directly prompt the model.

6.5.5.3 Participants disagree on which system provides more control

Participants had varied experiences when it came to control, transparency, and creativity. While the survey results suggested that the design probe's feature set was clearer (Figure 6.9), there was some disagreement regarding the feedback provided by both systems (Figure 6.10).

Some participants found the design probe to be more transparent than the baseline prototype, as it provided insight into how the system was generating music recommendations (P2, P3, P4, P7) and gave them more output parameters to tinker with (P2, P4, P7). "I felt like I was co-creating because it got my creativity flowing more and I could see what [the system] was thinking [...] It's giving me a bigger look inside the black box of AI" (P7). These participants felt that the baseline system required a significant amount of trial and error to achieve desired results (P2, P3, P7).

Conversely, the other participants felt *less* agency with the design probe, saying that its structured interface limited their ability to refine or control music output (P1, P6). These participants found the baseline prototype easier to guide toward their desired result due to its focus on a single-song output and resemblance to familiar LLM interfaces and music search interfaces (P1, P5).

Participants also shared ideas for future features, some that would increase the amount of au-

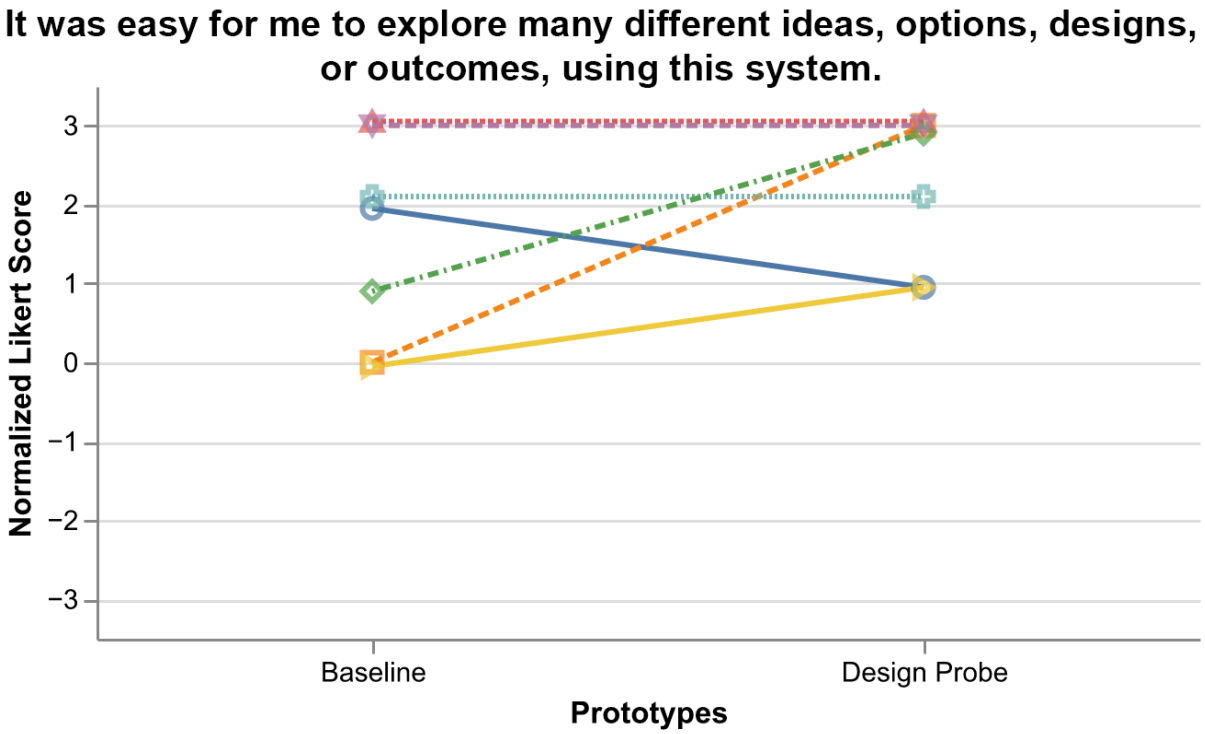


Figure 6.7: Participants' responses when asked about ease of exploring ideas using each prototype

tomated assistance, and others that would provide more fine-grained control. Interestingly, these ideas did not align with which prototype they preferred. For example, both P1 and P3 wanted the system to automatically synchronize the video with their music (increasing assistance), while both P5 and P6 wanted more precise control over the musical structure, such as specifying where a beat should drop (increasing control).

6.6 Discussion and Takeaways

Through our formative and user studies, we gained a deeper understanding of how video creators conceptualize music and explored how a creative assistant can facilitate music generation for videos. Below, we present three key takeaways that can serve as actionable recommendations for designers of future music generation systems.

6.6.1 Offer Initial Guidance Based on User Content

A significant challenge for video creators is describing the music they envision in words. Our findings suggest that offering *personalized recommendations* based on the user's content at the start of the music generation process can be a powerful way to address this problem. Personalized recommendations are more effective than predefined examples, as they directly align with the

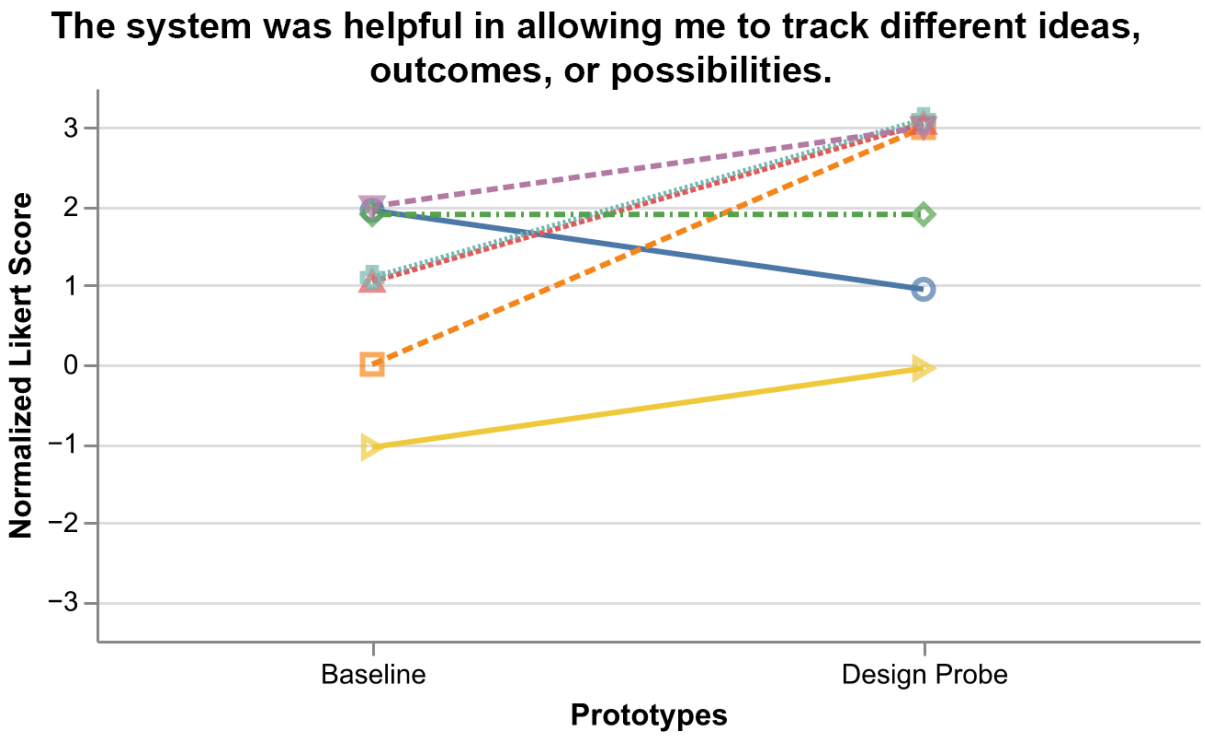


Figure 6.8: Participants' responses when asked about their ability to track ideas throughout their use of each system

user's unique context, enabling them to begin their creative process with guidance rather than guesswork.

In addition to being personalized, initial guidance should align with the user's mental model of a task, so that they can easily modify it to suit their intent. Our approach of describing the vibe of a video was one way to achieve this, but future systems could explore other user-friendly representations such as the timeline emotion tagging and mood boards we explored in our design process. Alternatively, systems could directly generate music given a video by leveraging video-to-music generation models [81, 273], but would need to provide mechanisms for users to iterate on their outputs and guide them towards their creative goals.

6.6.2 Surface Intermediate Structure for Productive Iteration

Prompting generative AI models is challenging [330], especially for users with limited expertise in the model's domain. In the context of text-to-music generation, these challenges are exacerbated by the inherently abstract nature of describing music and the subjectivity of music interpretation. Our findings suggest that surfacing *intermediate structures* can help users iteratively refine their output.

Under the hood, a text-to-music model implicitly interprets and transforms the text prompt in order to arrive at a coherent musical output. In our baseline prototype, the model's interpretation of prompts remained opaque, and participants struggled to infer how their input influenced the

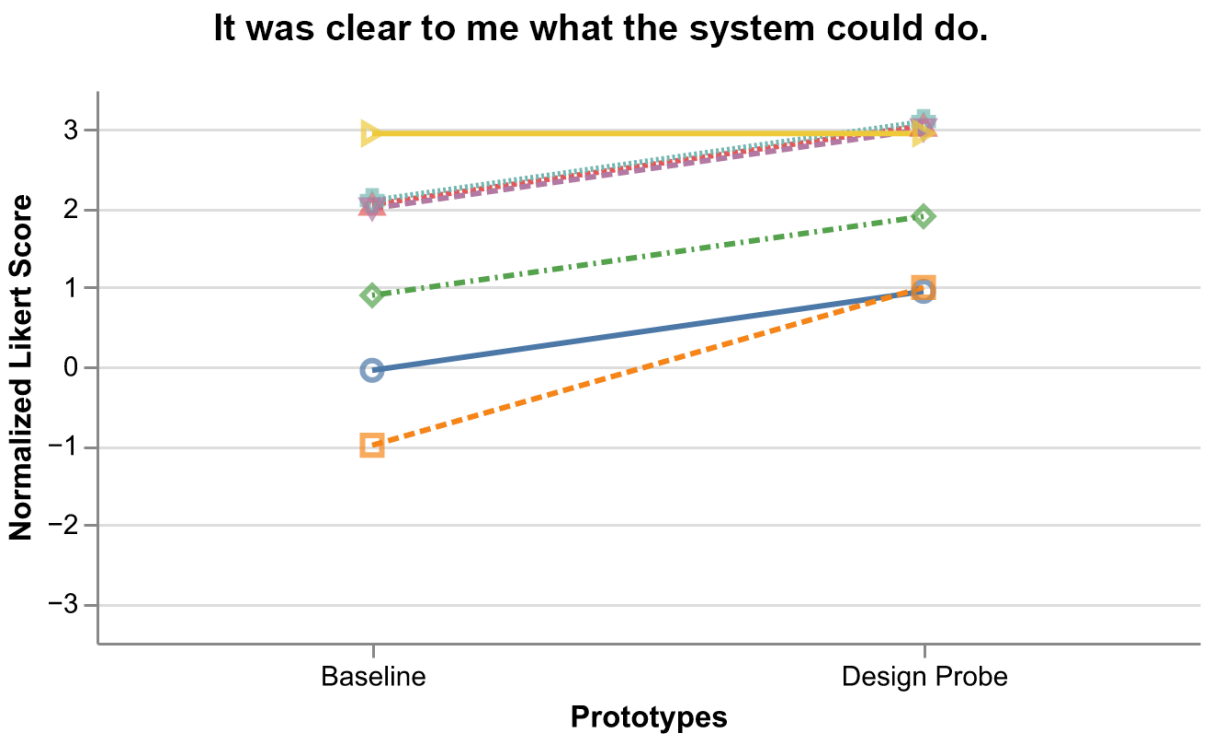


Figure 6.9: Participants' responses when asked about system clarity

output when refining their prompts. While this mismatch between user conceptualization and model interpretation was partially due to users' limited musical literacy, it also stems from the complexity of mapping subjective descriptions to structured musical outputs. Our design probe introduced a structured intermediate representation that allowed users to iterate on their input more effectively. By making the system's interpretation more explicit, this approach provided users with concrete axes of control, supporting more productive exploration and refinement.

While we explored one potential representation for intermediate structure, many other representations are possible. For example, Music Creation by Example [106] employs a grid-like interface that allows users to mix and match five attributes across five generated outputs. Future work should consider other interpretable intermediate structures that provide clear opportunities for refinement and empower users to engage more effectively with generative music tools.

6.6.3 Allow Users to Navigate Between Levels of Control

Our study revealed conflicting opinions among participants regarding which prototype provided better control. Even within the same session, participants' preferences shifted depending on their task or creative intent. This finding highlights the importance of offering users multiple *levels of control* rather than a one-size-fits-all interface solution.

While some participants found the design probe easier to control because of its structured refinement interface, others felt the predefined categories were overly constraining. Similarly, some participants found the baseline's open-ended text box familiar and straightforward, while

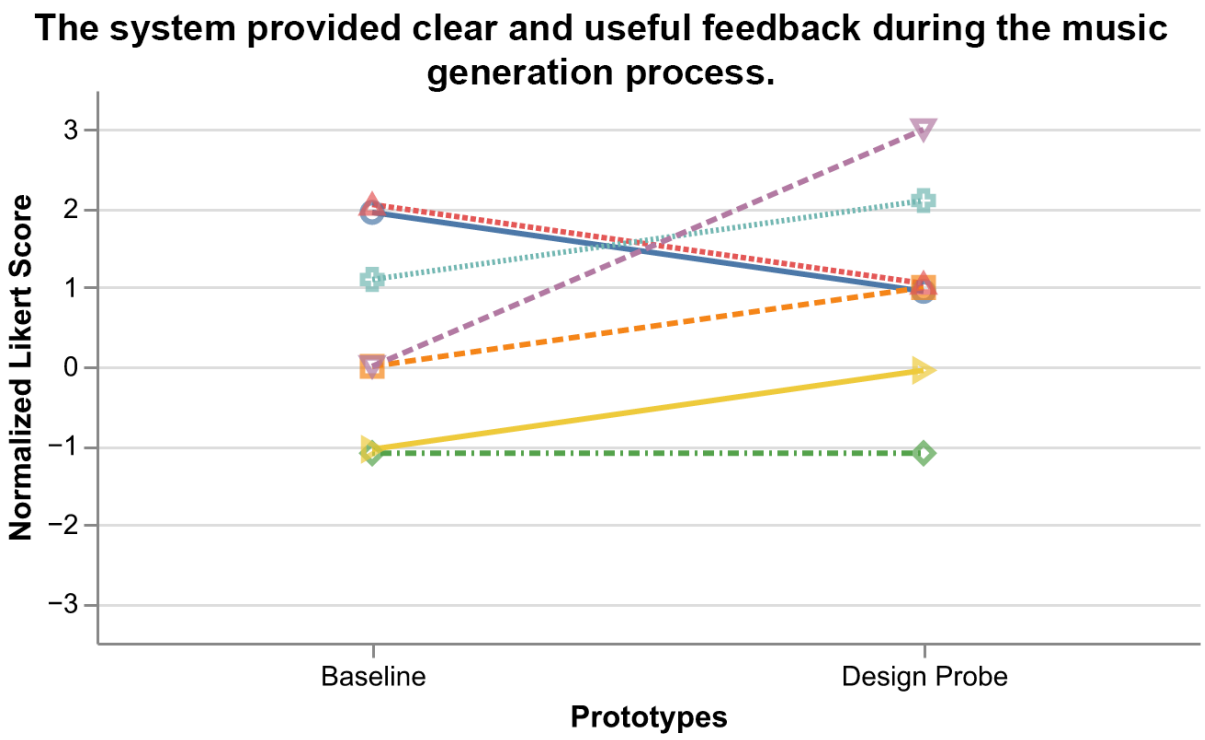


Figure 6.10: Participants' responses when asked about feedback provided by both systems

others found it daunting and unproductive without clear feedback. Future systems must account for these varying needs, offering intuitive ways to move between structured and unstructured modes of interaction. Systems could allow users to toggle between high-level guidance (e.g., vibe-based recommendations) and detailed customization (e.g., control over tempo, instrumentation, or structure) to seamlessly navigate between levels of abstraction as their needs evolve during the creative process. Such systems could also provide educational scaffolding to help address the gap in musical literacy observed among non-expert users, such as tooltips or interactive explanations of musical attributes. This dual role—both as a creative tool and a learning platform—could empower users to make more informed decisions about music while building their confidence in interacting with generative models.

6.7 Limitations and Future Work

Our study has several limitations that point to opportunities for future research. First, the small number of participants and the exploratory nature of our design probe limit the generalizability of our findings. Rather than conducting a full evaluation, we aimed to observe how video creators interacted with our creative assistant to uncover initial insights into this design space. While our exploratory visualizations of the survey data suggest some potential trends, larger studies with a more diverse participant pool are needed to validate the effectiveness of our proposed workflow. Second, our study only tested one music generation model, and some participants had concerns about the quality of the generated music. While other models may vary in their ability to interpret

natural language prompts and quality of their outputs, music generation technology is rapidly improving in general. Future work could compare different models to better understand their relative strengths and weaknesses. Finally, we did not directly compare our creative assistant to participants' existing music search workflows, nor explore how the two might work together. Future work should consider when one approach might be preferred over the other, or even how the two could be blended to combine their strengths—for example, music "outpainting" technology [209] could be used to generatively extend a short stock song to match the duration of a user's video.

More broadly, our design probe represents one point in a broad and largely unexplored design space. We therefore highlight some of the choices we made to emphasize opportunities for future work. First, our prototype requires the user to initiate music generation by selecting a region on the timeline; future versions could enable a mixed-initiative experience by proactively suggesting regions to add music to based on the video's structure. Second, our prototype only provides song-level control over music generation, though some participants wanted to specify how the dynamics or mood of a song should change over time. Future work could explore how we might leverage model improvements that enable finer-grained temporal control (e.g., dynamics or musical structure [208, 209, 319]) by automatically suggesting temporal parameters based on the user's video or providing intuitive mechanisms for users to specify such parameters (e.g., sketching to specify musical structure). Third, our current method of reusing the same seed for a modified prompt does not guarantee that the re-generated result will sound similar, especially if more than a few words in the prompt are changed, because the seed only controls the start of the generative process. Future systems could use the previously-generated song as additional input to guide re-generation, and leverage model optimization techniques to provide specific editing controls such as "inpainting" (replacing a selected region of a song while leaving the rest unchanged) [208, 209] and adding or removing instruments [338]. Finally, future work could also explore how AI music tools might apply to other types of videos, beyond our focus on narrative-style content.

6.7.1 Ethics and Social Impact

While generative AI tools have the potential to democratize creativity, many creative professionals are also concerned about the potential of such tools to disrupt their livelihoods [298]. Our design process and choice to focus on background music for social videos was informed by the desire to enable more creators to do what they otherwise could not: create and customize unique music for their own videos. Our hope is that by reshaping the creative process in this way, our approach might diversify the landscape of music in narrative videos, as creators may be less likely to reuse the same stock songs in multiple videos.

However, the diversity of a model's generated music depends on its training data, which often under-represent non-Western music [308]. In addition, many AI music generation models have come under recent scrutiny for training on copyrighted music without permission [41, 170], thus producing music that directly competes with the data it was trained on. While training models only on licensed music can help ensure that creators have agreed to the use of their music as training data, questions remain around how this broader shift toward generated content might impact music creators, and how to ensure that generated music represents a broad range of styles

and cultures. By addressing these questions, we can build AI tools that empower creators while fostering equitable and sustainable artistic communities.

6.8 Conclusion

This paper explored the challenges and opportunities of incorporating AI-generated music into video editing workflows for narrative social video creators. Insights from a formative study informed the design of a vibe-based creative assistant that balances automated guidance with user control. Using this creative assistant as a design probe, we conducted a user study that compared it against open-ended text prompting. Our results showed that the creative assistant helped spark ideas and enabled exploration. However, it also highlighted the need for creators with specific musical goals to have more direct control. These findings emphasize the importance of flexibility in the design of text-to-music interfaces for video creators, whose needs of exploration, control, and creativity can change depending on expertise and video content. As text-to-music models continue to evolve, our work underscores the need for a design space for music generation interfaces for video workflows that meet the needs of video creators.

VibeClip was the project in which the thread running through my earlier work became clear to me. Its vibe recommendations worked because the system drew on the video the creator was in the middle of making, rather than on descriptors supplied from outside that work. At the same time, VibeClip read only a single state of that process, the rough cut in front of it, and knew nothing of the takes, edits, and earlier decisions that had led there. Looking back across V-Light, MARS, and VibeClip, each system drew on data that creative work was already producing, but none was designed around that data as an accumulating record of the creator's process. In Part II of this dissertation, I begin my formal exploration of creative activity traces and how it can be used in systems to make trace-aware tools.

Part II

Creative Activity Traces and Trace-Aware Tools

Chapter 7
Tracing Creativity: Mapping Activity Traces in Creativity Tools Research

Across my doctoral work, I have designed and built creativity tools in three different media domains, each addressing a distinct challenge faced by creators. V-Light explored how edge computing could expand the expressive possibilities of mobile AR by removing the technical constraints that limited co-located multiplayer experiences. MARS introduced a real-time, game-aware architecture for live streaming, using streamer gameplay data to enable new forms of audience engagement. Finally, VibeClip supported video creators in generating emotionally aligned music by analyzing their own video content to infer mood and intent. Working across these systems revealed a pattern: data, whether from players, streaming stakeholders, or video content creators, has the potential to align tool behavior with creative goals.

Although these systems targeted distinct problems, each relied on robust data pipelines to power adaptive, real-time behavior. Notably, by working on VibeClip, I realized that meaningful adaptation arises when the data utilized in the system reflects the creator's process. What began as a technical effort to build efficient data pipelines to help achieve system goals has since become a deeper inquiry into the nature of the creative process itself. Over time, I came to see that creative practice inherently generates a trail of byproducts that can reveal how creators think and work. These byproducts, in turn, can be a powerful data resource for designing tools that natively understand and adapt to a creator's personalized practice.

In this chapter, I formalize creative activity traces and examine how they are currently used in creativity tools, and in the next, I explore what it means to work with traces as a design material. This chapter is presented largely as published, and it adopts the definition of traces introduced in Section 1.1. It also establishes the basic building blocks of traces: traces *consist of* data, but a trace is not simply the data itself. An individual piece of trace data, such as a single draft, annotation, or prompt, becomes part of a trace as it accumulates alongside other trace data over time. The framework's two dimensions reflect this distinction: its data characteristics describe the trace data itself, while its trace awareness dimensions describe how systems engage with that data as it unfolds over time.

This work was previously published: Noor Hammad, David Chuan-En Lin, Amy Smith, Max Kreminski, Erik Harpstead, and Jessica Hammer. 2026. Tracing Creativity: A Design Space For Creative Activity Traces in HCI. In Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems (CHI '26). Association for Computing Machinery, New York, NY, USA, Article 1644, 1–24. https://doi.org/10.1145/3772318.3791263

7.1 Introduction

Creativity tools have long been a pillar of HCI systems research across multiple areas, including Creativity Support Tools (CSTs), computational creativity, and human-AI co-creation [104, 141]. Despite differences in terminology and technical approach, these systems share a common aim:

to help creators ideate, produce, and share their work in new or improved ways [141, 255]. Today, such tools are deeply integrated into modern creative production and actively shape how creative work is organized and carried out [176].

Creative production involves overlapping processes of iteration. Creators make things over time [117, 267], they grow their skills through the process of making [224], and they develop workflows that suit their needs [194, 216]. As they work, they generate a range of *intermediate outputs* such as reference materials, annotations, partial artifacts, and iteration histories, which are produced alongside the final products ultimately shown to the world [42, 123, 249, 259]. For example, in creative writing, drafts, comments, outlines, and accumulated feedback trace the evolution of a polished narrative and allow writers to revisit prior decisions [257]. In time-based media such as video production, footage, scripts, edits, and sound assets enable creators to compare versions, reuse material across projects, and explore alternative directions [131, 304]. Although these materials manifest differently across domains, they share a common role: they capture the creator's evolving decisions, provide a temporal record of how work develops, and support key aspects of creative practice such as reflection, backtracking, reuse, and adaptation across projects [267].

Creator workflows are both supported and constrained by the tools available to them [27, 176]. Although digital tools are fully capable of capturing aspects of the creative process, current systems differ widely in what they record, how they represent it, and whether they preserve its temporal structure. This inconsistency often flattens creative production into atemporal snapshots, obscuring the iterative journeys and evolving decisions described above. In doing so, tools lose information about where creators *have been* and where they are trying to go, reinforcing the "normative ground" that implicitly dictates how creators should work [176]. These limitations point to a broader gap: we lack a systematic way to describe how tools capture, represent, and act on process data, and have yet to understand how these design decisions shape creative practice.

In this paper, we propose *creative activity traces* as a construct for examining how creative processes become encoded within digital tools. We use the term to broadly refer to process-related data that systems can capture or appropriate, including histories of artifact iterations, reference materials, annotations, and the ways creators organize and document their assets. Digital creativity tools, by virtue of being software, are capable of capturing parts of creative evolution via these traces. Framing intermediate materials as activity traces allows us to analyze, across systems and domains, how aspects of creative process are captured, represented, omitted, or surfaced in creativity tools. In this way, traces serve not only as data, but as a lens for comparing what different tools attend to, ignore, or make visible about creative practice.

Across creativity tools and related HCI systems, process data already appears in many forms [211, 229, 267]. However, activity traces have either been used implicitly in system design without being foregrounded as a critical part of design decision making, or have been examined in a piecemeal fashion across different research communities. As a result, research that engages process data often grapples with similar issues, but lacks a shared vocabulary for describing how such data is captured, represented, or operationalized within tools. This fragmentation makes it difficult to compare systems or to understand the broader design implications of incorporating creator process data.

The need for such a unifying construct is especially urgent, as the integration of generative AI into creative tools has made the capture of complex, multimodal, and fine-grained process

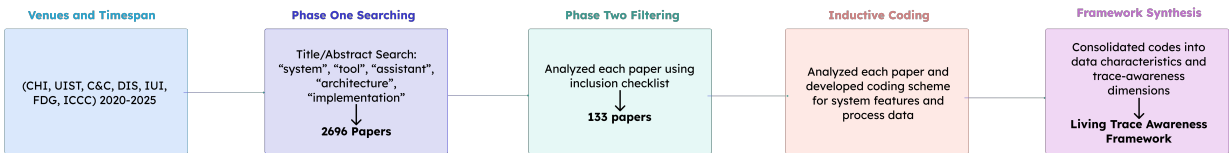


Figure 7.1: Overview of our methodology

data nearly trivial [218, 220]. Without a clear conceptual framework for how traces are captured and used, we risk reinforcing assumptions and practices that were never critically examined, embedding flattened and opaque models of process into increasingly powerful tools. In light of this, this paper addresses the timely question: *How have creativity tools leveraged activity traces to power system features?*

To answer this, we propose a working definition of activity traces within the context of creativity tool design, and use it as an analytical lens for examining how existing systems engage with process data. We collected 133 papers from major HCI venues, spanning research traditions including CSTs, computational creativity, and human-AI co-creation. We analyzed these systems from a traces perspective, taking stock of: (1) how traces are used to power system features, (2) where gaps or underutilized opportunities appear, and (3) how trace usage differs across creative domains. We synthesize our findings in the form of a Living Framework for Trace Awareness, which characterizes the types of creator data that constitute trace, the degree of "awareness" that systems exhibit toward them, and the ways traces are utilized to support system functionality. From this mapping, we surface underexplored opportunities for integrating traces into system design more deliberately and responsibly.

With this work, we aim to establish a descriptive foundation for studying creative activity traces more intentionally. Our goal is not to resolve longstanding debates on creativity, nor to prescribe a definitive model of what constitutes a trace. Instead, we use the term activity traces as a deliberately broad construct that encompasses process data ranging from low-level interaction logs to higher-level intermediate materials. This breadth is intentional, as it allows us to explore concepts that have been historically studied separately while highlighting opportunities for how traces can be used as a "design material" in the future [323]. Ultimately, we argue for more intentional and principled consideration of how traces shape creativity tools and impact creative processes, ensuring that creator data is not left as an unexamined encoding of process, but is intentionally leveraged to represent, support, and align with the evolving richness of creative practice.

7.2 Related Work

7.2.1 Literature Reviews on Creativity Tools

HCI and adjacent fields have produced a steady stream of literature reviews on creativity tools, many of which have become touchstones for the community. The most influential are the broad mappings by Frich and colleagues, which surveyed two decades of creativity research [105] and offered the first structured overview of CSTs [104]. Follow-up work examined how CSTs

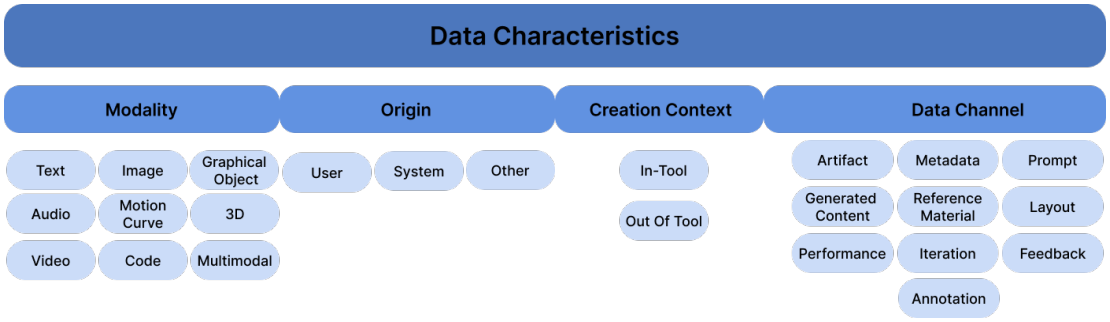


Figure 7.2: The Data Characteristics Of The Framework

are evaluated, noting short study durations, weak theoretical grounding, and few longitudinal deployments [232], while others compared HCI with psychology-based creativity research [105] or scoped creativity support systems in conceptual design [108]. Collectively, these reviews established taxonomies of domains, roles, technologies, and evaluation practices that frame how researchers understand creativity tools.

More recent reviews have narrowed to specific domains or technologies. Chung et al. analyzed 111 art-making CSTs to map intersections of users, roles, interactions, and technologies [63], while Lee et al. produced a design space for intelligent writing assistants [168]. Other examples include Fang on AI in story writing [97], Rodrigues on convergent thinking tools [240], Abi Saad on virtual teams [7], and Zhou on tangible affect [343]. These efforts expand our understanding of domain-specific creativity tools, but remain focused on interface taxonomies of systems and not how such tools understand the creative process.

The rise of generative AI has produced another wave of reviews. Hu et al. cataloged 189 interactive GenAI systems for art and creativity [142], Cong Fang et al. examined GenAI in conceptual design [96], Pang traced the ”LLM-ification” of CHI [218], and Park analyzed HCI applications of pre-trained models through artifact analysis [220].

Taken together, these prior reviews rarely treat creator data as a primary object of analysis. When data is discussed, it is typically at the level of input and output modalities (e.g. text, video, audio) or the data used for model training. The rich creator process data generated within tools, including artifact iterations, edits, annotations, or AI generated materials, are typically collapsed into outputs or not discussed. For instance, story-writing reviews describe how AI generates paragraphs from user input [97], but not how systems capture, represent, or act on intermediate traces. Our review addresses this gap. Like Park et al. [220] and Lee et al. [168], we adopt a systems-oriented approach to our analysis, but we explicitly code our corpus of systems from the lens of activity traces. We analyze what modalities are captured, where traces originate, whether and how they are surfaced in the interface, and whether systems use them beyond visibility. Our goal is not exhaustive coverage of all possible trace usage, but a characterization of trace usage in systems across creative domains.

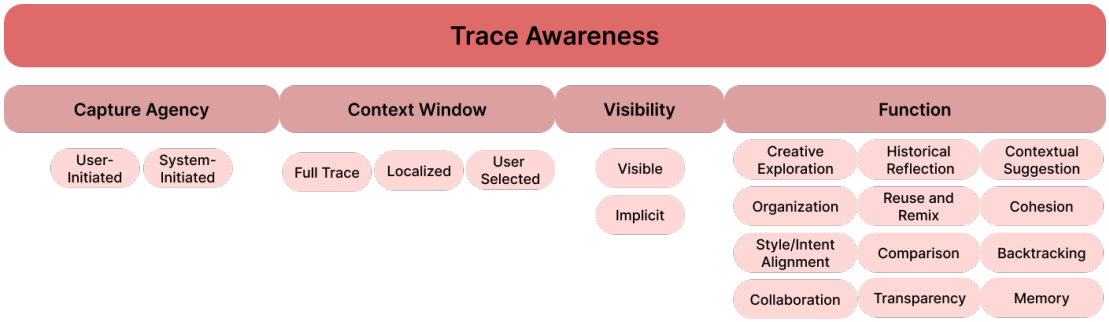


Figure 7.3: The Trace Awareness Dimensions Of The Framework

7.3 Method

We conducted a systems-focused literature review to address the lack of a systematic baseline for how activity traces shape system design. We assembled a corpus of 133 technical papers published between 2020–2025 across major HCI and creativity tool venues, and analyzed each paper using a traces-oriented lens. Our analysis focused on characterizing (1) what features are implemented in the system, (2) what creator process data are captured to constitute an activity trace, and (3) how these data are used to drive functionality. This approach allowed us to identify both the forms of data that systems operationalize and the ways traces are mobilized in practice. Below, we detail our methodology for corpus construction and coding. Figure 7.1 provides an overview of our approach.

7.3.1 Selecting a Corpus

In assembling our corpus, we draw inspiration from Lee et al.'s design space formulation of creativity support tools and Chung et al.'s domain scoping of art-making systems, while extending this approach to focus specifically on creative media contexts [63, 169]. Below, we describe our choices of venues, timespan, and filtering strategies that shaped our scope.

7.3.1.1 Venues and Timespan

We targeted venues where creativity tools are most often published in HCI, specifically the proceedings of *Conference on Human Factors in Computing Systems* (CHI), *User Interface and Software Technology* (UIST), *Designing Interactive Systems* (DIS), *Intelligent User Interaces* (IUI), and *Creativity and Cognition* (C&C). To better capture creative media domains that were of interest, such as game design and performing arts, we also included *International Conference on Computational Creativity* (ICCC) and *Foundations of Digital Games* (FDG). Together, this set of venues provides us with disciplinary breadth while remaining closely aligned with HCI.

We scoped the review to papers published between 2020 and 2025, providing us with a contemporary and bounded time frame. This time span offers us a window to analyze systems before and after the rise of generative AI, while still keeping the corpus size manageable.

7.3.1.2 Search and Filtering Strategy

Because creativity tools are published under diverse labels (e.g. CSTs, creative assistants, domain-specific tools) [104, 141], a keyword-only search for creativity support tools risks omitting relevant systems. We therefore employed a two-phase process: (1) a broad search to capture all *systems contributions* first[315], followed by (2) manually filtering these results using an inclusion checklist (see 7.3.1.3).

In the first phase, we constructed a broad keyword search string including the search terms "system", "tool", "assistant", "architecture", and "implementation" applied to titles and abstracts of papers in the ACM Digital Library and the proceedings of ICCC. We derived these terms from CHI's list of contributions and from Wobbrock et al's contribution types in HCI [315]. Across the seven venues and years 2020-2025, this search string yielded 2696 results. For the smaller venues (UIST, DIS, IUI, C&C, FDG, ICCC), we manually screened the proceedings to validate coverage and confirm the search string reliably captured all systems contribution papers. This gave us the confidence that the automated search strategy sufficiently covered CHI, where manual screening was infeasible.

In the second phase, we manually applied an inclusion checklist (7.3.1.3) against our 2696 results from the previous phase. Because the creativity tool research spans a wide range of professional domains, including engineering, scientific visualization, productivity software, and artistic creation, we needed to articulate a clear boundary for this review. Prior work shows that creativity tools appear under many labels and across heterogeneous practices [104, 141], which creates ambiguity about which domains should be treated as creative media contexts, and introduces feasibility complexities with regards to our analysis. To resolve this ambiguity and produce a coherent analytic scope, we centered our corpus on systems designed for *creative media production*, meaning tools that support creators in producing expressive digital artifacts in domains such as visual art, animation, audio, storytelling, video, or games. As a result of this scope, systems targeting CAD, product design, or professional software engineering were excluded because they center functional, structural, or engineering problem solving rather than expressive media making. Borderline cases were adjudicated by examining the intended creator population and the practice the system supports. Systems were included when programming or modeling served an artistic or expressive purpose, such as creative coding environments aimed at artists [20], and excluded when the primary users were engineering professionals, such as tools designed to support software developers in analyzing their programs [206]. By selecting this focused and conceptually coherent subset of creative media domains, we ensured that the systems in our corpus share a common relationship to creative artifact production and that the trace analysis that follows is consistent across systems. We revisit the implications of this scoping in the limitations and future work section (see 7.6).

Each paper was reviewed by the first author, with team consultations for refinements to the inclusion checklist and on ambiguous cases. This process removed non-systems contributions (e.g. studies without an implemented tool) and creativity tools that fell outside our scope. This process resulted in a corpus of 133 papers. This corpus was verified by the research team and analyzed using the coding approach described below.

7.3.1.3 Inclusion Criteria

To guide filtering in the second phase, we applied the following inclusion criteria. These criteria ensured that the systems under consideration were fully implemented and within scope for our activity trace analysis.

System Contribution

- Papers must describe a fully implemented system
- Only full proceedings papers were considered; shorter formats such as posters or late-breaking work were excluded.

Creative Domain

- Systems must be designed to support creative processes in creative media domains, such as art, audio, storytelling, or games.
- Focus on creative practitioners who produce digital media artifacts, such as art, audio, storytelling, or games.

Creative Composition / Authoring Orientation

- Included systems focus on supporting users to author or perform creative media.
- Papers where the primary contribution was a digital artifact (e.g., a digital game prototype rather than a tool for making games) and not a tool to support making such artifacts were excluded.

Creative Agency

- Systems must preserve meaningful creative decision-making or authorship for users.
- Fully automated tools with no avenue for user control or authorship were excluded.

7.3.1.4 Coding Approach

To analyze the systems described in our corpus, we focused on the paper itself and used any provided video figure as a supplement to understand the system more. This focus on the paper and video figures ensured consistency across the analysis. Our approach for analyzing the systems is inspired by Lee et al's analysis of intelligent writing assistants and Park et al's analysis of genAI prototypes [168, 220]. We adopted an inductive coding approach focused on how systems capture and use activity traces. A pilot round of the analysis was conducted by the first author, where 20 systems were randomly sampled from the corpus. This round surfaced preliminary codes around (1) system features and (2) types of process-related data those features relied on. The preliminary codes were refined in discussion with the research team and iteratively consolidated into a coding scheme.

The final coding was conducted by two authors, with regular consultations to resolve ambiguities and ensure consistency. In this phase, we deliberately defined creator data broadly to encompass any material appropriated or generated during the creative process (e.g., reference

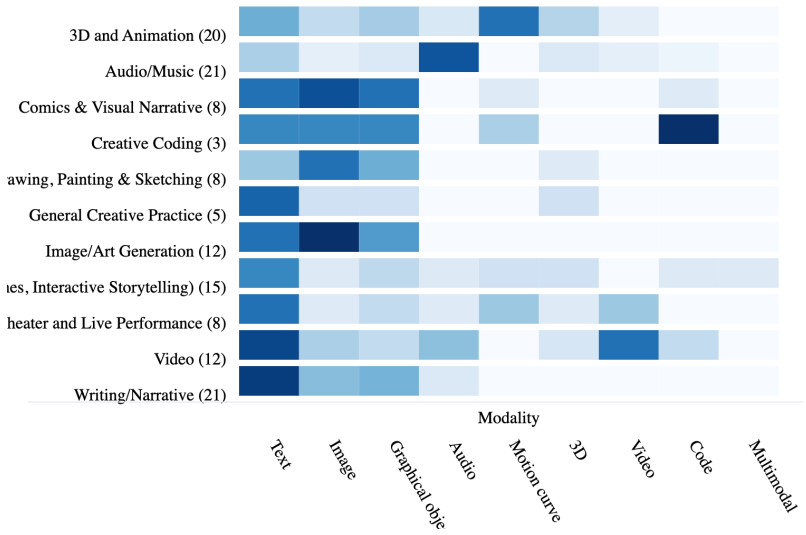


Figure 7.4: Heatmap distribution of modalities across domains.

materials, intermediate drafts, annotations, AI-generated content). Each system was then analyzed with attention to the kinds of data captured, the characteristics of that data, and the way the system utilized the data.

As a result of this coding process, we developed the Living Framework for Trace Awareness, which distills our analysis into a set of dimensions that characterize traces and capture the degree to which systems attend to their temporal evolution. We present this framework in the Results section, where it structures our comparative analysis.

7.4 Results – The Living Framework For Trace Awareness

Our analysis of 133 systems was centered on understanding how traces are being used in system design. We structure our findings through a Living *Trace Awareness Framework* that describes the dimensions of trace usage in HCI in terms of the raw *data characteristics* of trace data and the *trace-aware* capabilities that systems can exhibit. This distinction allows us to examine both the material that constitutes traces and the ways systems temporally leverage that material in practice. Figure 7.2 summarizes the codes we used to characterize data within traces, while Figure 7.3 extends this to dimensions of trace awareness, which capture how the temporal features of traces are utilized. Tables 7.1 and 7.2 provide examples of the data characteristics and trace awareness dimensions respectively.

Our framework can be used both to analyze individual systems and to identify gaps in trace usage within and across creative domains. We describe it as a ”living framework”, drawing from Pais et al.’s terminology [215], to emphasize that this is a preliminary structure that we expect to refine in dialogue with the HCI community as we continue to study traces.

We grouped the systems in our corpus into 11 mutually exclusive creative domain groups, ranging from 3D and Animation to Theater and Performance. Figure 7.5 summarizes the do-

mains and their counts of systems. We also created an additional domain group for "General Creative Practice", which includes systems for general creative ideation [310] and systems designed to support creativity across domains [153, 265]. We use these domains and their associated systems to illustrate how different dimensions of our framework manifest in system design and to surface differences across fields.

Our dataset skews toward post-GenAI publications, with 90 systems published in 2022 or later compared to 43 prior to 2022. This is attributed to both the increase in HCI publications during that time as well as the ubiquity of GenAI-powered features in the field [218].

7.4.1 Data Characteristics

The first dimension of our framework focuses on the underlying *characteristics* of data that are captured within creative systems and that together form activity traces. In our framework, characteristics describe the raw material that traces are built from and provide the foundation for understanding *what* systems are actually capturing. By attending to the properties of the captured data itself, we can see what kinds of creative activity systems consider worth recording, representing, and storing. We identified four recurring dimensions of these characteristics: modality, origin, creation context, and channel.

7.4.1.1 Modality

The modality characteristic describes how trace material is represented in a system. Rather than treating modality as a simple input or output channel, we define it as the medium that systems use to capture and store process data. Table 7.1 lists the modalities observed in our corpus.

Across the corpus, text emerged as the most frequent modality, especially in systems published after 2022. Conversations with LLMs and transcripts increasingly stand in as representations of creative activity, even in domains that are inherently visual or multimodal such as video editing (e.g., [283]) or 3D animation (e.g., [185]). In many cases, multimodal traces are funneled into textual proxies, such as transcripts, captions, or keywords to enable generative AI models to operate over them (e.g., [28, 60]). This strategy highlights the growing role of text as a lingua franca for activity traces, as it is easily understood by users, can capture aspects of intent, and is highly compatible with LLMs. Yet this same reliance risks flattening richer forms of creative practice into textual summaries, privileging interoperability over fidelity.

Other modalities remain central within their own domains. Image data (e.g., sketches, rendered stills) dominates drawing and art generation systems, while audio traces structure music and performance tools (e.g., [46, 107]). Video data is essential in editing and choreography systems, though often translated into transcripts for system analysis. Motion curves and 3D models appear in animation and immersive systems, capturing gesture, pose, or spatial form as process data. Graphical objects refers to vector-based or structured visual elements like shapes, nodes, or UI objects (e.g., nodes, cubes, interactive objects) and are frequently used in systems where a canvas layout is used [20, 60].

These patterns reveal clear domain differences: writing and narrative systems overwhelmingly rely on text, video systems on video+text hybrids, music on audio, and animation on motion curves and 3D (see Figure 7.4). Creative coding tools privilege code and graphical objects,

	Code	Examples
Modality	Text	Prompts, transcripts, annotations
	Image	pixel-based visual media, including photos, sketches, or rendered images
	Graphical object	Vector-based or structured visual elements like shapes, nodes, or UI objects
	Audio	Recorded or reference audio
	Motion curve	keyframe animation
	3D	Meshes, spatial models
	Video	Edited clips, captured footage
	Code	Source code
	Multimodal	Systems spanning multiple modalities
Origin	User	Direct edits, sketches, writing
	System	AI-generated images, dialogue, or suggestions
	Other	External creators or datasets
Creation Context	In-tool	Data captured within the tool
	Out of tool	Imported reference material
Data Channel	Artifact	Core creative outputs
	Metadata	Attributes of objects
	Prompt	User prompts to AI
	AI-generated content	AI-generated material
	Reference material	Uploaded sources like text, images, or audio
	Layout	Spatial arrangement of components
	Performance	Ephemeral actions, e.g. live improvisation
	Iteration	Versions, edits, or branches
	Feedback	Peer or AI feedback
	Annotation	User notes or markup

Table 7.1: The data characteristics of the framework

while general creative practice systems lean on graphical objects coupled with metadata.

Taken together, the modality dimension shows not only what systems record, but also how they choose to represent creative activity. The prominence of text as a cross-domain standard reveals both a pragmatic design choice and a potential constraint: enabling system intelligence while narrowing the expressive modalities that traces can carry forward.

7.4.1.2 Origin

	Code	Examples
Capture Agency	User-Initiated	Quickpose branches created by user [229]
	System-Initiated	Immersive Sampling auto-logging metadata [265]
Context window	Full Trace	Rosetta using full dialogue history in improv [77]
	Localized	Lotus applying edits only to transcript region [28]
	User Selected	CoColor allowing color palletes to be save dto library [137]
Visibility	Visible	VRGit surfacing collaborator edit histories [335]
	Implicit	Creative Arc Negotiation using performer dialogue for robot agent behavior[154]
Function	Creative exploration	WhatIf's branching narratives [204]
	Historical reflection	LyricJam showing creator generated lyrics [297]
	Contextual suggestion	Scones using conversation history for next-turn generation [143]
	Organization	D-Graph collecting and linking ideas [265]
	Reuse and remix	Spellburst allowing branches to be merged [20]
	Cohesion	Improv systems maintaining narrative flow [154]
	Style and intent alignment	StyleFactory aligning to user-ranked preferences [342]
	Comparison	BeyondTheChat showing side-by-side drafts [333]
	Backtracking	PromptPaint slider to roll back generations [283]
	Collaboration	Katika exposing iteration histories to peers [155]
	Transparency	Lumimood surfacing intermediate AI outputs [212]
	Memory	ChoreoCraft supporting quick recall of recorded segments [132]

Table 7.2: The trace awareness aspects of the framework

The origin dimension is about *who* created the data, whether it is authored by the user, contributed by another human, or generated by the system itself. Across our corpus, user-authored data remains the dominant origin, reflecting that most tools still ground traces in the creator's direct activity. For example, drawing inputs in WhenIsATool or edit operations in Lotus demonstrate in-tool origin [28, 165]. At the same time, many systems combine origins. Some rely on hybrid traces produced jointly by user and system, such as Scones, which logs both user text and AI conversation history [143], or BlockAndDetail, where each user stroke triggers a system-generated continuation [242]. Others integrate material from other creators as reference materials, such as ChoreoCraft, which lets users upload music files composed by others to guide dance generation [132], and Frid et al's system which uses example songs as input to generate new songs [107]. These multi-origin traces highlight how creative practice is rarely bounded to one actor or dataset. The origin dimension underscores the importance of specifying origin clearly to clearly delineate whose data is represented in a trace, how sources intersect, and what counts as part of the creative record.

7.4.1.3 Creation Context

The capture context dimension is about *where* the data was created, whether it originates within the tool itself (in-tool) or is brought in from outside (external). This complements the origin code by showing not just who created the data, but also the environment in which it was produced. Most systems in our corpus captured data directly in-tool, such as recording edits, prompts, or generation actions as the creative session unfolded (e.g. [336]). A smaller set drew on externally created material, such as EuterPen, which lets musicians import audio clips as references [46], MoodCubes, and and Immersive Sampling, both of which curate inspirational artifacts collected outside the system. Most video content creation systems such as LAVE and ExpressEdit demonstrate external data that is user origin, specifically previously recorded video footage [283, 302]. These examples show how capture context can range from session-specific activity to external material that can be of any origin. Systems that rely solely on in-tool traces emphasize user-created content, while those that integrate external data expand the creative record to include prior work, cultural resources, or other creators' contributions.

7.4.1.4 Channel

The channel dimension describes the *contribution* that a given data item makes within a trace. Whereas modality captures how data is represented (e.g., text, image, audio), channel captures what *role* that data plays in the creative process. We developed this dimension inductively, as systems often captured very different kinds of information even when using the same modality. For example, text could serve as a prompt, a feedback message, or a metadata label, each representing a distinct channel of contribution. Categorizing data in terms of channel proved an effective way of distinguishing whether traces were centered on artifacts, iterations, annotations, or other contributions to creative work.

Across the corpus, the artifact channel was the most prevalent, encompassing direct creative outputs such as sketches, drafts, or videos which will become part of the final artifact. In creative coding systems like Quickpose and Spellburst, the artifact channel were complemented by

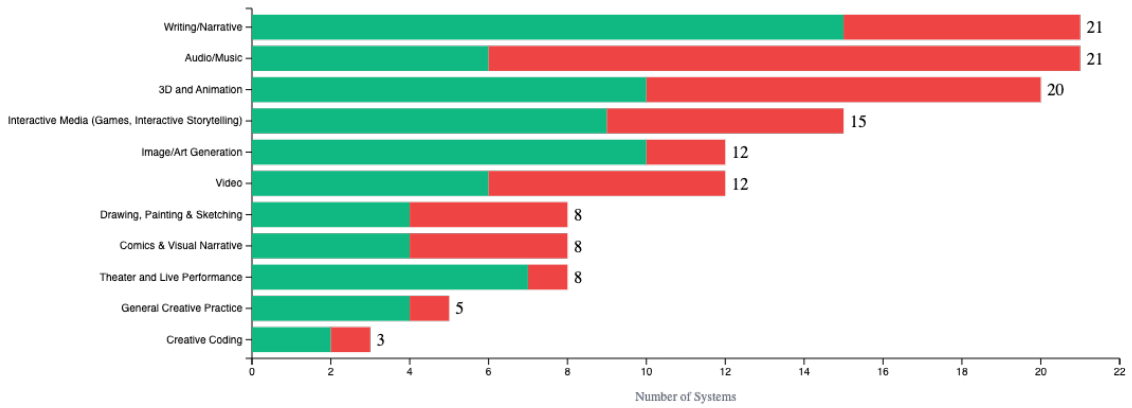


Figure 7.5: Distribution of systems across creative domains, with each bar showing the proportion of trace-aware and non-trace-aware systems (green is trace-aware).

the iteration channel that logged branching edits or runtime versions, emphasizing process over static outputs [20, 229]. By contrast, general creative practice systems such as MoodCubes and Immersive Sampling leaned on reference and metadata channels, where imported inspirational materials and their attributes (e.g., color, type, scale) became the core of the trace and powered contextual suggestion features [153, 265].

Other domains highlight more specialized channels. Theater and performance systems captured ephemeral creative actions, namely dialogue, gestures, or choreography, through the performance channel, logging live acts rather than durable artifacts [154, 197]. In post-GenAI systems, prompt and AI-generated content channels became especially prominent, as in ExpressEdit or StyleFactory, where prompts and generated outputs were central to system behavior [283]. More rarely, we observed annotation channels, where users explicitly documented their work (e.g., TaleStream's labeled tropes [60] or Quickpose's annotation of nodes [229]), or feedback channels, where collaborators or the system itself offered evaluative comments (e.g., Katika for animation feedback, or VISAR for writing critique) [155, 340]. A small subset of systems also logged layout channels, capturing spatial arrangements of elements such as nodes or interface objects, which became meaningful design data in themselves [59].

7.4.2 Trace Awareness

In analyzing the systems, we observed a key design distinction: some tools operate on trace data only as *static snapshots*, while others make use of traces *temporally*. We describe the latter as *trace-aware* systems, which form the second key dimension of our framework. Data characteristics apply to any system, since they describe the raw material that is captured. What makes a system trace-aware is that it treats this material not as isolated records, but as *true* activity traces, i.e. data unfolding *over time* that can shape system behavior. A trace-aware system leverages this temporality both implicitly, by adapting to prior edits, prompts, or choices, or explicitly, by surfacing past states for reuse, comparison, or branching (See visibility code below). By contrast, non–trace-aware systems may store data, but do not interpret them as part of a temporal

record; each action is treated in isolation from what came before, and the system is agnostic to how the process data is yielded temporally. For example, a writing tool that merely uses the latest saved version of a draft to generate content suggestions is not trace-aware, while one that looks at a sequence of edits to suggest style improvements or generate alternative continuations is trace-aware.

Trace-aware systems account for 77 of the 133 systems in our corpus (57%) (Figure 7.5). This distinction emerged inductively from our coding, as we found many prototypes that truly modeled or surfaced traces to inform system behavior. In what follows, we treat trace awareness as a multi-dimensional aspect of system design, unpacked through the categories of capture agency, context window, visibility, and function.

7.4.2.1 Capture Agency

Capture Agency considers *who* initiates the capture of trace data. Most systems log data when a user creates, edits, or saves (user-initiated capture), while a smaller set capture data automatically in the background (system-initiated capture).

Patterns in our corpus reflect the traditions of their domains. In creative coding, systems such as Quickpose and Spellburst both rely on user-initiated capture, where traces are logged when the user creates a new branch or runs code [20, 229]. These systems' features are built on the user deciding when the right time is to capture trace data.

Some systems introduce more hybrid cases, especially ones reliant on LLM chat features. CharacterChat, for example, allows users to pin chat messages, indicating user-initiated agency, whilst the system automatically lists relevant character attributes that are revealed during the chat [246].

Overall, capture agency highlights an important fault line in trace-aware design. Most systems place responsibility on users to initiate the capture of their traces, reflecting a norm where creative actions drive the record. Far fewer systems capture data automatically in the background, which suggests an opportunity for future designs to balance user-led and system-led capture in ways that enrich traces without overwhelming creators.

7.4.2.2 Context Window

The second dimension of trace awareness is *Context Window*, which describes the temporal *scope* of data that a system uses when engaging with a trace. Some systems draw on the full record of activity, others rely on a localized slice of recent actions, and a smaller set allow users to define the scope themselves. Context window is important because it determines how much of a creator's past activity is available to shape future interactions. In conjunction with the channel dimension, it tells us a great deal about the richness of a trace and how it can be leveraged in practice.

Performance-oriented systems often depend on continuous, session-long windows. Improv and acting systems such as Rosetta and Creative Arc Negotiation rely on dialogue histories across the full performance to maintain cohesion and narrative flow [77, 203]. In video editing, Lotus takes the opposite approach by applying a localized window: transcript edits propagate only within the affected region rather than across the full sequence [28]. In terms of user-specified

context windows, the most prevalent example is the ”save to library” feature that exists in many systems as a form of light-weight history keeping [38, 111, 306]. CoColor is one such system, as it allows users to save generated color palletes to a library for later use [137]. User-set context windows are limited only to library-saving features, which illustrates a gap in how variable context windows can be more meaningfully leveraged to inform trace-aware system behaviors. This suggests a design opportunity; giving creators more control over how much of their past work is brought into play could open new possibilities for reflection, adaptation, and reuse beyond just saving libraries of content.

Across domains, we found most systems default to either full or localized windows, with none allowing for user-steered scopes that functioned as anything more than a library of saved artifacts.

7.4.2.3 Visibility

Visibility captures if a system *surfaces* traces directly to creators and/or leverages them *internally* without exposing them. Visible traces can be presented as part of lists of saved content, galleries of generated alternatives, timelines of edits, or collections of inspirational material. However, while most trace-surfacing behaviors in the data are dedicated to saving whole artifacts, AI-generated content, or reference material, trace surfacing can also serve as a form of *material interaction*, giving users something tangible to reflect on, manipulate, or repurpose [229, 267]. By contrast, implicit trace usage shape outcomes behind the scenes. These systems adapt behavior based on past actions, propagate changes across artifacts, or align with user style without necessarily making the underlying data visible.

This distinction is central because it fundamentally changes the role traces play in creative practice. When surfaced, traces become a creative material in their own right; they can support reflection, comparison, reuse, and remix (see Trace Function dimension below). In contrast, implicit traces often function as *invisible scaffolding*. For example, Creative Arc Negotiation uses performance history to guide narrative direction in improvisational theater, but participants never see the trace itself [78].

Domain traditions contribute to how visibility or implicit usage are handled. Writing tools frequently emphasize visibility, surfacing drafts, annotations, or conversation histories to give users direct control over revision and reflection (e.g. [257, 333]). By contrast, performance-oriented systems more often privilege implicit use, because surfacing traces during live improvisation would disrupt flow or create cognitive load. Here, the system uses traces internally to maintain cohesion or style continuity. Creative coding tools like Quickpose and Spellburst demonstrate a third case: visibility as a rich form of material interaction [20, 229]. Here, traces are deliberately surfaced as branching versions or annotated layouts, giving creators direct, manipulable histories. Some systems surface all or part of the trace while also using trace data implicitly. Meanwhile, conversational interfaces such as Scones or MVPrompt embody both paradigms simultaneously: the chat log is a visible trace for users to scroll and edit, but also an implicit input for the LLM, shaping output without requiring explicit user intervention [143, 167].

Both forms of visibility can coexist, but they represent different design implications. Systems that emphasize visibility position traces as a resource for creators, potentially giving them agency

over how process history is interpreted and reused. Systems that emphasize implicit use position traces as infrastructure for adaptation, making the system itself the primary interpreter of process data.The balance between the two reflects not just technical feasibility, but deeper implications about how traces should serve creators, systems, or both.

Across our corpus, trace surfacing was more common, though often in lightweight forms such as conversation histories or saved libraries. Implicit trace usage was comparatively narrower, typically limited to artifact or prompt histories rather than drawing on richer, cross-channel or cross-modal traces to drive system behavior.

Notably, audio/music systems exhibited the lowest levels of trace awareness overall, and when traces were surfaced, they were typically limited to basic historical reflection (e.g., showing generated lyrics or previously played clips). Only SHARP, a live coding system, offered deeper surfacing by letting users track iterations as part of improvisation. This stands in sharp contrast to creative coding, where visibility was used to enable powerful material interaction, and to domains like image/art genration, where implicit traces underpinned predictive features. Figure 7.6 illustrates the trace visibility and implicit usage of music/audio systems, writing, and live performance systems.

7.4.2.4 Function

Function describes what systems actually *enable* with traces once they exist. Whereas visibility describes whether traces are surfaced or used implicitly, function describes the kinds of affordances that become possible through that design choice. These functions support different types of trace affordances, from users simply seeing their data, to seeing but not directly manipulating it, to actively working with traces as creative material.

Across our corpus, we identified twelve main functions (see Table 7.2). Taken together, the twelve functions cluster into three broad interpretive orientations that describe how systems make use of traces in practice: supporting retrospective understanding, enabling material manipulation, and guiding future creative action. These orientations are not additional coded categories. Rather, they are analytical groupings that help illustrate how the functions relate to one another.

Retrospective functions help creators look back on what has been done. These include *historical reflection* [111, 137, 342], in which users revisit their trace data to understand how their process evolved. In *memory recall* [132], the system surfaces prior decisions or reference to support users in quickly recalling information that may have otherwise been forgotten. *comparison* lets users compare previous alternatives side by side, a common pattern in tools for image, video, and music creation [131, 275]. *backtracking* allows users to return to earlier states, giving them a mechanism to undo their most recent creative decisions [326]. These functions are often lightweight but foundational, as they treat traces as *evidence* of past work and make that evidence inspectable in ways that support understanding and decision making.

Manipulation functions treat traces as actionable material. Systems enabling *reuse and remix* of prior materials position sketches, code snippets, or earlier artifact iterations as building blocks for new ideas [20, 229]. *Organization* brings structure to accumulated traces by enabling sorting, grouping, spatial layout, or clustering of trace data [301]. Finally, *collaboration* is supported by extending these ideas through shared creative contexts, where traces become a key manipulated material in joint creative work [155, 335]. When traces become manipulable elements that

can be combined, edited, reinterpreted, or reframed, they then sit at the heart of material interaction. Creative coding tools, version control interfaces, and ideation canvases exemplify this orientation, since their interaction models depend on treating traces as resources that creators can directly engage with.

Propsective functions use traces to shape what comes next and are focused on supporting future creative moves. These include *creative exploration* [204], where prior states or suggestions act as springboards for creators to explore new alternatives, and *contextual suggestion*, where the system uses traces as context to propose likely next steps [221]. Other systems emphasize maintaining *cohesion*, drawing on traces to maintain consistency when generating new narratives [154, 203]. Some go further by aligning system behavior with user *style and intent* [160], using accumulated traces to personalize future generation. Finally, *transparency* lets creators inspect intermediate results or model reasoning, providing users with insight into how AI generations were produced [212]. Transparency was especially notable in systems like PromptPaint and LumiMood, which let users inspect intermediate image generations, an intriguing trace function that helps users understand LLM-driven functionality [62, 212].

Taken together across these orientations, function is where material interaction and implicit trace-aware system behavior come together (Figure 7.8). Systems that surface traces but do not offer manipulation tend to emphasize retrospective functions [111, 137]. Alternatively, systems which use traces to support manipulation functions treat traces as pliable material, which opens up richer affordances such as remix or exploration [20, 265, 336]. Systems that use traces implicitly, on the other hand, lean more towards prospective functions such as cohesion or contextual suggestion, since these rely on internal interpretation.

Across domains, we observed uneven coverage of these functions (Figure 7.7). Music tools rarely went beyond organization and creative exploration, with SHARP as a rare exception [197]. Video systems, despite their temporal and stylistic richness, offered little in the way of style or intent alignment, relying instead on reflection or transcript-driven and prompt-based suggestions to create content. Creative coding systems provided some of the deepest manipulation affordances, but in narrow implicit forms rooted in version-control paradigms. These patterns suggest that traces are still underutilized as a mechanism for future-oriented, prospective creative support. Expanding the functional role of traces presents an opportunity to move beyond documenting the past, and instead move toward systems that fully engage with the rich trajectory of creative work.

7.5 Discussion

In this discussion, we step back from individual findings from our framework to consider what it means to treat traces as a design material, where the field might go next, and what limitations or risks need to be addressed. By combining these perspectives, we aim to situate our framework not only as an analytical tool but also as a generative tool and starting point for conversations about the role of activity traces in creativity-oriented systems.

7.5.1 Traces as a Design Material

While we use our framework for analytical purposes, our aim was to shed light on the fact that activity traces are not just data used in system interaction. We argue that these encodings serve as the system's working model of how creators operate, how their processes unfold, and how their artifacts evolve over time. Capturing trace data is only the first step. Once a trace exists, the challenge becomes how to make sense of it, which parts to foreground, and how to incorporate it into meaningful system features [162].

In practice, many systems we reviewed, including those we classified as trace-aware, leverage traces in a limited and ad hoc fashion. Systems often focus on artifact channel-related data, with trace visibility often limited to "save to library" features that nominally preserve history but rarely go beyond lightweight visibility and historical reflection. In terms of implicit usage, conversation history with LLM chatbots function as a de facto trace, which offloads trace behavior to third-party infrastructures and treats the full context window as a homogenous entity. Even within specific creative domains such as video editing, we see tools providing only narrow forms of contextual suggestion and rarely using traces for style or intent alignment [302]. Our analysis suggests that while features across domains recognize the value of traces, they stop short of fully leveraging them.

Our framework calls for a shift in orientation. Rather than treating traces as incidental, we argue for treating them as a design material that can be intentionally captured and implemented in service of creative practice. This requires asking not only what traces are captured by creativity tools, but also when, how, and for what purpose. Treating traces as a design material introduces a bottom-up approach to system design, where rather than determining trace data usage after features have been defined, we instead begin by examining how traces can help align with the user's goals.

Even in cases where traces are not directly used as design materials, we believe researchers and designers should make explicit what forms of creator data their systems rely on, and how those data influence interaction. This perspective also allows us to situate our findings within and across prior work. Shneiderman's principle of history-keeping has long emphasized the value of retaining past states, but our contribution extends this principle by articulating how traces can drive a wide range of system behaviors. Our analysis complements earlier work that focus on creativity support [104, 176], and specific aspects of creativity like version control [267]. We treat our framework as an additional lens for understanding system design, supporting creator processes, and context-aware adaptation. Rather than constraining creators to a tool's predefined workflow, trace-aware features can be designed to align with the evolving contours of a creator's own process. It provides not only a way to expand the vocabulary of system features, but also a mechanism for thinking more carefully about how systems model, understand, and support creative processes.

7.5.2 Leveraging Traces In The Future

If traces do have this potential as a design material, how do we go about realizing it? We propose several directions that build on the gaps and opportunities surfaced in analysis and formulation of the framework. These are not meant to be exhaustive, but they outline areas where both

researchers and system designers can advance the state of creativity tools.

7.5.2.1 Personalized Creator Profiles

One striking gap in the systems we analyzed is the lack of serious attention to creator profiling or designer modeling. Few systems attempt to build models of creators' styles, preferences, or working patterns. This is a missed opportunity, particularly for tools that already frame their contribution around personalization or intent alignment. Alvarez et al. propose one possible approach would be to cluster design trajectories into personas or "designer models", predicting next steps based on archetypical styles rather than individual edits [15]. Such models would allow systems to differentiate between creators in meaningful ways and respond to the fact that people often take different design actions in the same state. Profiling can serve not only as a mechanism for personalizing creative process, but also differentiation across users, which could expand possibilities for tailoring system responses [169]. Importantly, this is consistent with Schneiderman's long-standing principle that creativity tools should support "many paths and many styles" [255]. Trace-aware design could provide the raw material for doing so in ways that respect the diversity of creative practice.

7.5.2.2 Moving Beyond Project-Based Traces

Another limitation we observed is the overwhelming tendency to capture and use traces strictly within current projects, with the only exception being mechanisms to upload previously created material. This project-centric view makes sense in the context of bounded prototypes, but it constrains how tools can model creative practice over time. What would it mean for a tool to incorporate long-term trace histories that span multiple projects, or to allow creators to navigate their processes across months or years? Doing so could open the door to supporting reflection, reuse, and recombination in ways that current project-bound systems cannot, and can potentially support creator profiles as discussed above. Sterman et al [267] argued that considering versions across project boundaries and timescales enables creators to reflect on their growth and return to ideas that might otherwise be lost [267, 268]. Li et al.'s work on horizontal movement across platforms also points toward the potential of traces that persist and travel not just between projects, but between platforms as well. Our framework further suggests that future systems could let creators dynamically set their own context windows or parameters for what traces should be considered relevant. This kind of flexibility would mitigate risks of opacity, but help expand opportunities for creator control over their data in the context of long-term trace capture.

7.5.2.3 Untangling User and System Processes

Our framework intentionally folded together user- and system-generated data, treating both as resources appropriated during creative practice. Yet future work could benefit from making the distinction explicit. Separating these forms of process data, potentially in the form user activity traces versus system activity traces could make visible how human and AI traces intersect, diverge, and influence one another in mixed-initiative contexts. This opens an opportunity to

study what can be called ”interrelated creative evolution,” where the trajectories of human activity and system intervention shape one another over time. Such a perspective could also inform evaluations of human–AI collaboration, since it foregrounds not just outputs but the traces of interaction that led to them, and our framework already demonstrates that some systems use traces to surface intermediate generations from AI. Bremers et al's work on Situated framework emphasizes that a key task for co-creative systems is to understand not just how to intervene but also when, and how much [40]. A trace-oriented lens could help us answer those questions by providing empirical grounding for how interventions ripple through creative processes.

7.5.2.4 Cross-Domain Opportunities

Finally, we see promise in developing activity trace theory as a cross-domain design language. Our framework allowed us to compare systems across domains, and we believe this comparative lens is itself a contribution. A trace-oriented vocabulary enables dialogue not only among HCI researchers but also among creators and designers working in very different media or research traditions [104]. For example, a system that provides contextual suggestions to musicians based on text traces has surprising parallels to one that supports visual storytellers through organization of image traces. Putting these systems side by side makes visible both the shared design logic and the domain-specific nuances. This opens up possibilities for new forms of interdisciplinary tool design that are anchored in a common trace-aware language. Specific opportunities include moving beyond artifact-centric version control to support deep reuse and remix, exploring style and intent alignment in video creation, and expanding the visible and implicit use of traces in music and audio tools. Taken together, these directions sketch an activity trace research agenda for the field. By moving beyond project boundaries, engaging with creator profiling, disentangling user and system traces, refining our frameworks, and pursuing cross-domain applications, HCI researchers and system designers can move towards realizing the potential of traces as a design material.

7.5.3 Challenges and Risks Of Trace Usage

So far we have highlighted the potential of activity traces to serve as a design material and outlined future directions for realizing that potential. Yet there are also real challenges and risks involved in adopting trace-aware design. Traces are not neutral; they are encodings of process data that carry assumptions, limitations, and costs. In this section we reflect on several critiques and challenges that emerged from our work, including normative ground concerns, technical constraints, and ethical considerations around creator data usage.

7.5.3.1 Reinforcing Normative Ground

Li et al. critically established that creativity support tools impose a ”normative ground” on creators, shaping how they think and act within a system. While we argued earlier that traces can help align tools more closely with the contours of creators' processes, there remains the risk that trace-aware systems may encode the assumptions of their designers in ways that limit rather than expand creative practice. It is important to distinguish between traces positively being used to

adequately support the norms of a creative domain, which can be productive and contextually appropriate, from traces usage that unintentionally encodes the designer's assumptions about the user's creative process. In our review we saw many examples of the former, namely improvisational systems that used text-heavy implicit trace usage to support real-time improvisation, video tools which drew on external material to contextualize edits, and version control systems which surfaced variations of an artifact to enable reuse. These reflect domain-specific values rather than misplaced constraints. The danger lies in cases where, in an attempt to approach system design from a trace-aware perspective, traces are used in a manner which reinforces the normative ground on the creator rather than support the creator's inherent process. Addressing this risk requires intentional design practices where we make explicit what assumptions are embedded in trace usage, giving creators greater visibility into how their trace data is captured and used, and allowing them to control or contest any trace usage when they do not align with their practice.

7.5.3.2 Limits of Trace Capture

Another limitation is conceptual; how much can traces feasibly capture about a creative process? Creative processes are messy, evolving, and often resistant to formalization. Kreminski et al's work on intent elicitation suggests that creator intent is often incomplete, which indicates that trace elements can mean different things at different times. The idea that an entire creative process can be operationalized in data is unrealistic, and treating traces as a comprehensive record risks oversimplifying the lived complexity of creative practice.

7.5.3.3 Risks of fixation and overuse

Trace-aware design is not always desirable, especially when the system cannot account for ambiguity, uncertainty, or shifts in meaning over time. Traces also carry risks when used implicitly. While systems now have the capacity to collect vast amounts of data, they can also encourage fixation by continually surfacing past states or making inaccurate suggestions. For some creators, revisiting old processes may even be harmful, reinforcing unproductive habits or reactivating negative associations with their work. The risks with trace-aware design are also computational in nature because capturing, storing, and reasoning over traces at scale can be expensive, and systems with large context windows often struggle to distinguish relevant information from noise. In this regard, expanding the scope of traces without clear design strategies risks overwhelming both systems and users.

7.5.3.4 Trace Ethics

Perhaps the most pressing critique concerns ethics. Our review revealed that most systems do not make explicit what forms of creator data they are capturing or how those data are being used. This lack of transparency, if carried forward in a trace-aware design future, creates risks for consent, provenance, and autonomy. Palani et al.'s value framework for creativity support tools is instructive here with regards to aligning systems with creator values [216]. We extend this notion to trace-aware design and advocate for systems that respect creator agency, provide visibility into how data is used, and allow for meaningful control over system state. Without

such safeguards, trace-aware design risks creating opaque feedback loops where traces shape interaction in ways that are invisible to the user.

Taken together, these critiques remind us that treating traces as a design material is not a straightforward path. Traces offer opportunities for richer alignment between tools and creative processes, but they also risk reinforcing norms, oversimplifying complexity, fostering fixation, or undermining transparency. A more nuanced conversation is needed in HCI to ensure that trace-aware systems are designed ethically and intentionally, with clear consideration of when traces should be captured, how they should be used, and whether they should be surfaced at all.

7.6 Limitations and Future Work

We sought to create the first living framework for understanding activity traces in creativity tool design. While our work contributes both a framework and a preliminary design language to analyze and compare trace usage in systems, there are important limitations to our approach that must be acknowledged, as well as promising directions for future research.

7.6.1 Refining The Framework

We have been deliberate in presenting our framework as a living structure and acknowledge that there are areas requiring further study [215]. Some dimensions proved difficult to code cleanly and were excluded from the final version. For example, distinguishing between data used to create content versus data used to express intent, or clarifying usage agency (who initiates trace use), remain unresolved challenges. Chung et al's review of users, roles, and interactions offers one parallel for thinking about direct and indirect inputs, which could help sharpen the content–intent distinction.

Our framework also collapses too much nuance into the "channels" dimension. Richer modeling of user actions could benefit from traditions such as Linkography or Workflow Graphs, which capture creative processes in more fine-grained ways. Another important area for refinement lies in linking across modalities and channels. The concept of abstraction [176] and ideas of orchestration [215] provide useful starting points. Abstractions allow for flexible representations of data fidelity [267], with low-fidelity trace data at earlier stages of the process and high-fidelity articulation in a later stage of refinement.

7.6.2 Methodological Challenges

Methodologically, our scope has a few limitations. We focused our analysis on creativity support tools related to creative media, such as art, video, audio, storytelling, and games. This choice excluded a range of creative domains from our scope that are of interest, including CAD, UX, and programming-related creativity tools. Activity traces are deeply relevant in these domains and we believe that our framework can be extended to them. We see the analysis of these kinds of systems from a traces perspective as an immediate opportunity for future work. Similarly, our corpus selection excluded several venues that would have enriched our analysis, most notably

CSCW. The relative absence of collaboration-related insights in our findings can be partly attributed to this gap. Expanding the corpus to include CSCW and related venues would provide a clearer understanding of how trace-aware systems function in creative-collaborative contexts.

Our coding process also involved trade-offs. We coded the trace data characteristics and awareness patterns associated with each system in our corpus all at once rather than focusing on individual characteristics in separate passes. Breadth also came at the cost of depth. Coding across 133 systems enabled comparative analysis, but necessarily sacrificed thick descriptions of individual systems in context. Moreover, because our framework is explicitly trace-oriented, systems that did not articulate trace usage risk appearing less "rich" than those that foregrounded it.

In addition, our temporal scope was limited to the past five years of HCI conferences, which foregrounds contemporary systems but under-represents longer trajectories of trace usage. This is especially important as our corpus skewed heavily towards post generative AI papers, which limits our ability to truly evaluate the difference between pre and post genAI trace usage.

Looking ahead, we see several directions for future research. Fleshing out the framework in accordance with the future directions discussed previously is necessary to truly understand how traces are manifesting in existing systems and for the establishment of a comprehensive design language. A deeper engagement with ethical and sociotechnical issues associated with creator data, specifically with regards to ownership, consent, and power in trace capture, is a crucial step to legitimize the responsible use of traces as a design material.

Creator-facing studies are also needed to understand how creative process can be modeled in creativity tools via activity traces. In particular, there is a need to complement our framework-based coding with empirical studies of creators to assess whether the coded dimensions align with lived creative practices. Together, these extensions will help refine and formalize trace-aware design as a concept, enabling both critical reflection and practical system-building.

7.6.3 Applying The Findings to Commercial Products

Our work here is motivated by a broader goal of improving creativity tool design, both within HCI research and in widely used commercial products. Many of the critiques that motivate this work apply equally to commercial tools [176, 194], including tools such as Premiere Pro, Figma, and PowerPoint, which play a central role in contemporary creative practice. Trace awareness as a design material is therefore relevant far beyond the systems published in HCI literature. However, for the purposes of this paper, we did not include commercial products in our corpus because our analysis depends on access to documented system rationale, data flows, and design decisions. Research papers provide this level of transparency, whereas commercial tools are proprietary, internally complex, and continually evolving, making it difficult to apply our coding approach with comparable rigor.

Even so, many mainstream tools reveal how the framework can extend beyond research prototypes. PowerPoint, for instance, retains an extensive context window to support large undo histories, version snapshots, and repeated design actions, yet it does not leverage creators' traces to inform future layout, narrative flow, or stylistic decisions. Similarly, Figma provides examples of trace visibility to support collaboration, such as collaborative cursors, comment histories, and multi-branch version control, but offers limited trace-aware adaptation that leverages these his-

tories to provide insights on each collaborator's preferred working style. Finally, Premiere Pro captures rich, cross-modal traces of edits, markers, transcript annotations, and timeline manipulations across an entire project, yet does not support cross-project trace functions such as style and intent alignment or creative exploration. These examples illustrate that the framework's existing dimensions can map meaningfully onto commercial tools, while also highlighting substantial opportunities for trace-aware design in widely used creative software.

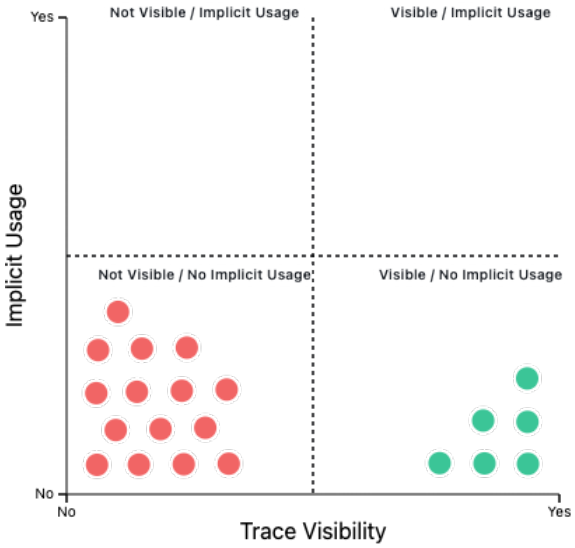
Beyond the limitation of transparency, applying our work to production software raises the potential for additional challenges distinct from those in systems literature. Commercial tools operate at far greater scales of data and collaboration, often spanning years of work across multiple projects, teams, and online user communities. Their feature sets are large and intertwined, making it difficult to isolate the data items or trace interactions that our coding scheme analyzes. Many aspects of trace awareness in commercial products occur behind the scenes, such as automated syncing, content-aware editing, or cloud infrastructure, where the underlying processes are not visible to researchers. These conditions pose new questions about how trace-aware collaboration operates at scale, what other trace-aware functions exist, and how systems implicitly or explicitly reason about creator histories.

Exploring these questions will require different approaches beyond our corpus analysis, including reverse engineering, interviews with industry designers, or direct collaborations with industry practitioners that provide access to internal system insights. We see this as an important direction for future work, particularly as commercial tools increasingly integrate generative AI features that rely on creator data. Understanding how traces are captured and leveraged in these contexts can inform more transparent and creator-aligned design. Our framework offers a structured vocabulary for beginning this analysis, even though a full comparison with commercial products lies beyond the scope of the present review.

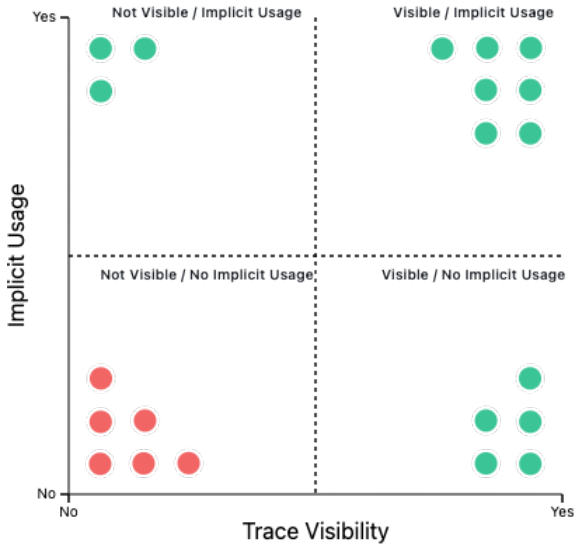
7.7 Conclusion

Creativity tools shape not just what creators make, but how they make it. In this paper, we utilize the concept of *creative activity traces*, i.e. the process-related data generated during creative work as a lens for examining creativity tool design. Through our analysis of 133 recent creativity tools, we developed the first Living Framework For Trace Awareness which can describe trace-related phenomena and compare it across systems and domains. Our work argues that traces are not incidental byproducts, but core materials that systems can capture, surface, and leverage to power functionality. By treating traces as design material, we can begin to see how tools implicitly encode models of creative practice, and how those models in turn structure opportunities for creators.

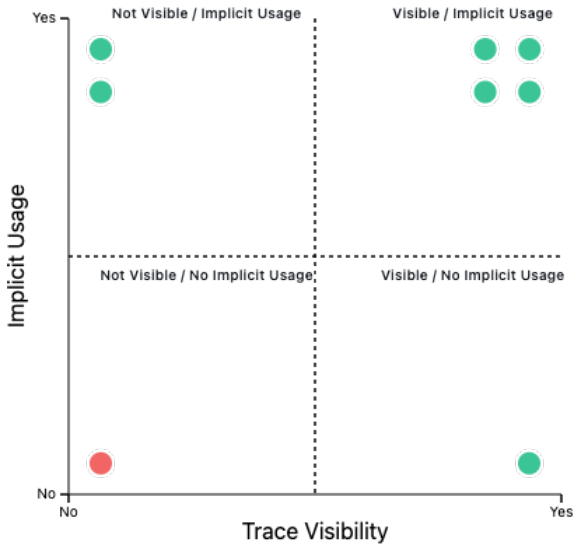
In advocating for trace awareness, we call for creativity tools that do more than passively collect creator data. They should leverage traces natively and responsibly to supporting creators in ways that respect their processes, foreground their agency, and open new possibilities for creative practice.



Trace visibility and implicit usage in Audio/Music systems



Trace visibility and implicit usage in Writing systems



Trace visibility and implicit usage in Live Performance systems

Figure 7.6: Quadrant visualizations for three creative domains showing how they each surface or implicitly use traces. Red dots indicate non trace-aware systems

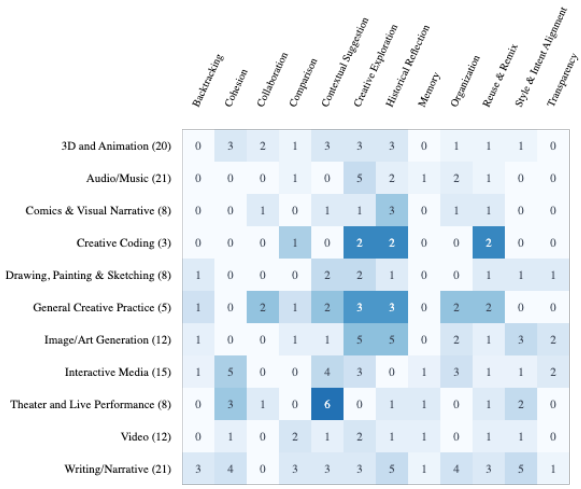


Figure 7.7: Distribution of trace functions across each creative domains.

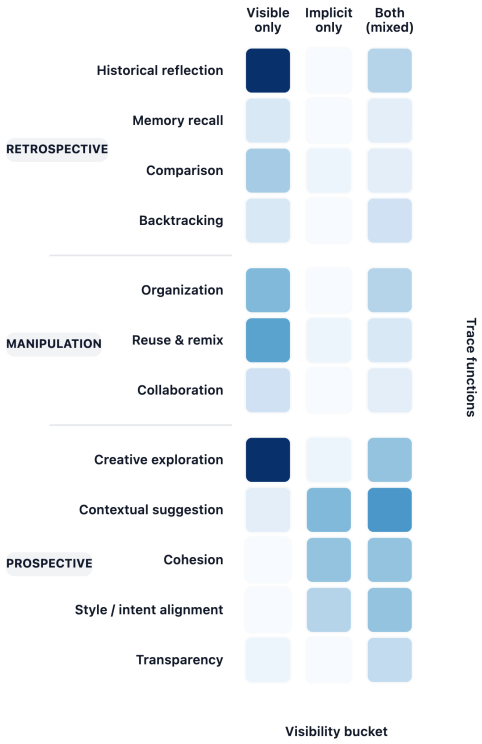


Figure 7.8: Trace visibility and implicit usage by function

Part III

Exploring Traces as a Design Material

Chapter 8
Exploring Traces In Longitudinal Video Content Creation Practices

8.1 Introduction

In my previous work, I introduced and formalized creative activity traces as a construct for understanding how tools capture, represent, and leverage the process data generated during creative work. Through a systematic review of 133 creativity tools, I described how researchers have leveraged traces to date and synthesized this engagement into the Living Framework for Trace Awareness. Central to this work, and to this dissertation, is the argument that traces should be treated as a design material rather than incidental data, and that they require *deliberate engagement* from tool developers and researchers to align with creator needs. How to effectively execute a traces-as-material approach, however, has not yet been studied, nor have the "material properties" of traces been closely examined [238]. Towards this goal, this chapter is a targeted exploration of what it means to use traces in this manner.

However, meaningfully exploring the use of traces as a design material requires a creative practice to investigate. Choosing such a problem is also an opportunity to support creators with the underexplored avenues in the design space laid out in my previous work. Specifically, trace-aware systems in HCI overwhelmingly operate at the scope of a single project, with little existing work investigating the value of trace awareness over time. Cross-project, longitudinal trace awareness is relevant for creators across many domains, as they have both a *portfolio of work* and a *creative identity* that extends beyond the walls of a single project.

Video content creation is inherently portfolio-driven, as successful creators need to sustain a prolific output of video content over time. To succeed in the long term, video creators must stand out in the saturated landscape of social media platforms such as YouTube, TikTok, and Instagram [270, 287]. However, as content creation experts note, developing a recognizable and consistent *personal brand* is essential not only for visibility, but also for authenticity and audience engagement. Personal branding gives creators a way to define what makes their work distinct, which helps them communicate a coherent identity across videos rather than reinventing themselves with every upload. Video content creators, in this sense, do not work in discrete, self-contained videos. They contribute to a continually evolving body of work over months and years, where each video is influenced by, and in turn influences, the others.

A video content creator's personal brand can be understood as the identity that communicates who a creator is, what values they represent, and how they express these through aesthetic and stylistic choices [270]. This brand becomes recognizable through a creator's repeated creative decisions, such as a consistent color palette or thumbnail style, a familiar intro sequence, a recurring music motif, or an editing rhythm that audiences come to associate with that creator [198, 270, 286]. These recurring stylistic choices, which I refer to as personal brand *markers*, serve as signals of authenticity and creative intent, helping audiences immediately identify a creator's work. Personal brand, then, is not simply about marketing. It is the mechanism through

which creators craft a recognizable creative voice.

However, discovering and maintaining a personal brand is a major documented challenge for creators [166, 286]. For novice creators, the process often begins with extensive experimentation or imitation of existing creators to find what resonates with them and their intended audience [270, 286, 287]. Moreover, once a recognizable brand begins to take shape, maintaining it requires a delicate balance between consistency and evolution: too much unintentional variation can confuse audiences and result in an incoherent body of work, while too little intentional variation can stifle creators' growth and self-expression [270, 286, 287].

Most critically, current tools offer almost no support for managing this ongoing negotiation from one project to the next, adding friction to an already demanding creative workload [166]. Creators often rely on their memory to track where stylistic assets are stored in their file systems, or they build ad hoc libraries of intros, fonts, and color schemes to maintain coherence between videos. While these improvised strategies can preserve short-term consistency, they are tedious to maintain, offer little visibility into how a creator's style evolves, and collapse at a longitudinal scale. As creators juggle multiple projects, audience engagement, and their tools, a personal brand becomes fragile; it is a moving target that must constantly be reconstructed from past work rather than being natively supported as a living, traceable part of the creative process.

A trace-aware solution has the potential to address these concerns. This gap aligns directly with the missed opportunities my previous work identified. Existing video content creation tools offer limited trace-aware capabilities, with current features relegating trace-aware behavior to library-saving features, AI chat histories, or revision mechanisms [148, 282, 303]. Few tools support long-term or cross-project trace use, and almost none help creators maintain or evolve a personalized creative profile [270]. Personal brand management is thus a well-articulated creative challenge for creators, and an instance of the cross-project trace awareness my prior work found underexplored.

To investigate how a trace-aware approach can help video content creators build, evaluate, and evolve their personal brand across projects, this chapter is guided by the following research questions:

- What stylistic elements do creators treat as part of their personal brand during post-production?
- How do creators create, manage, and reuse these elements within and across their projects?
- Where do existing tools fail to support creators in cultivating and maintaining personal brand?

Toward answering these questions, I conducted a trace mapping study with seven video content creators, combining semi-structured interviews with walkthroughs of their video content creation and cross-project practice. The study characterizes personal brand markers as trace data that video creators produce, examines the management strategies employed to deal with their traces across projects, and addresses the challenges associated with these strategies at a portfolio level. From this work, I derived a four-stage model of the traces lifecycle for brand markers: Genesis (how a brand marker is created); Elevation (how it is moved to a cross-project scope); Maintenance (how it is interacted with across video projects); and Meaning-Making (how it accumulates with other traces to produce interpretive opportunities for creators). This model serves as both a structured first account of how traces move through a creator's practice, and an instrument for asking more pointed questions about the traces themselves.

With this model in hand, I turned to a deliberate study of the material behavior of traces. I conducted a design and development phase that yielded two *inspirational bits* [277], which are small, quick, fully working but deliberately partial prototypes, in this case designed specifically to understand traces as a material.

The findings from this project contribute both to the study of longitudinal trace awareness in video content creation and to the broader argument calling for traces to be treated as a design material. The inspirational bits reframed Elevation as a prediction made under uncertainty, as the evidence for carrying a marker across projects only arrives in future work. At the other end of the lifecycle, the Meaning-Making stage emerges from the aggregate behavior of accumulated traces. On a longitudinal timescale, this makes it difficult to isolate the effect of a single trace on a creator's evolving practice. This suggests that trace-aware design at cross-project scope cannot proceed purely by categorizing trace characteristics individually, as the Living Framework currently does. Furthermore, though this work was scoped around brand markers, the walkthroughs revealed that participants routinely worked with other trace data, such as scripts, notes, and inspirational materials, which possess different material characteristics. Participants did not deliberately elevate this trace data beyond their current project and saw little clear value in doing so, which suggests that unlocking its cross-project value will require further work on how systems, rather than creators, might elevate it. Taken together with the inspirational bits, this points to trace data coming in differing *grains*, and to its accumulation over time forming a *texture*, and it is in this texture that identity and brand-influencing work takes place.

These findings set the stage for the next phase of creative activity trace research and call for (1) refinements to the Living Framework; (2) better modeling of the relationships among trace characteristics and system behavior; (3) more formal research methods for engaging with traces deliberately throughout the research process; and (4) expanded design opportunities in creativity tools. In concluding my PhD with this project, I demonstrate what deliberate engagement with traces as a design material looks like when carried through a real creative domain, and identify questions that a broader trace-aware research agenda will need to address.

8.2 Background

This chapter takes on two goals in service of understanding how to build better trace-aware tools. The first is to understand how personal brand is created, maintained, and changed in video content creation, a creative practice that unfolds across projects and an underexplored avenue in the design space my prior work laid out. The second is to understand what it means to treat traces as a design material.

Below, I begin by describing personal brand in video content creation, showing that while it has become a central concern for digital creators and is widely discussed in creator communities, HCI has engaged with it only obliquely, and existing tool support treats brand-relevant data as static artifacts rather than as evolving traces of practice. I then turn to the materials literature in HCI to frame how traces can be worked as a design material.

8.2.1 Personal Brand in Video Content Creation

Personal brand has become a defining concern for digital creators [166], but the concept has received little explicit attention in HCI research. Most of what is known about how creators construct and maintain a personal brand comes from industry-facing blog posts, creator guides, professional media commentary, and content creator online communities, all of which emphasize the importance of consistent tone, aesthetic, and values across videos [198, 270, 286]. These sources neither provide empirical insight into how creators themselves conceptualize and manage their brand in practice, nor offer a clear picture of which brand markers creators consider most essential to their identity.

HCI research has engaged more directly with the related question of personal style, though almost entirely at the level of finished artifacts. Lin et al.'s PseudoClient system computationally modeled a designer's personal graphic-design style from a small set of examples, enabling generative systems to learn and reproduce stylistic preferences [178]. Zhou et al.'s StyleFactory introduced an interactive workflow for aligning AI-generated images with a user's aesthetic preferences by ranking examples according to "style strength" [342]. Both systems demonstrate that computational models can recognize and replicate aspects of personal style, but they derive style from finalized or near-final artifacts rather than from the creative activity traces that produced them. As a result, they capture static snapshots of style expression rather than its temporal evolution or iterative development through practice. This orientation toward artifact-level style is characteristic of the broader HCI landscape for personal-style tools, and it leaves personal brand, which is by nature longitudinal and cross-project, without adequate representation.

A trace-aware approach offers a way to address this limitation. By leveraging the process-level data already embedded within creative workflows, trace-aware systems can model how stylistic identity emerges, stabilizes, and transforms across projects. This perspective treats personal brand not as a fixed entity to be extracted from finished work, but as an evolving set of traces produced through ongoing creative practice. Grounding this study in a trace-aware view of personal brand thus offers both a conceptual bridge to the Living Framework and a testbed for examining what it means to treat traces as a design material within a domain where cross-project creative identity is central.

8.2.2 Traces as a Design Material

My prior work in Chapter 7 argued that traces have so far been used in an ad hoc manner by creativity tool researchers, rather than treated as a foundational resource for aligning tools with creator needs. I argued that traces should become a design material that can be drawn on intentionally when building trace-aware tools. Longitudinal video content creation, together with the design opportunities around cross-project trace awareness it surfaces, gives me a real creative practice in which to begin this exploration of the materiality of traces.

In treating traces as a design material, I draw on a body of research, often called the "material turn" in HCI, that takes materiality as central to how interactive experiences are made [313]. A central commitment of this work is that a material, whether physical or computational, has properties that cannot be fully specified in advance and are instead learned by working with the material directly. In this view, where the material resists becomes as informative to a designer as

what it affords [276]. This is akin to what Schön called being in "a reflective conversation" with materials, in which a maker acts with the material, reads how the material responds, and lets that response redirect the next move [249]. My interest is in what that conversation looks like when the material is in the form of creative activity traces, and in what we as tool builders stand to learn about their limits and their potential by working them firsthand.

The material turn has examined materials ranging from physical materials to computation itself, and in studying each has built a vocabulary for the properties that working a material reveals. Much of the work most relevant here has been developed on digital materials, which have their own *grain*: the characteristic behaviors, resistances, and costs that keep a material from being easily malleable [294, 295]. Vallgårda names one such property directly as *temporal* form, the pattern of state change a computational entity enacts over time. She argues that this unfolding is something a designer must deliberately give shape to rather than leave as a byproduct of function [293]. Robles and Wiberg name a second, *texture*, to describe how the surface a material presents relates to its respective parts, and how readily one material combines with others as it is composed across a workflow [238].

Because traces consist of data, a relevant subfield for this work is one that treats data itself as a design material. Feinberg argues that data is never a neutral recording of the world but is designed at every level, through the vocabularies, schemas, and choices of aggregation that decide what is captured and in what form [98]. What matters for traces is that data carries the marks of those decisions into every later use, and so working with data always means working with something already shaped rather than raw. More recent work goes further, calling for data to be worked as a craft material with expressive and narrative properties rather than read off as clinical telemetry and discarded [118]. Traces are data of this kind: they are expressive and shaped by the tools and practices through which they are captured and kept. What this work has not asked is what changes when the data in question is not a dataset to be analyzed, but a single creator's own accumulating record of practice, kept in order to be built back into the tools that support it.

What treating traces as a material offers, then, is a way to ask what kind of thing a trace is and what properties it has, at the scope where a creator builds an identity over time rather than within a single project. The Living Framework I developed in my prior work began this inquiry, cataloging the characteristics of traces across existing systems as a first attempt to name the properties of the material. This chapter builds on that account in two ways: by working the material directly rather than surveying how systems have used it, and by doing so at a cross-project, longitudinal scope. Mapping how a trace moves through a creator's practice becomes the way I work the material, and the resulting account of that movement serves both as a first description of a trace's course through practice and as an instrument for asking sharper questions of traces themselves.

Along with mapping the trace lifecycle through the trace mapping study, the development phase in particular was a means of working directly with trace data. To do this, my approach draws on "inspirational bits", which are small, quick, fully working but deliberately partial prototypes built to expose a running material's properties [277]. In its original form, the method was a coordination device for multidisciplinary teams, a way to build shared understanding of an unfamiliar technology across people with different skills. I adapt it as a solo researcher and to a narrower end: to interrogate a single material whose properties I do not yet have an account

of, with the goal of developing a shared understanding that creativity tool researchers can take forward.

8.3 Trace Mapping Study

To examine personal brand from a trace-oriented perspective, I first needed to understand how video creators currently think about their own style and identity, how they manage their associated brand markers across projects, and where their workflows break down. To do this, I conducted a trace mapping study with seven experienced video content creators.

8.3.1 Participants

I recruited video content creators who produce content on YouTube, TikTok, and Instagram. I did not restrict recruitment by content genre, and as such, prospective participants included vloggers, video essayists, gamers, and short-form creators. My inclusion criteria required participants to be at least 18 years old, to manage both their own channel and their own post-production (i.e., they did not outsource editing), to be located in the United States, and to have a publicly available portfolio of work on at least one social media platform. I posted recruitment flyers on X, LinkedIn, Facebook, and content creation communities on Discord. Recruitment flyers were also distributed to email lists for content creation and film clubs at American universities.

Prospective participants completed a screener survey to establish eligibility. The survey collected their video editing experience, links to their public channels, how frequently they posted, and how long they had been producing video content, which allowed me to confirm both an established practice and a visible portfolio before scheduling a session. In total, seven participants took part in the study. Table 8.1 summarizes each participant's content type, posting platforms, editing tools, and the other tools in their workflow.

8.3.2 Procedure

Each study session was conducted remotely via Zoom and consisted of a 45-minute semi-structured interview followed by a 45-minute task walkthrough. Prior to the study, participants were asked to prepare an in-progress video project to share during the walkthrough. In total, sessions lasted approximately 90 minutes and were recorded and transcribed for analysis. Participants received a $50 Amazon gift card for their time.

Each semi-structured interview focused on participants' background in content creation, how they thought about their creator identity, and which markers of that identity they considered most central to their work. Questions covered how long participants had been making videos and across which platforms, how they would describe their style to someone unfamiliar with their work, what elements of their videos felt most recognizably theirs, and how their sense of identity had evolved over time. I asked participants to distinguish between markers they cared about deeply and markers they rarely thought about, as well as to describe the tension they experienced between staying on-brand and experimenting with something new. Throughout the interview, I probed for the concrete, editable implementations of stylistic choices participants

Participant	Content Type	Content Platform(s)	Editing Tool(s)	Other Tools in the Workflow
P1	Vlogs, hauls, review videos	YouTube, TikTok	Final Cut Pro, CapCut	Procreate (custom transitions), a Final Cut "editing toolbox" project
P2	Vlogs, travel and event series	YouTube, TikTok, Instagram	CapCut	Pinterest, Miro, external drives, asset/LUT folders
P3	Short-form documentary, day-in-the-life	RedNote (Xiaohongshu)	CapCut	Instagram and RedNote saved collections, ChatGPT
P4	Video essays, student podcast	YouTube, Instagram	Final Cut Pro, DaVinci Resolve	Google Docs, Notion, Procreate, Photoshop, per-marker project files
P5	Vlogs, day-in-the-life	Instagram, TikTok	CapCut	Google Docs "ideas list", YouTube screen recordings for SFX
P6	Educational STEM "infotainment"	YouTube, TikTok, Instagram	DaVinci Resolve	Ableton Live, Processing, custom animation software, Google Docs, physical notepad
P7	Gaming let's plays, comedy commentary	YouTube (incl. Shorts)	Premiere Pro, After Effects	OBS, Premiere presets

Table 8.1: Information about the trace mapping study participants, including the types of videos they make, the platforms they post on, the tools they edit in, and the tools outside the editor where their trace data accumulates.

described. When a participant used an abstract label like "moody" or "clean," I asked how that was translated into specific assets, such as a particular color filter, an exposure setting, a preset, or a copy-pasted vignette configuration.

Next, participants completed a task walkthrough and think-aloud using the in-progress project they had prepared. I asked them to show me a brand marker that felt especially meaningful and to walk through how they had added it to their edit. I interspersed the walkthrough with additional questions about their asset management practices, including how they kept track of their brand markers and whether they maintained collections of reusable markers across projects. I also asked about friction points in the editing process, and about the aspects of their style that lived in their tools versus in their own memory. When possible, I asked participants to open a second, completed project so that we could look for the elements that persisted across two otherwise different videos. Throughout the walkthrough, I took observation notes to capture additional data and to note any new questions that came up during the study.

The session closed with a set of questions about how participants imagined a video editing tool that understood their personal brand and creative history. These questions covered what

such a tool would enable them to do, which kinds of feedback would feel supportive versus intrusive, and, echoing the walkthrough, what a hypothetical assistant would need to know to keep a handed-off project on-brand.

8.3.3 Results

At the end of the study, I had seven sets of interview transcripts and accompanying observational notes taken during each session. I cleaned the session transcripts and analyzed them using an inductive coding approach to identify emerging themes [39]. The first pass of coding was fully inductive and produced seventeen codes, which clustered around three broad themes: the trace data participants worked with, the strategies they had developed for managing that data across projects, and the challenges they faced in doing so. I then made a second pass on a subset of codes to derive a set of trace-related *actions*, which led to the formulation of the lifecycle model I present in the following section.

The findings that follow are organized into five sub-sections. The first describes the brand markers participants produced and treats them as a form of trace data. I then identify a family of non-brand-marker trace data which surfaced alongside their brand markers and which behaves differently on the dimensions of the Living Framework for Trace Awareness. Next, I discuss the strategies participants had developed for managing their brand markers from one project to the next, followed by the challenges participants encountered in doing that work. I close with a set of emergent findings that sit above the single-project scope and that shape how longitudinal trace data may be studied.

8.3.3.1 Brand Markers

The study revealed the decisions that unfold in video content creation practice in the form of brand markers. Following the distinction established in Chapter 7, brand markers are a form of trace data rather than traces in and of themselves. A font sitting in a project file is trace data. The series of decisions, actions, and reuses that accumulate around that font over time is what makes it part of a trace. This distinction matters throughout what follows, since the marker types I describe below are the raw material creators produce, and following them as they accumulate across projects is what allows me to study them as longitudinal traces.

Table 8.2 summarizes the trace data found in the study. Across the seven participants, brand markers clustered into three groups. *Visual identity* includes fonts, titles and on-screen text, graphics and overlays, logos and icons, memes, animations and motion graphics, color grading, color palettes, and thumbnails. *Video structure* includes intros, outros, and transitions. *Audio* includes music, sound effects, and voice-overs.

Participants often reached for several of these markers at once when asked what their brand consisted of. P1 described brand markers as "recognizable elements, when I think of a brand, there's certain icons, colors, fonts, and even within that there is some range." P2 said, "I have a little intro video that I usually play [at the start of each of my videos]. There's certain transitions that I use. I always use the same font style. I have a lot of little background noises. There's a lot of typing sounds in my YouTube videos."

Mapped onto the data characteristics of the Living Framework, the brand markers found in this study were concentrated in the artifact and layout channels, and spanned the text, image, audio, and motion curve modalities depending on cluster.

All seven participants made systematically different stylistic choices across the platforms they published to, adjusting fonts, pacing, level of polish, and sometimes their overall persona between YouTube, TikTok, Instagram, and RedNote. Participants described this as matching the tone and audience expectations of each platform. TikTok and other short-form platforms were consistently treated as less time-intensive and more casual, while YouTube long-form work was described as more "buttoned up" (P2) and more attentive to craft and consistency. In practice, this meant that the marker set enumerated above was not a global property of a creator's identity but was indexed by the platform a given piece of content was destined for.

Though there were overlaps in the general usage of brand markers by different participants, they prioritized and valued them differently, sometimes strikingly. Fonts illustrate this cleanly. P1 described browsing for fonts as a source of pleasure and identity work: "I can experiment with fonts, and, I love fonts. So going down rabbit holes and discovering fonts, that's just fun." P5, by contrast, explicitly named fonts as low priority: "I'm not actively exploring, what font works for me... I don't see the font itself as... a way for people to identify me." A similar contrast emerged for the intro sequence: P2 spent "maybe two or three days" hand-drawing hers in Adobe Fresco, while P6 said the intro received some attention but that "a lot more thought goes into the rest of the video".

These are not simply differences in stylistic taste. The same marker type carries very different identity weight for different creators, which suggests that brand markers considered in isolation tell us little about the creator behind them. A creator's relationship to their markers (what they claim, what they reject, and where they spend their time) is at least as informative about their brand as the markers themselves.

8.3.3.2 Other Trace Data

Alongside their brand markers, participants also described a substantial amount of trace data that did not directly instantiate a stylistic identity in the finished video, but rather structured how they approached their work. This included mood boards, inspirational collections, scripts, notes, idea lists, plans and outlines, annotations, and collaborator notes. P2 described starting a new video series by first "getting on Pinterest" to ideate, and maintained a set of themed boards ("I have this one board called [P2 Name] Show. I have a board called TikTok. I have this YouTube board, and there's a secret board that I have called New Logo"). P6 called Google Docs his "main scratch pad," and elsewhere the place where "a lot of the magic happens," using it to hold his script alongside notes and annotations.

This material does not consist of visual assets that can be described as brand markers, but it is clearly part of the activity traces produced during video content creation. Where brand markers concentrated in the artifact and layout channels, this second family of trace data spread across the reference material, annotation, and feedback channels. Where brand markers were almost uniformly produced in-tool by the creator, non-brand-marker trace data was often produced out-of-tool (e.g., on Pinterest boards, Google Docs, or Miro whiteboards) and mixed user, system, and other origins, particularly when audience comments or platform analytics became part of a

creator's reference material.

Table 8.2: Brand markers and non-brand-marker trace data surfaced across the seven participants, grouped by cluster.

Cluster	Trace data
Visual identity	fonts; titles and on-screen text; graphics and overlays; logos and icons; memes; animations and motion graphics; color grading; color palettes; thumbnails
Video structure	intros and outros; transitions
Audio	music; sound effects; voice-over
Non-brand-marker trace data	mood boards; inspirational collections; scripts; notes; idea lists; plans and outlines; annotations; collaborator notes

8.3.3.3 Management Strategies

Beyond the boundaries of a single project, participants developed strategies for carrying their trace data forward from one video to the next. The reasons they gave for doing so mapped onto several of the trace functions identified in Chapter 7: predominantly reuse/remix and cohesion, but also historical reflection, cross-project comparison, and memory recall. P1, for example, described mechanisms that served both reuse and cohesion, letting her pull known-good styles and transitions into new work without rebuilding them and maintain a consistent visual grammar across her channel.

Each participant had developed some strategy for moving markers from one project to the next. These strategies varied widely, from carefully constructed asset libraries to more ad hoc approaches, but they consistently served the same underlying purpose: sustaining a consistent look and feel across a body of work that no single project file could contain. In some cases, participants only realized how substantial their own management practice had become when we walked through it together. Midway through showing me his setup, P6 said, "I can't even believe I did all of this," surprised by the accumulated shape of his own workflow.

Several participants had built explicit structures inside their video editors for holding brand markers across projects. P1 described her strategy as an "editing toolbox," which she noted "is literally a video project" rather than a folder: a separate editing project whose timeline held reusable title styles, custom transitions, and other visual elements available to be copied into new work. This shape mattered, in her account, because a video project rendered its contents in the same visual grammar she used elsewhere, making the toolbox both a storage medium and a browsable reference that matched the conventions of the tool she used in every project: "I'm able to go in and copy similar title styles at the video." P4 had a structurally similar practice for his podcast, maintaining separate projects for reusable elements ("Melvin Life Transitions, it's literally just a collection of the transition I made").

Other participants used folder-based systems. P2 kept an editing assets folder that served both TikTok and YouTube work, and inside it a LUTs subfolder for color grading. P6 maintained

a thumbnail folder; P7 kept folders for VFX and for assets that met an "important enough" threshold to warrant a dedicated location. Time partitioning was another common structure: P2 organized her raw footage into "Noel 2025" and "Noel 2026" hard drives, further subdivided by month, while P1 built a per-vlog folder containing footage and any related graphics for that video.

Participants used templates and presets where available, regardless of how they stored markers across projects. P3 relied heavily on platform-provided templating features for her thumbnails ("all my thumbnails… they're low key the same, because… you can reuse the template"), whereas P4 and P6 created their own reusable templates within their editors, whether for podcast episode structure or for audio presets that captured signal chains they had spent time tuning ("once I did it once, you know… I never have to do it again. I just save it as a preset").

A substantial part of participants' management practice happened outside the video editor entirely, across platforms whose primary purpose was something other than video production. These external practices also held most of the non-brand-marker trace data. Google Docs was the most common of these. P6 used a single document to hold his script, notes, and annotations for a video; P4 organized his ideation notes (his "rant," as he called them), his final script, and a color-coded "visual script" with brand-marker cues; and P2 kept an ideas list that spanned video concepts and specific music selections. Pinterest played a similar role for visual brand-kit work. P1 maintained long Spotify playlists as a vetting-and-storage medium for music, while P3 used screenshotting and ChatGPT as an intermediate step to figure out how to reproduce styles she encountered online, and P4 described the practice of building mood boards as part of his brainstorming and color palette selection.

These platforms functioned as containers in the same sense as the in-editor toolboxes and folders, holding trace data that would inform subsequent work. But they were shaped by the affordances of each platform rather than by the video editor. Google Docs is worth highlighting in particular: nearly every participant who used it did so differently, adapting it as a general-purpose trace container that no video editor gave them.

Not every participant maintained a formal management practice. P7 described his approach as "a pile of trash, and I know where everything is in that pile of trash." Though she relied heavily on folders and external tools to hold cross-project trace data, P2 said much of her process still "lives in my head," and P5 similarly kept "mental notes" that surfaced when she began editing a new video. P4 also kept his inspiration list mentally, noting that "if it sticks in my head, then it'll be just something I want to try out."

A related informal strategy was to copy directly from the previous video rather than maintain any store at all. P6 described this as his default: "I'm probably just gonna reuse the previous video." P1 did the same for smaller series where building a toolbox wasn't worth the overhead: "If I'm editing something like a smaller series… sometimes I won't make an editing toolbox, I'll just end up copy and pasting between a lot of videos." In these cases the previous project itself functioned as the store of brand markers.

8.3.3.4 Challenges

Across the seven sessions, I identified a range of challenges connected to how participants managed their trace data. These challenges spanned tool fragmentation, the mechanics of moving

markers between projects, the cost of maintaining a management practice at all, and the tensions surrounding when and how to change a brand over time. They are rooted in the trace data participants produced and in the demands of cross-project practice specifically, rather than in the single-project work of editing any given video.

Even before markers reached cross-project use, participants described difficulties in producing markers in the first place. Several noted that they had, at various points, been forced to settle on a marker they were not fully satisfied with, whether because they lacked the skill or knowledge to realize what they had in mind, they ran out of time before landing on something they liked, or they could not find a tool that supported the specific effect they were after. While these challenges are rooted in single-project production, they matter for this study because the markers that creators produce under these constraints are the same markers that enter the cross-project practice described in this section.

No participant's cross-project practice fit inside a single tool. Making, managing, and reusing a brand marker often meant working across several applications, then finding a way to bring the result back into the editor. P6 captured the shape of this most vividly: "there's like 20,000 different tools I have, there's Google Docs, there's Resolve, there's Ableton Live, there's Processing, there's FFmpeg, there's ShaderToy… a lot of friction of having to like open all these tabs and like move different things between them." P1 described the same: "that's why I'm having like Chrome open, Final Cut Pro, like Procreate, like all these different tools." The fragmentation participants described was not just the tool sprawl any single video project already required. Cross-project practice compounded it, since management strategies depended on a different subset of the same tools.

Over and above the fragmentation itself, moving a brand marker from where it was made to where it needed to be reused incurred a translation cost. P3 described a routine of screenshotting an inspirational style, adding it to her collection on Instagram, selecting a particular one from the pile, taking it to ChatGPT to understand how to reproduce it, and then implementing it back in her editor. She noted that this friction discouraged creators from trying to diversify their personal brand materials at all.

Even for markers that were stored inside some type of template or preset, reuse remained largely manual. P1 called copying and pasting the same titles across videos "annoying." P4 said that in past videos he had had to recreate reusable elements from scratch, an approach that "just [wasn't] viable." For markers that had no ready-made template or preset, creators had to build their own, which several participants found effectively out of reach. P3 named the absence directly: "one of the things that I've found the hardest [is doing this from] scratch… there's no centralized bank for me to get all these stuff." P6 similarly observed that although some tools offered ways to save transitions or presets, none quite matched what he wanted to preserve. This was not only a retrieval problem but an elevation problem, indicating that even when participants wanted to elevate a custom marker into cross-project use, the absence of a mechanism that fit their specific work often left them to build ad hoc storage or leave the marker in the project that had produced it. As a result, even when creators successfully imported a reusable marker, additional manual work was often needed to fit it into the new project, such as resizing, adjusting timing, or fiddling with placement. P1 described this as one of the reasons she avoided overcommitting to any specific version of a marker in her toolbox, since she expected to change it downstream anyway.

Beyond the friction of the reuse work itself, the act of running any portfolio-level management practice imposed costs of its own. P1 described this most directly: "you're working on a project... you're working on two projects at the same time at all times." Elsewhere she noted that "the management after a while will not be fun." P6 named the same phenomenon aspirationally, saying "I should totally get, like, a common library" while acknowledging that he had not built one because the overhead had never seemed worth it. Occasional lapses in a management practice also came up. P1 recalled the recurring "shit, I should have saved that" moment as one of the triggers that had prompted her to build a toolbox in the first place, so as to avoid the challenge of vaguely remembering a specific brand marker she had used from a few videos ago and needing to search through old projects to find it.

The moment of elevation itself was also a source of difficulty. Several participants described being unsure what was worth keeping in the first place, particularly for material that had no obvious later use. This uncertainty was especially acute for non-brand-marker trace data such as notes, annotations, and reference material, where creators could rarely tell at the moment of production which future project a given piece would inform. The result was that this material often stayed embedded in the project that produced it, or was saved into a general-purpose scratchpad such as a Google Doc without any expectation that the creator would return to it. As I return to in the discussion, this asymmetry between brand markers, which participants elevated deliberately and by rule, and non-brand-marker trace data, which resisted the same treatment, has implications for the treatment of traces as a design material.

Libraries also failed to scale gracefully once they grew. P1, who had the most developed management practice among the participants, articulated the concern most sharply: "imagine this gets really long... as a user, it's like, okay, how does it scale gracefully in a non-overwhelming way?" She noted that browsing a large asset library was not well supported and that storage size itself became a concern, since "you don't want to have multiple libraries that take up like 20-30 gigabytes." P4 similarly flagged that the visual density of some cross-project management surfaces made them harder to use, describing a wish for better ways "to switch between these two without having to have this visual clutter."

Where participants kept much of their process implicit rather than in an explicit management practice, they described this as a source of cognitive load. P6 named this tension: "so much of your process actually does exist [in your head, which] is a very, very classic ADHD thing, which unfortunately tools don't accommodate very well." P2 similarly wished for a space in which to externalize this material: "it'd be cool if there were some type of white space that I could just like get out all of my ideas on for all of my projects."

Deliberately externalizing everything in support of a cross-project portfolio had its own costs, however. P6 noted that "management overhead, whether it's in the brain or in the real world, has some real overhead that would be nice to have less of," and elsewhere described the tradeoff more precisely: "having [things written] down... is like good to keep it out of my head, but also... it takes time to like plan what to write on a to-do list in the first place."

Of all the challenges participants raised, experimentation with brand markers was the most pressing. The most straightforward obstacle was skill and knowledge. Several participants explicitly did not feel that they had the design or editing knowledge needed to experiment with certain marker types across videos. Even where participants had the knowledge, they often did not have the time. Several participants described editing itself as expensive enough that exper-

imentation on top of it was hard to sustain. As P1 put it, "if it was more sustainable, then I for sure would do more experimentation with my brand."

Platform pressures exacerbated the tensions around deciding when to remain consistent with their brand and when to evolve. This was clearest for P7, who described the YouTube algorithm as effectively rewarding sameness for smaller channels: "YouTube doesn't really reward experimentation... for smaller YouTubers, [it locks you] out of that one game like box." A related concern was the tension between remaining authentic to an existing brand and pivoting to follow viral trends. Several participants deliberately kept a distance from trend-driven markers so that their work continued to feel like theirs. P5 named this most explicitly: "I rarely use templates in CapCut because it doesn't really feel true to myself." Participants also worried about their existing audience and were uncertain about evolving their brand out of fears of losing their community. As P3 put it, "I'm more scared of losing my existing audience because they follow me because they recognize my style, but if I pivot... it's like betraying them... I'm scared for the feedback if it's bad." P4 gestured at a related concern, noting that any change in brand marker use needed to preserve the audience's ability to recognize the creator behind it.

Complicating the experimentation question further was that participants often could not say with certainty what had caused a drift in their brand or a sense of inauthenticity in their recent work. When such shifts registered, they could rarely be traced to a single decision or a single marker. Instead, participants described the change as something they noticed only in retrospect, and only against the accumulated body of their prior work. This made it difficult to know what to correct, since the source of drift was distributed across many small choices rather than located in any one of them.

The final two challenges played out over a longer timescale than the rest, which made them harder both to recognize and to address. Underneath all of these more concrete obstacles was a more foundational difficulty: knowing what to experiment with in the first place, and whether the resulting changes to brand marker use would remain authentic to the creator. As P3 put it, "you can't tell specifically what parts makes your content... more authentic to you." P3 framed the deeper cost of getting it wrong in terms of long-term brand building: "a lot of people... focus on present, focus on that one video that I want to actually post, but a lot of times building up a brand [across videos], or actually building up your own, internet persona, it's like it requires a lot of like planning, a lot of like intentional decisions over time."

A related but distinct challenge was projecting into the future at all. P5, still in the process of discovering her creator identity, said, "As someone who's still discovering herself as a content creator... it's pretty hard to conceptualize, like what do I want for my future... for these like personal branding things." P6 similarly acknowledged that his early plans for what he would create had turned out very different from where he had ended up.

Taken together, the challenges I describe here clustered at specific moments in creators' practice: making new brand markers in the first place, moving from one project to the next, retrieving and reusing brand markers stored across projects, and evaluating a body of work as it accumulated over time. These moments foreshadow the four-stage trace lifecycle model I introduce in the next section, which describes how brand markers become full-fledged traces that move through a creator's practice, and where existing tools most fail to support them.

8.3.3.5 Emergent Findings

Beyond my main research questions, participants revealed additional insights around how they approach cross-project work. In particular, when I asked participants to talk about their practice as a body of work rather than as any specific video, I saw them draw on a coherent set of concerns: how they thought about themselves as creators, how they worked across multiple temporal orientations at once, and where they located their creative value. I report these here as emergent findings. Each of them is directly relevant to a trace-aware account of cross-project practice and shapes how longitudinal trace-aware support might be designed for content creators.

Identity. Every participant had a distinct sense of who they were as a creator, and each expected their brand to make that identity legible across their work. Their self-descriptions varied widely, from "fabulous" (P2) to "a very analytical person" (P4) to "relatable, very real" (P5). P3 had settled into an explicit educational niche, while P7 grounded his channel in welcoming his viewers as "part of the video." What mattered for the purposes of this project was not the specific content of these identities, but that participants treated their brand markers as the vehicle through which those identities were meant to travel from one project to the next.

Not every participant, however, had a fully settled sense of their brand. P2 described being in "a creator identity crisis," having wanted to evolve her brand for some time but being unsure how. P5 was in a similar position, though she struggled with understanding what her brand was to begin with when reflecting on her full portfolio of content. Both wished a tool could reflect their own accumulated work back to them or offer some future-facing support to settle their identity crises: as P5 put it, "I would want something or someone to tell me what is my personal brand... so hard to think about myself from, like, a different perspective." This wish is only meaningful at a cross-project scale. It cannot be satisfied by looking at any single video.

Temporality. Across the interviews and walkthroughs, it became clear that participants did not work linearly. That is, they do not simply plan, shoot, edit, and publish in a repeated cycle. Instead, participants operated across three temporal orientations at once; the past work they returned to, the current project in front of them, and the future arc they were building toward. With regard to their past work, participants routinely rewatched or browsed their prior content as a form of reflection, or to mine for previously used brand markers. P6 described a practice of letting videos "sit in the tank for like a week" after finishing them so that he could catch things he otherwise would not ("that's when you really catch the things"). P4 said he got "a lot of satisfaction from watching my own videos right after they're... uploaded, but given, like, [a few] months, I'll watch it again" and would use that later viewing to notice what he might have said more clearly.

Participants also worked forward, planning against a future arc for their channel or portfolio. P1 spoke of "the channel portfolio... and then there's the little mini series or the travel sub genre," actively deciding how new videos should relate to that larger shape. P2 was actively deliberating whether her channel should undergo "a season two rebrand," and P6 imagined a future in which he had "like 20 videos, and they're all bangers." The tensions around experimentation described in the previous section were, in this light, a symptom of the same forward-facing orientation. Participants were making brand marker decisions now that they knew would shape a future they could not yet fully see. Non-brand-marker trace data played a related role, since participants like P2 maintained idea lists explicitly for revisiting later. In each case, the cur-

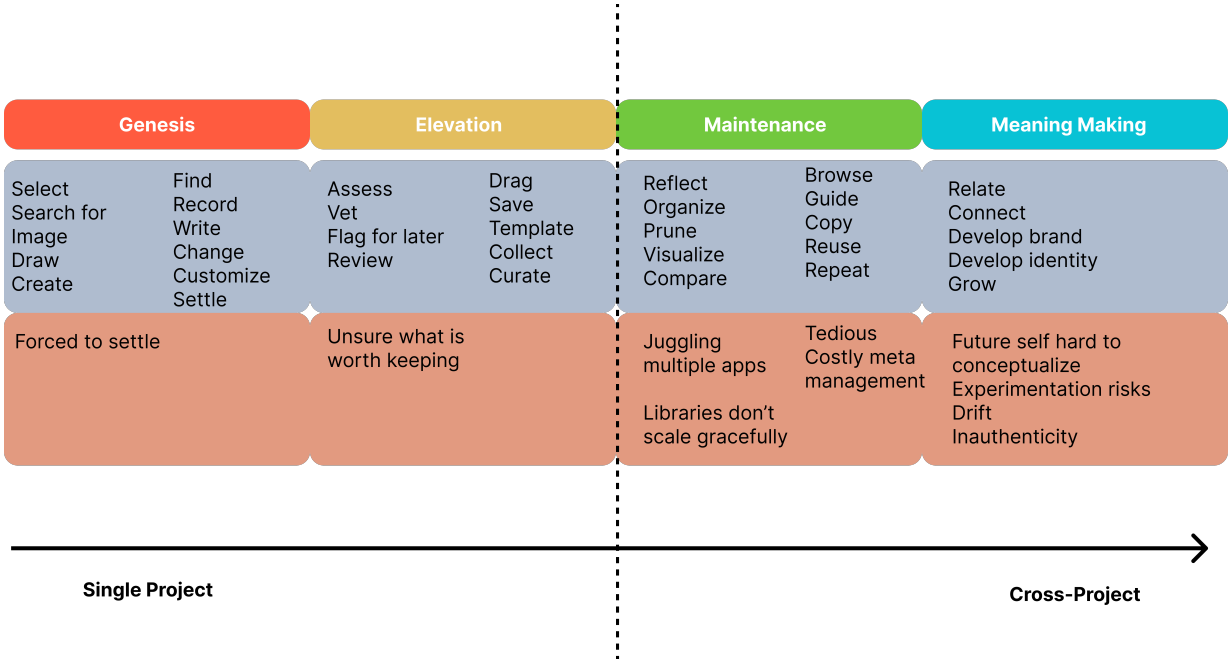


Figure 8.1: The trace lifecycle model for brand markers. Each stage lists its actions (blue) and frictions (red).

rent project sat at the intersection of reflection on past work and projection into a future one, and participants understood their videos as accumulating into something larger than any of them individually.

Creative ownership. A final, briefer thread concerned where in the video production pipeline each participant located their creative value. Participants held strong and specific views about what parts of their work they wanted to keep hands-on and what parts they were open to having tools support or automate. P2 gave the most explicit account: she was "absolutely happy" with the idea of an editor automating her color grading, but "would probably be upset if the editor were to somehow edit my intro or outro," because those were the parts she authored deliberately. What was claimed and what was open to delegation varied substantially from participant to participant. What did not vary was the presence of a boundary. Each creator had one, each was clear about where it sat, and each expected any future tool to respect it.

These emergent findings, together with the challenges and management strategies described in the previous sections, motivate the four-stage trace lifecycle model I introduce next.

8.4 Trace Lifecycle Model

The brand markers surfaced in the previous section are the raw material of a creator's personal brand, but they are not yet traces. A transition sitting in a project is trace *data*, as is a single thumbnail. What makes any of these part of a trace is what happens to them *over time*, as creators produce them, elevate them into cross-project use, revisit and reshape them across subsequent work, and eventually settle them into a legible personal brand. This section develops that

temporal account in the form of a lifecycle model that describes how brand markers evolve from single-project artifacts into longitudinal traces that carry a creator's identity across their body of work. In doing so, the model brings together two threads from the emergent findings: *temporality*, as the mechanism by which brand markers become traces, and *identity*, as what those traces come to represent over time.

The Living Framework for Trace Awareness introduced in Chapter 7 defines what trace data looks like inside a system, via the data characteristics dimensions I used above to describe the brand markers in this study. What it does not describe is how that data comes to constitute a trace that systems can then engage with through the trace awareness dimensions, a gap that became sharp during analysis. The brand markers I had cataloged were static objects, but every participant described them as things that were elevated into a toolbox once they proved reusable, pulled back into a new project, reworked to fit different platforms, quietly abandoned, or actively pruned as their sense of brand shifted. Without a way to describe that movement, I could not translate the trace mapping study into anything a system could act on, nor into a material that can be studied. The lifecycle model developed here fills that gap.

To derive the model, I made a second coding pass over the qualitative data, this time coding for verbs or actions creators reported taking on their brand markers as their work moved between projects. This produced a corpus of trace-related actions such as saving, presetting, dragging, reviewing, reflecting, and pruning. I then affinity clustered these verbs, along with the challenges derived from the first coding pass, to look for structure. Early attempts drew on journey mapping to sequence individual actions along a creator's timeline [115].The model's eventual form was influenced by Amershi et al.'s guidelines for human-AI interaction [18], which organize guidelines temporally by the moment in an interaction at which they apply. Following that shape, the clustering surfaced four stages that denote key events in a brand marker's evolution. Each stage clusters a coherent set of creator actions and characteristic frictions, making the temporal structure of the material legible.

The four stages of the model are Genesis, Elevation, Maintenance, and Meaning-Making. Each stage has its own set of actions on brand markers and its own frictions, as seen in (Figure 8.1).

8.4.1 Genesis

Genesis is the stage at which brand markers first come into being. A creator selects a font, records a voice-over, draws an intro frame, chooses a color palette, or settles on a transition. This is single-project work in the sense that the marker exists as a component of the video being made and has not yet been treated as anything more. From a trace perspective, Genesis is the least eventful stage. Brand markers are produced, but they have not yet accumulated into the cross-project traces that the later stages describe.

Verbs at the time of Genesis cover the range of ways in which brand markers are produced in the first place. Some are active production: drawing, recording, writing, creating. Others describe selection or customization from existing options: choosing a font, adjusting a preset, changing a color, accepting a default. A third, smaller group comprise inspiration-gathering activities that generate reference material rather than final assets, such as watching other creators' work or collecting images to draw from. What these verbs share is that they operate inside the

boundaries of a single project. Creators varied most at this stage in how much investment they put into any given marker type, from P2 spending two or three days hand-drawing her intro to P5 treating fonts as low-priority background choices. The frictions at this stage are the single-project production challenges described in the previous section, concentrated at the moment markers are made. Participants' accounts of being forced to settle on a marker they were not fully satisfied with fall here.

8.4.2 Elevation

The second stage, and arguable the most crucial; for understanding longitudinal traces, is Elevation. Elevation is the stage at which a brand marker crosses from a single project into cross-project use. It is where a font, transition, or template stops being a component of the video it was made for and becomes something the creator intends to reach for again. In the data, this crossing over was almost always a deliberate act rather than an accident of storage. Creators had developed rules, sometimes explicit and sometimes tacit, that governed which markers earned elevation into their cross-project practice. Elevation is defined by this deliberateness as much as by the transition in temporal scope.

The verbs that define Elevation cluster around two related activities: assessment and consolidation. Participants described the assessment side as reviewing, vetting, flagging for later, and, in P6's phrasing, "graveyarding," the practice of setting things aside in a designated space to be reviewed or discarded at a later pass. The consolidation side comprises the actions that formally move a marker across the project boundary, such as dragging it into a shared editing project, saving a preset, building a template, or moving it into a folder that lives outside any single video. The two sides usually operated together. Assessment produced the judgment that a marker was worth keeping, and consolidation gave that judgment a home.

Participants articulated their elevation rules in different ways, but the underlying logic was consistent. P7 named the criterion as simply "is it going to be reused?" P4 used an importance threshold, keeping dedicated folders for material that met it and discarding what did not. P1 described her editing toolbox as receiving markers she found herself repeating across videos ("if I'm like, oh, I really liked how I did this in this video, this deserves to go into the editing toolbox"). What varied was the mechanism tied to the transition into cross-project scope. Every participant with a cross-project practice had some criterion for what earned Elevation.

The frictions at this stage are the elevation-side challenges described in the previous section: the difficulty of judging what is worth keeping; the mismatch between custom work and the elevation mechanisms available in current tools; and the acute version of both that appears for non-brand-marker trace data. What the stage structure makes visible is that these are all difficulties of the same kind, arising at the same moment in the lifecycle, and that a tool concerned with supporting Elevation would need to address them together rather than treating them as separate problems.

8.4.3 Maintenance

Maintenance is the stage at which brand markers, once elevated, live and are put to work across projects. It is the nexus between the single-project scope where markers are made and the cross-

project scope where they accumulate into something larger than any individual video. Where Elevation is a boundary-crossing event, Maintenance is an ongoing state. It is bidirectional by nature, as markers move outward from the cross-project store into new single-project work, and back into the store as creators reorganize, reshape, and prune the material they have accumulated. These two directions are not sequential. Retrieving a marker often surfaces a reason to update the store, and reorganizing the store often triggers a fresh round of retrieval. Maintenance is where the cross-project practice actually lives, and where its running costs are paid.

The verbs at Maintenance cluster around two directions of flow. The *retrieval*-facing verbs describe pulling from the cross-project accumulation into new work by recalling, browsing, finding themes, comparing, copy and pasting, reusing, repeating, and guiding. These are the actions that let a creator draw on their accumulated material when starting or working through a new project, whether by copying a title style from an editing toolbox, pulling a color palette from a mood board, or simply reopening a previous project as a template for the next. The *curation*-facing verbs describe the work of keeping the store itself in usable shape, namely by organizing, structuring, sectioning, separating into folders, pruning, and deleting. In practice, participants moved between these two directions constantly. Experimentation belongs here as well, in the sense of trying variants of an elevated marker or comparing alternative uses of it across a body of work, though the broader strategic question of whether to experiment with one's brand at all belongs to the Meaning-Making stage.

The frictions at this stage are the cross-project management challenges described in the previous section, and they concentrate at Maintenance because Maintenance is where creators bear the running costs of a practice that Elevation only opened. The tool fragmentation, the manual reuse tedium, the libraries that failed to scale, and the cognitive cost of process that lived in creators' heads rather than in any external store all show up here. The stage structure makes visible that these are not incidental frustrations distributed across the practice, but the standing tax of running a cross-project store at all.

8.4.4 Meaning-Making

Meaning-Making is the stage at which brand markers, having accumulated across many projects, come to shape a creator's sense of who they are and what their body of work is becoming. This is the longest of the four stages, in the sense that its unit of time is not a single project or a series of them but the *arc* of a creator's practice as a whole. Unlike the earlier stages, Meaning-Making does not operate on individual markers so much as on the *shape* those markers make together. A single intro sequence does not constitute a brand. Nor does the same title style copied across a short series of videos, as P1 did for her smaller series; that is Maintenance doing its work. A pattern of intro sequence choices across a hundred videos, considered against the audience that has grown up around them, does. The stage is characterized by this shift in the object of the work, from markers as components to markers as evidence of an identity taking shape over time.

The verbs of Meaning-Making differ in kind from those at the earlier stages. Where Genesis, Elevation, and Maintenance describe operations that creators perform on their markers, whether making them, elevating them, retrieving them, or curating them, the verbs at Meaning-Making describe the orientations that creators hold toward the accumulated whole. Some participants spoke of developing a brand, developing an identity, aligning with a desired theme, making their

work consistent or familiar over time, and scaling their practice gracefully as it grew. Others spoke of the outward-facing side of the same activity, of catching viral trends and optimizing for the platforms they published to. These are not verbs one applies to a single marker for reuse from one project to another. They describe the longitudinal posture a creator adopts toward the body of work which their earlier markers have built. The broader question of experimentation also lives at this stage, in the sense of deciding whether and how to change one's brand over time, as distinct from the act of trying out variants within an established brand that belongs to Maintenance.

The temporal structure of Meaning-Making differs from the earlier stages as well. As I described in the emergent findings, participants routinely operated across three temporal orientations at once: reflecting on past work, working on the current project in front of them, and planning toward a future arc for their channel or portfolio. At the single-project and cross-project scopes, one of these orientations tends to dominate at any given moment. At the long-term scope of the Meaning-Making stage, all three run simultaneously, and creators' decisions about their brand pull on each of them at once. Meaning-Making is where past work is evaluated for what it has come to mean, present work is calibrated against that evaluation, and future work is projected against the arc the past and present suggest. This overlap is part of what makes the stage structurally different from the others, and part of why its content resists the discrete, action-level description that fit the earlier stages.

The frictions at Meaning-Making are the long-term brand challenges described earlier: the experimentation tensions around authenticity, audience, and platform pressure; the identity crises of participants still working out what their brand should be; the way drift and inauthenticity register only in retrospect and cannot be traced to a single decision; and, underneath all of these, the difficulty creators had in seeing the value of their accumulated traces at all. What the stage structure makes visible is that these frictions share a common shape. They are difficulties of acting under delayed feedback and confounded attribution, arising at a timescale where cause and effect are separated by many projects and where the object of the work is the accumulated whole rather than any individual marker.

Because Meaning-Making sits at the longitudinal end of the lifecycle, its outcomes depend on what happened at the earlier stages. What was elevated shapes what accumulates. What was maintained shapes what is available to draw on. Frictions at Elevation, such as being unsure what was worth keeping or being unable to elevate custom work into a durable form, propagate forward as gaps in the material which a creator has to make meaning from years later. Furthermore, frictions at Maintenance, such as libraries that failed to scale or process that lived only in creators' heads, propagate forward as retrieval failures at the moment Meaning-Making would draw on them. This cascading structure is part of what the model makes visible. Because the earlier stages feed the later ones, interventions targeted at earlier stages may improve Meaning-Making without acting on it directly, which is a design implication I return to in the discussion below.

Taken as a whole, the model surfaces what the results alone could not. The stages make visible that the challenges participants described are not scattered across their practice but rather concentrate at specific moments in a brand marker's evolution. The model also makes visible that these stages feed each other. That is, what happens at Elevation shapes what is available at Maintenance, and what happens at Maintenance shapes what a creator has to draw on at Meaning-Making. Perhaps most importantly, brand markers that move through Genesis, Elevation, and

Maintenance eventually do the identity work of Meaning-Making, carrying a creator's sense of who they are across a body of work that no single project can contain. Rather than treating this shape as a specification for a system to build, I use it as a set of entry points for studying traces as a design material. In the following section, I describe a set of inspirational bits targeting two stages in this model, each designed to expose what it is like to work with traces at those points in time.

8.5 Inspirational Bits

The trace lifecycle model made the temporal evolution of a creator's brand markers legible through a visual account of how brand markers become traces. More importantly for the purposes of this thesis, it offers a set of entry points for studying traces as a design material.

To use those entry points, I turned to *inspirational bits*, a form of material-driven prototyping introduced by Sundström et al. [277]. A bit is a small, quick, fully working but deliberately partial prototype. Its goal is not to solve a user's problem but to expose the properties of a design material through direct, hands-on encounter with it.

This project has thus far surfaced no shortage of opportunities to design trace-aware support for creator needs, and a system assembling them would certainly be needed in the future. Building it, though, would mean committing to those features before earnestly exploring the nature of traces. In adopting the method described by Sundström et al, I reorient prototyping away from developing trace-aware features and toward asking specific questions about the trace material. Designing full trace-aware systems is thus left to future work.

Toward that goal, I began by generating questions that emerged from the trace mapping study and the modeling stage. These questions were then used in a low-fidelity generative sketching process to produce as many bit ideas as possible. These ideas ranged from a "fresh eyes" bit which would lock a video editor after a project is complete to force reflection on its traces after a certain amount of time has passed, to a "ghost" bit of a video editor whose deleted brand markers do not disappear but drift into the margin as faint sediment, so that a finished video sits surrounded by a visible cloud of everything the creator decided against. In the end, I selected two ideas, one focused on Elevation and one on Meaning-Making. Specifically, the questions I sought to answer in this phase of the project are:

1. What happens to a brand marker when it is elevated past the project boundary?
2. What happens to traces at the long-term scope required at the Meaning-Making stage, where identity concerns form and no single trace can be isolated as their cause?

I developed the bits using Claude Code with Claude Opus 5 and Claude Sonnet 5. Below, I describe the bits that were created towards answering the above questions, and outline the design process and lessons learned throughout.

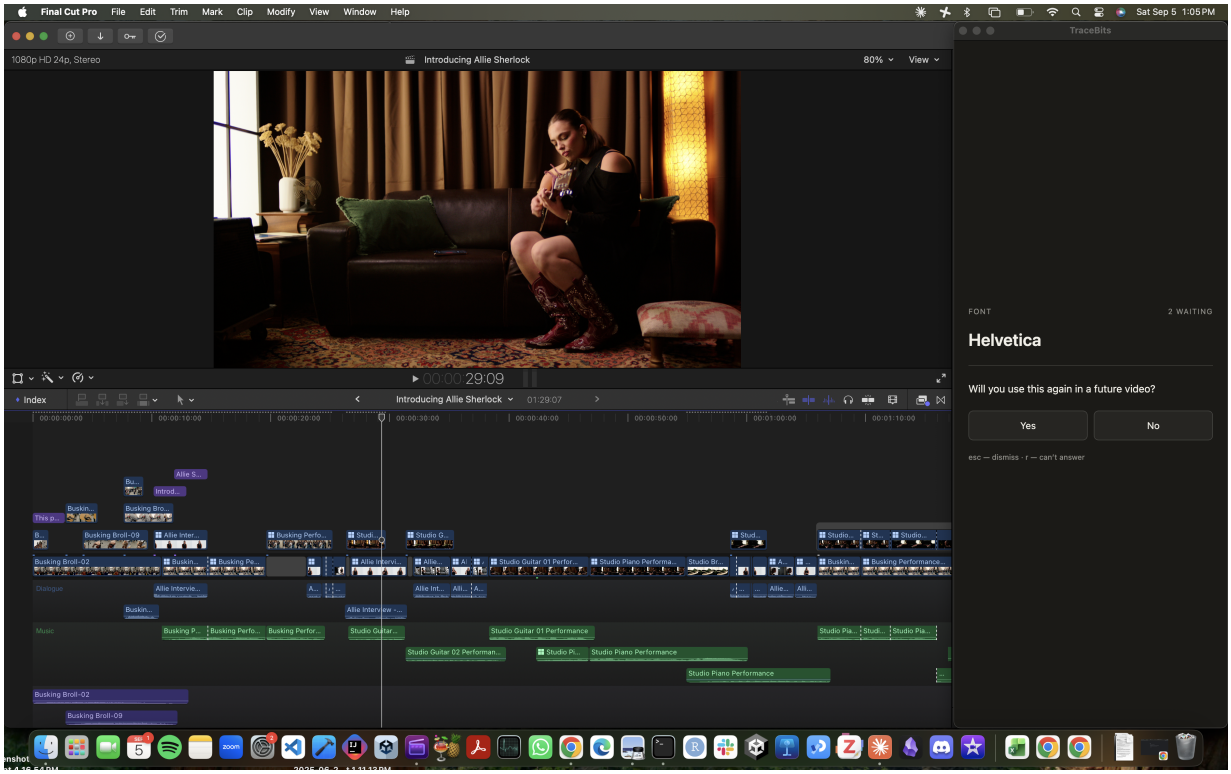


Figure 8.2: The Elevation Palette's reuse panel, asking whether a marker will be used again.

8.5.1 Bit 1: Elevation Palette

8.5.1.1 Premise

Elevation is the point in the model where a marker stops simply being a component of the video and becomes a cross-project entity. In this sense, Elevation is the stage in which the *transition* to a longitudinal trace occurs. What exactly happens to the material at this transition?

The Elevation Palette is a small Electron panel that runs beside Final Cut Pro and watches the projects a creator works on. The first time a brand marker appears in a project, the panel surfaces it once and asks a single question: will you use this again in a future video (Figure 8.2)? The creator answers yes or no. This repeats for each new marker, one at a time, after which the panel falls silent. A few projects later, at a distance that is itself an adjustable parameter, the panel returns and reveals what the creator actually reused in their projects thus far (Figure 8.3). A marker that was elevated and recurs is a hit; a marker that was elevated but never recurs, or that recurs without ever being elevated, is a miss. In this manner, the palette is built to feel like a "quiz" whose answers arrive later, often as a surprise.

8.5.1.2 Architecture

The bit consists of an Electron front end that relies on a set of Python components communicating through a single append-only JSONL log. Markers are read from Final Cut Pro's native FCPXML. Final Cut Pro writes FCPXML only on an explicit export and not on save, and ac-



Figure 8.3: The Ledger Panel, revealing which markers were reused across videos

cessing it in real-time was difficult to implement. Instead, the bit watches a folder into which the creator exports the FCPXML at chosen checkpoints. To mimic real-time use, the export is bound to a hotkey that quickly writes the project's FCPXML for the prototype to read. The extractor keys each marker found in the FCPXML on the coarsest value the format exposes such that Helvetica Bold and Helvetica Regular, for example, are treated as the same marker. Each marker the extractor finds is written as a single `item_extracted` line to the log, tagged with the project it came from and its identity key. The panel then appends each elevation query and the creator's response as `asked` and `answered` lines against that same key. Recurrence is never recorded on its own. It is inferred at scoring time whenever a key reappears in a later project's export.

The "ledger" or scoring panel renders results by a single fold over the log. In one pass it collects, for each marker key, the ordered list of projects the marker appeared in and the first prediction made about it, then judges that against whether the marker recurred in a later project. Evidence is counted only from projects after the prediction, and a negative verdict is withheld until a two-project lag has elapsed, while a recurrence is credited whenever it occurs. The fold emits the two columns the panel displays: markers elevated but never reused, and markers that recurred but were never elevated.

Because cross-project traces cannot be examined without a meaningful catalog, and I had no longitudinal creator Final Cut Pro projects to work with, I manually created three Final Cut Pro video projects and automatically generated ten variant FCPXML projects. The generator emits real FCPXML parsed by the same extractor as the authored data. This catalog was sufficient to exercise the bit across a project sequence, but it is a thin substitute for a real longitudinal corpus.

I treat this as a central limitation of this approach to studying traces and return to it in Section 8.7.

8.5.1.3 Design Process and Lessons Learned

The Elevation Palette exposes a property of brand markers that only becomes visible once the panel is used across a run of projects. That is, the information that would tell a creator whether a marker is worth elevating does not exist at the moment they are asked to elevate it. Answering the panel's question project after project, I was committing to markers with no way to see what any commitment would come to. Whether a marker recurs, and so whether elevating it was warranted, is settled later, in work that has not happened yet. Elevation is therefore a *prediction* made under uncertainty. The creator places a bet about their own future practice at a point where the evidence that would settle the bet is unavailable, as a marker can be elevated and never recur, recur regularly without ever being elevated, or be stranded in the project that produced it. Read as a claim about the material, a brand marker's value as a trace is temporally blocked at the moment of elevation.

The ledger makes a second property visible: the set a creator elevated and the set that actually recurs do not necessarily coincide. In other words, the deliberately elevated library is a biased sample of what persists in a creator's work rather than a faithful record of it. What a creator can curate and what their practice retains pull apart. There is a further reading of the elevation itself. When a creator elevates a marker they are predicting its reuse, but they are also recording a judgment, made at that moment, that the marker was worth carrying forward. The elevated set is thus a record of what a creator valued at each point in their practice.

Because elevation runs on prediction and the elevation set is already skewed, the consequences do not stay local to the project where a marker is decided. The markers a creator commits to, warranted or not, become entities that Maintenance curates and that Meaning-Making must later make sense of. A misidentified building block can thus have cascading consequences at every later stage of the lifecycle.

8.5.1.4 Bit 1.5: Applying the Same Instrument to Non-Brand-Marker Data

The study and the model were scoped to brand markers, and deliberately so. Brand markers are discrete, nameable assets, which allowed me to examine cross-project trace behavior without first having to resolve what counts as a meaningful trace unit. But brand markers were only part of what the study surfaced. Creators also produced trace data that belonged outside the artifact and layout channels, namely scripts, planning notes, idea lists, and annotations. These forms of trace data resisted the model. A script draft has no clear passage through Genesis, Elevation, and Maintenance, so its life as a trace is hard to locate, yet it is unmistakably part of a creator's activity.

Rather than build a second instrument for this, I used the Elevation Palette on it unchanged, reasoning that where the instrument broke would reveal where the differences lay. To examine these other forms of trace data, I used video planning documents akin to those described by the participants in the study in conjunction with the Elevation Palette panel. The architecture did not require major changes to function with a video planning document: a second extractor reads an exported .docx document in place of an FCPXML file. The Electron panel is left intact, and, as

with the brand-marker catalog, I automatically generated a small set of planning documents to run the bit across projects.

The question the brand-marker case had let me sidestep, what counts as a meaningful trace unit, now had no obvious answer. A sentence, a paragraph, the whole document, or its section structure are each a defensible unit to treat as a candidate for elevation. Furthermore, from a traces perspective, the document offers no reason to prefer one over another. I keyed the extractor to the sentence level for the sake of running the test, and the panel fired its reuse question hundreds of times per document, compared with the handful of times it asked per project in the video case.

The deeper failure concerned the elevation criterion itself. Reuse was the wrong question to ask, and the scoring logic from the ledger broke as a result. A brand marker is elevated deliberately, and the act of elevating it is itself a prediction that it will return. No one elevates a planning document in that sense, and creators do not set out to carry these files forward the way they carry an outro or a transition. In this manner, the actual value of this data within a trace is something that creators cannot foresee.

None of this is to say that such data is not a critical part of the trace or is without longitudinal value, especially since creators clearly lean on these documents from one video to the next. From a trace function perspective, this trace data is likely most useful for prospective functions, such as contextual suggestions or ones that have yet to be identified (see Section 8.6). Materially speaking, the two cases suggest that traces do not have a uniform grain. In the material turn, grain names the characteristic behaviors and resistances that shape how a material can be worked [294, 295]. Applied to traces, grain describes what individual trace data is made of, the data characteristics that shape what it affords and what it resists. Brand markers and planning documents share key temporal characteristics, but their grains differ: brand markers readily afford deliberate elevation and reuse, while the cross-project value of planning documents is not yet clear and will require treatment suited to their grain.

8.5.2 Bit 2: The Oracle

The second bit interrogates the meaning-making stage, the stage least accessible to study and, for that reason, the one most in need of a way to examine it. Until this point, my research had concentrated on the most fundamental building blocks of traces. Meaning-making sits at the opposite end. Here, traces have accumulated for long enough that they begin to contribute to identity-forming behavior, and the individual trace stops being the useful unit of analysis. A single color palette choice, or even the evolution of that choice over time, no longer explains much on its own. What carries a creator's identity is the way their traces have accumulated and woven together, not any one of them in isolation. At this stage a creator's traces can only be reasoned about through the *shape* they have taken on collectively, and it is this shape that the Oracle sets out to render.

8.5.2.1 Premise

The Oracle is an "all-knowing" entity that claims to know the creator's future (Figure 8.4). Upon opening the bit for the first time, the user is greeted with an invitation to understand what lies



Figure 8.4: The Oracle Bit's opening screen

ahead for them. Behind this intro sits three views of the "shape" of the creator's traces, presented as separate tabs in the interface (Figure 8.5). Each view renders the shape of the traces differently and pronounces its own single, confident sentence about who the creator is, with nothing offered to justify it. That unearned certainty is deliberate. Identity at this scale is precisely the thing a creator cannot see plainly in their own work, and the Oracle's flat assurance is a deliberately blunt way of putting that on screen rather than pretending to resolve it.

The first view, *spread*, draws the trace of each brand marker type as a "thread" running the length of the creator's history. A thread's tightness shows how settled that marker type is: a type whose markers recur steadily across projects reads as a tight, near-straight line, while one whose markers are unsettled and inconsistent is frayed and torn. Here the Oracle speaks in the tone of a stable reading, telling a creator with tight threads "you have known who you are for a long time." Spread is the view where the traces are still separable, and where the creator can tie the Oracle's statement most clearly to the shape of the traces; the creator can, for example, look at the music/audio thread, see it holding steady, and find in it the ground for what the Oracle claimed.

The second view, *investment*, keeps the same trace threads, but thickens each in proportion to how heavily the creator has invested in that marker type across their history. Thickness and spread can disagree. Transitions used constantly but never the same way twice read as thick yet frayed, while sound effects used sparingly but identically read as thin yet tight. Here the Oracle speaks to what the creator pours themselves into, naming the heaviest thread as a source of their enjoyment, whether or not it is the most consistent. Inspection now only half pays off. The creator can see which threads are thick, but not why weight should mean what the Oracle says it does.

The third view, *authenticity*, pulls the separate threads into a single knot. The threads are

still there, but they are now wound through one another so densely that no single one can be followed from end to end. Here the Oracle makes its most confident and least accountable claim, pronouncing on whether the creator's body of work still feels like theirs.

8.5.2.2 Architecture

Unlike the Elevation Palette, the Oracle is a one-shot session rather than a live tool. It opens, reads, and renders from a single precomputed file, and is a separate Electron application that borrows the Elevation Palette's extraction logic. The file it reads is produced ahead of time from the same simulated catalog used in Bit 1: FCPXML taken from three manually created video projects and ten variant projects.

A crucial design constraint placed on this bit is that it does not attempt to measure or quantify any real trace shape. The numbers it computes are crude, and exist only to render something from the available FCXML. This precomputed file holds two values per brand-marker type. *Spread* measures how consistently a marker type recurs: a type with a marker that appears in every project scores near one, while a type whose markers are all one-offs scores near zero.

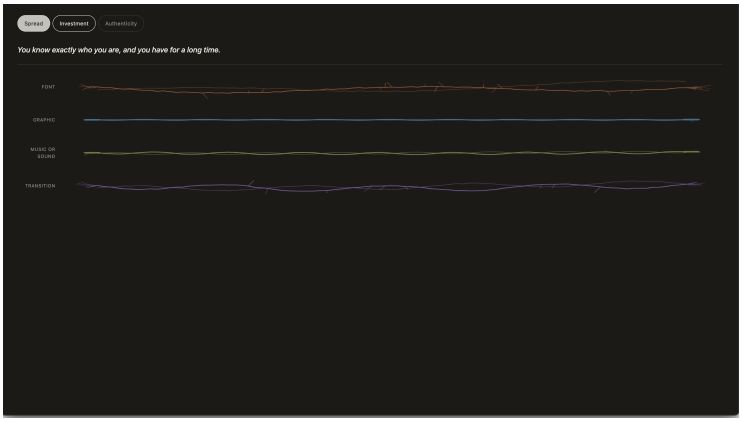
Investment is simply how many times markers of that type are placed across the whole run, counting every placement rather than collapsing repeats. The two diverge because one tracks recurrence and the other tracks sheer volume, which is what lets a brand marker type read as thick yet frayed. Investment is the harder of the two to represent, since how much a marker type matters to a creator is not a quantity easily derived from brand markers alone.

The third view computes nothing, as the knot takes no argument and reads no field of the file. It is a fixed, hand-drawn image. This is not an expedient but a consequence of the design process, in which I ran into the limits of trying to distill meaning-making into separable traces. The whole the knot represents cannot be reconstructed from the threads beneath it, so at this scale the whole is an authored image rather than a computed one.

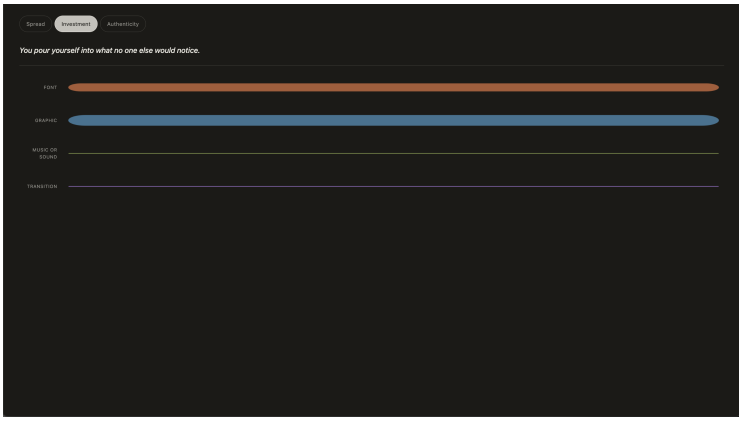
8.5.2.3 Design Process and Lessons Learned

Of the four stages, meaning-making was the one I least knew how to study, and also the one I most wanted to. It occurs only once traces have accumulated over time, as a creator's identity forms across more work than any single project holds. What I wanted from this bit was a concrete way to grapple with what traces become at this scale, and the lessons came almost entirely from the ways the material refused the instruments I built for it.

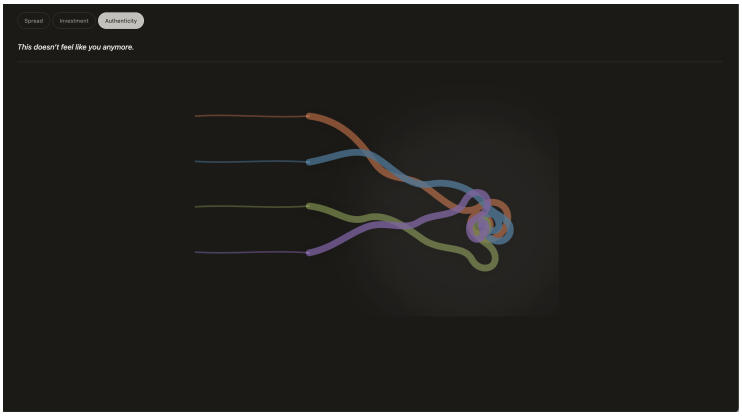
My earliest iterations of the Oracle bit operated on a single trace at a time, and their purpose was to show that the single trace is the wrong thing to operate on here. In one, the Oracle delivered a confident but flawed reading and let the creator dig through their traces to find where it had gone wrong. In another, the creator pruned individual traces, cutting a font or a transition and watching the Oracle's pronouncement shift. Both did what I designed them to do, and both made it plain that a creator's identity cannot be reached by isolating disparate traces. But a demonstration that the parts cannot be operated on is only a negative result; on its own, it says nothing about what the material has become. Accumulated to this scale, the traces had reorganized into something that could no longer be read off any one of them, a whole with a shape of its own, and rendering that shape is what the three views set out to do.



(a) The spread view



(b) The investment view



(c) The authenticity view

Figure 8.5: The Oracle bit's three shape views

Taken in order, the three views are a single experiment in how much of that shape a creator can still reach by inspecting the parts. Spread is the baseline, and it is the one view where inspection holds. The threads are still separate, and a creator can pull the music thread, see it running steady, and find in it the ground for what the Oracle claimed. The whole is legible through the parts here, because in this view the parts still stand for it.

Investment is where that begins to fail. I designed it knowing how hard investment would be to quantify, and built it anyway because creators plainly value their markers unevenly. In building this view, I had wanted to see whether a trace-based instrument could reach that relationship at all. The threads are still there to grab, thickened by use, but what the Oracle reads off their thickness, how much of themselves a creator has put into a marker type, is not a quantity the brand marker traces ever recorded. Two creators can build on the same font and value it entirely differently, one indifferent and one deeply invested. This is a difference my study surfaced directly, and nothing in the brand marker traces I had collected thus far tells them apart. I stood the view up on a crude placement count precisely because the real quantity was never there to measure in the grain I was reading.

Authenticity is where the shape of the traces becomes too difficult to discern. This was the view I most wanted to work, because creators care about authenticity enormously and can almost never locate it. Drift in tone, and the slow slide of a body of work away from itself, is something they reported noticing only long after it has set in. But authenticity is not a property of any trace. It lives in the relations among them and in the pattern they make together. In the material turn's terms, this is a matter of *texture*: how a material's parts relate to the whole it presents, and how readily it combines with others [238]. Where grain describes what a single piece of trace data is made of, texture describes how individual traces weave together as they accumulate, and it is in this texture that identity lives.

In this view the threads are wound so tightly that none can be followed end to end. When a creator tries to inspect it as they could in the first two views, there is nothing to take hold of, no thread to pull, no part to which the verdict can be traced. The hand-drawn knot is what remained after the attempt to quantify, or even categorize, the traces failed.

Through Genesis, Elevation, and Maintenance, the creator acts on the traces: making markers, elevating them, curating them. In this manner, traces as a material are constructed and are *worked on* by the creator. At meaning-making the relation inverts. The accumulated work begins to act on the creator, settling into an identity that shapes how they see themselves and narrows what they can become. This is where the challenge of the stage finally shows itself. A creator who wants to change how their work reads, to pull out of a drift they have noticed and dislike, still has to reach for individual markers, because the parts are the only handles there are. Yet the thing they are trying to move exists only in the texture, where no single edit can be isolated from the rest. The same gap makes the change almost impossible to judge, since its effect cannot be read until the traces have settled into a new whole, by which point it is confounded with everything else that shifted.

8.6 Discussion

This chapter set out to achieve the following: to understand how video content creators build, maintain, and evolve a personal brand, a longitudinal, cross-project problem that trace-aware systems have so far left alone; and to work with traces as a design material and ask what that material is like. Through this work, I move toward building better trace-aware tools that leverage traces more effectively and address the design challenges creators face.

The trace mapping study showed that video content creators care about their personal brand a great deal, but they approach their brand differently from one another. Asked what their brand consisted of, participants reached for similar inventories of fonts, intros, transitions, color, and sound, yet the same marker type carried very different weight from one creator to the next. Underneath these differences sat a self-concept that each creator held, or wanted to hold, and that lived beyond any particular marker. The markers served as the vehicle through which that self-concept was assembled and carried from one video to the next. Creators were, in this sense, highly opinionated, and they could articulate their brand and the obstacles to sustaining it with real precision. A personal brand, then, is not simply a set of assets a system could read off finished videos. It is a stance a creator holds toward their own material, and that stance is what a trace-aware system must reach.

However, these perspectives and maintenance behaviors did not necessarily extend to the rest of the trace data their practice produced. More specifically, the scripts, notes, and mood boards they leaned on constantly were rarely recognized as carrying cross-project value at all. From a traces perspective, this data may hold value of a kind the brand markers cannot. In the Living Framework, the trace functions the brand markers support lean toward material interaction, such as reuse, remix, and cohesion, whereas scripts, notes, and boards sit closer to *how* a creator works than to *what* they produce. This in turn points toward workflow-oriented functions, such as workflow alignment, that the current framework has not yet captured.

Much of the difficulty of sustaining a brand across projects sat in creators' tools. No participant's cross-project practice fit inside a single tool. What matters for this thesis is not only the particular frictions but the stages at which they occur. The trace lifecycle model synthesized from the study data provides a map of how traces evolve and where trace-aware support is best suited. Read across its four stages, the scattered challenges participants reported resolve into a single trajectory: a brand marker coming into being at Genesis, crossing into cross-project use at Elevation, being kept and reused at Maintenance, and accumulating into identity at Meaning-Making. Because traces accumulate over time, a marker wrongly committed at Elevation surfaces its cost later, at Maintenance and again at Meaning-Making.

This cumulative behavior makes Elevation the most promising place for a trace-aware system to act. Elevation is where a single marker enters a creator's cross-project trace and where the temporal scope shifts. In the case of brand markers, elevation occurs deliberately on a per-marker basis. Creators must assess each marker's cross-project value before committing it, and as the Elevation Palette showed, that assessment is a prediction made under uncertainty, since the evidence that would settle it only arrives in later work. Addressing deliberate elevation practices is thus key for making creators' "palette of materials" more aligned with their needs [267]. The other trace data found in this study is not deliberately elevated at all, and instead accumulates in disparate folders and documents with no clear path into longitudinal cross-project practice.

Both of these challenges point to a form of support I call *system-supported* elevation, as opposed to user-driven elevation. By this, I mean mechanisms that carry a trace forward when the creator will not, or cannot, do it themselves. For the first case, system-supported elevation can take the form of trace-aware features that watch what a creator actually reuses across projects and prompt them to elevate a recurring marker they never deliberately saved. The Elevation Palette gestured at what that evidence might be, since the markers a creator actually reuses and the ones they deliberately elevate may not line up. For the second case, system-supported elevation might instead surface workflow-facing trace data, such as a note or idea list, when later work makes it relevant, since creators could rarely tell at the moment of production which future project it would inform. What that relevance looks like remains an open question, given how unsettled this data's grain is. What to elevate, when, and toward what end remains a fruitful avenue for future trace-aware support in this area.

Meaning-Making is where this logic reaches its limit. It is the stage at which traces become inseparable entities, and where feedback for creators arrives many projects later. System-supported elevation can help ease these issues by reducing mistakes that cascade into later stages. This does not fully address the challenges faced in Meaning-Making, however. Because the difficulty here is that no single trace can be isolated as the cause of how a body of work reads, the support a system can offer is not corrective but reflective, surfacing the accumulated *shape* of a creator's traces back to them, so that a drift they can feel but not locate becomes something they can at least see. How to conceptualize and model this shape is a harder and more open design problem than elevation, and exploring how traces weave into the texture that gives rise to meaning-making concerns will require deeper study.

None of the above findings were reachable from the Living Framework on its own. That framework was drawn from a corpus of creativity systems that operate almost entirely within a single project. The framework says a great deal about what a trace is made of and what a system can do with it, but it has no way to represent what becomes of a trace over time, especially once traces accumulate past the boundary of one project. It characterizes trace data one piece at a time and has no account of how traces combine, yet it is their texture, the way a creator's traces weave into something none of them carries alone, that does the identity work that creators are concerned with over time. The lifecycle model offers one way to begin filling that gap, as a *lens* that sits between the framework's data characteristics and its trace awareness dimensions, though it is unlikely to be the only lens needed.

What kind of material, then, are traces? Working with them across the study, the model, and the bits, the clearest thing I learned is that they are not a uniform material that can be treated consistently throughout the design process. The differences between trace data found in this study, and how they were applied in Bit 1 and 1.5, show this nuance. Traces share key temporal characteristics and can consist of different building blocks. These differences affect the actions a creator applies to them, the value a creator locates in them, and even what counts as a single unit of the material. Traces, it appears, are less a single material than a material of differing grains [294, 295], whose individual trace data weave together into a texture [238]. A brand marker, for instance, can be seen as a coarse, discrete, artifact-facing grain a creator can name and hold. A series of planning documents for videos, on the other hand, can be seen as fine, diffuse, workflow-facing grain whose unit, whether a sentence, a section, or the whole file, has no settled boundary. This difference in grain tracks how differently the two moved (or failed to

move) through the lifecycle. Traces are therefore not a material a developer can work uniformly. Different grains serve different goals, and each has to be handled on its own terms rather than captured and surfaced the same way.

Working the material also showed the difference between what a single trace omits and what it merely holds out of easy reach. The Oracle could render how consistently a marker type recurred, since recurrence is written into the markers themselves. It could not, however, render how much a marker mattered to the creator. But that signal is not absent from a creator's traces so much as absent from the grain I was reading. Reaching it is a matter of working a finer grain, not of information the material never had. Authenticity, on the other hand, is a case of the individual threads of the material being inadequate on their own. No richer reading of an individual trace recovers it, because it lives in the texture: how a creator's traces combine, in the pattern the body of work makes as a whole. The lifecycle model let me see how individual brand marker traces can evolve, but not how traces weave. Fully leveraging traces in system design therefore requires studying both the varying grain of individual trace data and the texture traces form as they combine across creative practice.

Methodologically, inspirational bits rest on the premise that a designer comes to know a material by making something quick and partial with it by hand. However, the defining property of a trace, that it accumulates over time, cannot be easily held in hand without first acquiring high quality "ingredients" for the material. A material whose behavior appears only across months or years is one that current ways of prototyping, and current ways of evaluating, are poorly equipped to reach. Treating traces as a design material therefore asks for more ways to model the texture traces form as they combine, and ways to work the material without waiting out the timescale over which its value emerges.

8.7 Limitations

Though this work advances the understanding of longitudinal trace awareness and of working with traces, it has limitations that shaped its outcomes and that future work will need to address. The most significant of these was the lack of real longitudinal data. Traces accumulate over projects and years, which is exactly the property I could not observe directly. Both bits ran on a simulated catalog of three Final Cut Pro projects and ten generated variants rather than a creator's actual body of work, which meant I could not deploy them against creators' own histories or encounter them, as the method intends, with the real material. The same difficulty is part of why the exploration centered on two bits rather than a wider set. As a consequence, most of what I learned came from designing the bits rather than directly using them.

The formative study and the model were also limited in scope. Seven experienced creators cannot capture the full range of video content creation, and my focus on brand markers left the rest of creators' trace data examined only in passing. The model reflects this same emphasis, and the Maintenance stage in particular received less attention than the stages the bits were built around. Eliciting trace data through interviews and walkthroughs was also limited, since much of the trace data creators produce is difficult for them to articulate, and better elicitation techniques will be needed before cross-project practice can be studied more fully.

Finally, I applied a method designed for multidisciplinary teams on my own, so the bits

reflect a single designer's conversation with the material rather than the shared understanding the method was built to produce.

Chapter 9
Conclusions

In the six years of research that have gone into this dissertation, a trajectory was formed around how we might develop tools that align with the reality of creative work. This has taken me from developing tools to mitigate the constraints existing platforms place on creators, to asking how we might design process-aligned tools more broadly.

My early attempts to address the challenges creators face were domain-specific, centered on making tools that stream data as a means of circumventing known bottlenecks in existing platforms, with a particular interest in building data pipelines for real-time rendering. With V-Light, I did this through an edge computing architecture that offloaded pose estimation to the edge. What I learned here is that giving a tool better data-driven capabilities can change the kind of designs, and by extension the kind of work, a creator can do.

MARS is where I spent most of my PhD. This research arc began as an effort to open up an ecosystem that placed heavy constraints on how creators and their audiences could interact. Developing MARS to use live gameplay data showed how existing live data can be used for new purposes, and in turn to reshape the relationship between a streamer and their viewers. Most importantly, through follow-up projects, I learned that this data is never neutral, but a dynamic material that acts as both a resource and a constraint depending on which stakeholder engages with it. This recognition, that creators' data is a material they value, was the turning point toward my current trace-oriented lens.

VibeClip, similarly, was an effort to put emerging generative music models to work in a way that suited how creators preferred to express themselves when adding music to their videos. Building its context-aware pipeline taught me how much narrative context can be drawn from the very artifacts creators are in the middle of making, and how that context can be used to align a tool with the story a creator is trying to tell.

Looking back, my PhD has been a sprawling story full of its own dead ends, iterations, and problems to solve on the way to helping creators. Reflecting on this trajectory, and my own traces, if you will, led me to reorient my approach to creativity tool research away from building domain-specific tools and toward studying how we can align tool behavior with the individual, often bespoke workflows through which creators actually work.

In my latter PhD work, I examined the commonalities across the creative workflows I have studied and formalized creative activity traces as the key construct anchoring this orientation. Through a systematic review of 133 creativity tools, I categorized the data these tools use and synthesized it into the Living Framework for Trace Awareness, defining trace-aware tools in the process. My final project, a trace mapping study of longitudinal video content creation, produced a trace lifecycle model for brand markers and, through two inspirational bits, a first account of traces as a design material. Trace data is not uniform but comes in differing *grains*, and its accumulation forms a *texture* in which a creator's identity lives, one that no individual piece of trace data carries on its own. The study also showed that Elevation, the moment trace data is carried across projects, is a prediction made under uncertainty.

9.1 Personal Brand and Beyond

Although the trace mapping study was scoped to personal brand in video content creation, it was also an exploration of longitudinal trace awareness and of using traces as a design material. As such, its findings generalize along both of these lines. The longitudinal concerns participants described are not unique to video. Illustrators, designers, game developers, and even academic researchers all build *portfolios* of work that take shape over years [27]. One of the central findings of the trace mapping study was that a creator's identity does not reside in any single project, but emerges from the accumulated body of work, and this holds for any practice organized around a portfolio.

What, then, of personal brand itself? The concept of a brand is not unique to video. Designers, illustrators, and game developers often cultivate recognizable personal brands across their portfolios. Even researchers build a recognizable scholarly identity through the questions they return to and the ways they frame them. I described personal brand in Chapter 8 as the identity a creator communicates through repeated aesthetic and stylistic choices, and what distinguishes it from pure style is its orientation *toward* an audience; a brand is shaped as much by what a creator believes others expect of them as by their own preferences. Brand is therefore one of several identity-bearing qualities that matter to creators across a body of work, alongside a creator's style, voice, or taste. Its most relevant inward-facing counterpart is perhaps self-expression. That is, the value a creator places on making work because they enjoy it or because it feels like their own, regardless of how it reads to others. Participants' accounts from the trace mapping study suggest that brand and self-expression do not always move together. P1 browsed fonts simply because she enjoyed it, while P5 avoided templates because they did not feel true to herself, and the drift participants noticed in their work only in retrospect may reflect exactly this kind of divergence between a consistent brand and a changing sense of what feels like one's own. This tension is unlikely to be specific to video content creation, since any creator who presents a body of work to others must weigh the identity that work has come to signal against what they want to make next, whether that is a designer balancing client-facing work against personal projects or a researcher deciding whether to pursue a new interest that departs from an established agenda. The lifecycle model does not depend on the commercial framing of brand, however. It depends on creators deliberately carrying the same choices from one project to the next, and it is therefore better understood as a model of how trace data accumulates into identity, with brand markers as the instance this study followed.

Seen this way, the concerns of the Meaning-Making stage are likely to arise wherever creators build a body of work over time. The insights on Elevation should likewise hold wherever creators deliberately decide what to carry forward into future work, and they extend to the trace data creators do not elevate deliberately, where system-supported elevation might surface a designer's discarded explorations or a researcher's reading notes when later work makes them relevant. What changes across domains is what kind of trace data gets elevated and how it is maintained. An illustrator, for example, might carry forward a set of custom brushes, a designer a component library, and a researcher a signature figure style, each kept and reused through tools particular to its field.

The findings about traces as a material generalize in a similar way. In the trace mapping study, brand markers were concentrated in the artifact and layout channels and were produced

almost entirely within the video editor, while scripts, idea lists, and mood boards spread across the reference, annotation, and feedback channels and were often produced outside it. These are the same data characteristics the Living Framework identified across systems in eleven creative domains, which suggests that differences in grain arise wherever trace data of different channels and creation contexts accumulate side by side, and that the framework offers a vocabulary for describing them in any domain. Texture generalizes for a related reason: wherever trace data accumulates across a body of work, it interweaves into a whole that no individual piece carries on its own. Testing these findings in other portfolio-based practices, where both the units creators elevate and the grains of their trace data will differ, is a natural next step within the trace-aware research agenda that follows.

9.2 Towards a Trace-Aware Research Agenda

I conclude this dissertation with a call to action: to establish a clear *trace-aware research agenda* for creativity tool researchers. Such an agenda concerns how we collect, model, and evaluate the traces that power our tools, and, critically, how we can do so responsibly, given what this data represents about the creators who produce it. Any such agenda begins with the definition of traces that anchors it.

Throughout this dissertation, I defined creative activity traces as the process-related data generated as creators work, forming an evolving record of the decisions, explorations, and refinements that unfold over time. Treating this definition as a deliberately broad construct was advantageous for this work. It allowed me to bring together bodies of research that had studied process data separately, and to survey how creativity tools engage with that data without prescribing in advance how it should be structured or attended to. It also allowed me to begin working with traces directly as a material before settling what does or does not count as a trace. That breadth, however, left several decisions about the nature of traces to be made in the course of the work itself. The most fundamental of these is that trace *data* becomes part of a trace only as it accumulates *over time*, a commitment that runs through every chapter and that I consider non-negotiable in future work involving traces and trace-aware development.

A trace-aware research agenda, as well as any future set of best practices for building trace-aware tools, will require more precision than this broad working definition. However, that precision, I argue, is unlikely to come from a binary definition that sorts data into what is and is not part of a trace. In principle, a creator's traces encompass everything involved in producing their work, from low-level interaction logs and cursor coordinates to the drafts, references, and decisions that accumulate across a body of work. What differs is how much of that record, and at what *granularity*, a system or a researcher can meaningfully leverage. Precision therefore depends less on drawing a single boundary around traces than on being explicit about which part of the trace a given study or system engages, at what granularity, over what span of time, and for what purpose. Developing ways to make these choices explicit is a central aim of the trace-aware research agenda I lay out in the remainder of this chapter.

I first discuss the short-term agenda that is an immediate follow-up to my previous work. This is followed by what I find to be the most pressing direction for the community to chart: creating a trace-aware research toolkit, grounded in an account of trace ethics. Finally, I discuss

future applications of trace-aware tools beyond the domains investigated here.

9.2.1 Framework Refinements

The Living Framework for Trace Awareness I introduced in Chapter 7 was built to describe trace usage in existing single-project systems, and the longitudinal study in Chapter 8 exposed where it needs to grow. Several of its dimensions warrant sharpening. The channel dimension, for one, could be renamed and restructured to better denote the different grains within the trace material. The distinction between artifact-facing trace data, such as brand markers, and workflow-facing trace data, such as video planning documents, can be captured cleanly at the level of data characteristics, but its consequences are not carried through to the trace awareness dimensions, where a dimension like trace functions does not yet reflect how differently these two kinds of data behave. Refining the existing dimensions so that they accommodate shifts in temporal scope is a further necessary step, without which longitudinal traces cannot be properly incorporated into the framework.

The deeper limitation Chapter 8 exposed is that the framework does not model how trace data evolves into traces, offering instead the building blocks of data characteristics on one side and trace-aware feature possibilities on the other. Accounting for how trace data evolves into traces, and how those traces weave together into a texture, is the missing intermediate layer in our understanding of traces. This includes refining the trace lifecycle model from Chapter 8 to understand individual trace evolution, and also modeling how different traces interrelate over time. Closing this gap will require multiple *lenses* for modeling traces and their relationships, rather than a single one. Because traces encompass data that can be meaningful at very different granularities, such as a sequence of clicks within a single editing session versus a series of videos published across a long-running YouTube channel, each lens is best understood as attending to a particular part of the trace for a particular goal. The trace lifecycle model is one such lens, following brand markers at the cross-project scope in order to understand how they accumulate into a creator's identity. Linkography and its recent computational extension, Fuzzy Linkography, offer a lens at a much finer granularity, representing a creative process as a structure of links between individual moves rather than a set of isolated actions [117, 259]. Behavior Latticing [341] offers another source of inspiration for modeling how different trace data relate. It infers identity and motivation from behavioral traces, however, a larger interpretive leap than linkography's structural summaries, and it was developed for personal-AI user modeling rather than creativity tools, so it is best treated as adjacent-domain evidence rather than a method to adopt directly.

9.2.2 Trace-Aware Research Toolkit

A key argument of this dissertation is that researchers need to engage deliberately with traces throughout the research process. In this sense, trace awareness is not only a property of the tools we build, but also a stance that researchers, tool builders, and creators can take toward the traces that creative work produces. The work in Chapter 8 was a firsthand attempt at that engagement, and its lessons point toward a methodology. My work studying and building with traces made clear that specialized methods are needed to do this. To engage with traces this way, we need

methods to *elicit* traces from creators, to *model* them, to *design* with them, to *develop* systems around them, and to *evaluate* the tools that result.

For elicitation, better methods are needed, since the interviews and walkthroughs I relied on in Chapter 8 were poorly suited to surfacing trace data that creators themselves struggle to articulate. Participatory methods such as journey mapping and co-mapping with creators, and in-situ methods such as experience sampling that pair logs with in-the-moment explanations, are promising starting points.

For modeling, the Living Framework and the lenses described above offer a starting point for characterizing and comparing traces. For design, a deliberate engagement approach works bottom-up, starting from traces rather than features, and calls for generative frameworks that connect feature behavior to trace data. To develop future trace-aware tools, we need scalable trace-aware data pipelines that would make trace-aware features practical to build. A critical aspect of the toolkit that was not addressed in this dissertation is evaluation. The field has almost no way to isolate a trace and measure its effect on a system or a creator. One path is akin to an ablation study: holding a feature fixed while swapping in trace data of different grains, or accumulations with different textures, and observing what changes. Establishing metrics, evaluation frameworks, and simulation or experimental designs for doing so is essential. Taken together, these pieces begin to form a trace-aware research methodology and a systematic way to conduct creativity tool research in this space while keeping traces at the center.

9.2.3 Trace Ethics

None of the above should be pursued without an intentional account of what I call trace ethics. Traces are intimate creator data, intertwined with the people who produce them, and those creators already feel their work and livelihoods under threat. This concern is not hypothetical: most systems in the Tracing Creativity review did not make clear what they captured or how it was used. Collecting creator data is now almost trivial, but the trace-aware future I advocate for is a pro-creator one, with consent, ownership, and control built in. This means using traces intentionally and within a genuine ethical structure.

This includes meaningful user control over what is captured and how it is used, customizable rather than blanket trace collection, and a habit of interrogating capture itself by asking, for any trace-aware feature, when we should collect, what we should collect, for what purpose, and at what cost. Answering those questions well will mean working with creators and developers to build real ethical frameworks around ownership, consent, and power, rather than prescribing them from the outside.

9.2.4 Future Applications

Finally, trace awareness has the potential to reach well beyond the creative communities this thesis studied. The trace lifecycle model should offer some purchase wherever identity accretes across projects, including not only other creative media but writing, software and game development, data visualization, and more. Two shifts make this reach pressing rather than merely possible. As generative systems absorb more of the creative pipeline, agent experience design

becomes a trace-aware problem in its own right, one that calls for trace-aware user profiles, adaptive agent behavior grounded in a creator's history, and ways of disentangling a human's traces from an agent's. The same orientation can also be turned on agents themselves, treating what an agent does as traceable activity that could support transparency in human-agent collaboration. As creative work grows more collaborative, trace awareness may help translate across disciplines and hold a shared body of work together, provided it does so without collapsing collaboration into hyper-individualized workflows. Across all of these, the through line is the question this dissertation began with. Traces are how we keep the creator visible in the systems we build, and designing those systems around traces is how we ensure the tools of the future align with the reality of creative work, supporting creative practice rather than merely its outputs.

References

[1] Easily create apps for PC games on the overwolf framework. https://overwolf.github.io/, 2011. Accessed: 2023-7-21. 4.2.1.2

[2] Accessible player experiences (APX). https://accessible.games/accessible-player-experiences/, May 2018. Accessed: 2023-4-5. 4.4.2

[3] Tweening to create animated action. https://www.adobe.com/creativecloud/video/discover/tweening.html, 2021. Accessed: 2023-7-24. 4.3.2

[4] Open broadcast software, 2021. URL https://obsproject.com/. 4.3.1

[5] Twitchtracker statistics and charts, 2021. URL https://twitchtracker.com/statistics. 4.1

[6] Kaho Abe and Katherine Isbister. Hotaru: The lightning bug game. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '16, page 277–280, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450340823. doi: 10.1145/2851581.2889472. URL https://doi.org/10.1145/2851581.2889472. 3.2.1

[7] Elie Abi Saad and Marine Agogué. Creativity in virtual teams: Systematic review, synthesis and research agenda. *Creativity & Innovation Management*, 32(1):117–140, 2023. doi: 10.1111/caim.12540. URL https://doi.org/10.1111/caim.12540. 7.2.1

[8] Adobe. Seeds - adobe firefly API, 2024. URL https://developer.adobe.com/firefly-services/docs/firefly-api/guides/concepts/seeds/. Accessed: 2025-1-17. 6.4.3.4

[9] Adobe. Advanced search features, 2025. URL https://stock.adobe.com/search-features. Accessed: 2025-1-10. 6.2.2

[10] Andrea Agostinelli, Timo I. Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matt Sharifi, Neil Zeghidour, and Christian Frank. Musiclm: Generating music from text, 2023. URL https://arxiv.org/abs/2301.11325. 6.1, 6.2.3

[11] Karan Ahuja, Mayank Goel, and Chris Harrison. Bodyslam: Opportunistic user digitization in multi-user ar/vr experiences. In *Symposium on Spatial User Interaction*, SUI '20, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450379434. doi: 10.1145/3385959.3418452. URL https://doi.org/10.

1145/3385959.3418452. 3.6

[12] Riffusion AI. Riffusion, 2024. URL https://www.riffusion.com/. 6.2.3

[13] Stability AI. Stable audio, 2023. URL https://stability.ai/stable-audio. 6.2.3, 6.4.2, 6.5.1

[14] Stability AI. Stable audio - user guide, 2025. URL https://stableaudio.com/user-guide/prompt-structure. Accessed: 2025-01-19. 6.2.3, 6.4.3.3

[15] Alberto Alvarez, Jose Font, and Julian Togelius. Toward designer modeling through design style clustering. *IEEE Transactions on Games*, 14(4):676–686, 2022. 2.3, 7.5.2.1

[16] Amazon. Twitch.tv, 2021. URL https://www.twitch.tv/. 5.1

[17] Amazon Web Services. Amazon ec2 concepts. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html, 2021. Accessed: April 5, 2023. 4.3.4

[18] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N. Bennett, Kori Inkpen, Jaime Teevan, Ruth Kikin-Gil, and Eric Horvitz. Guidelines for human-ai interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–13, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450359702. doi: 10.1145/3290605.3300233. URL https://doi.org/10.1145/3290605.3300233. 6.5.3, 8.4

[19] Barrett R Anderson, Jash Hemant Shah, and Max Kreminski. Homogenization effects of large language models on human creative ideation. In *Proceedings of the 16th conference on creativity & cognition*, pages 413–425, 2024. 2.1

[20] Tyler Angert, Miroslav Suzara, Jenny Han, Christopher Pondoc, and Hariharan Subramonyam. Spellburst: A Node-based Interface for Exploratory Creative Coding with Natural Language Prompts. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 979-8-4007-0132-0. doi: 10.1145/3586183.3606719. 7.3.1.2, 7.4.1.1, **??**, 7.4.1.4, 7.4.2.1, 7.4.2.3, 7.4.2.4

[21] Anna Anthropy and Naomi Clark. *A Game Design Vocabulary: Exploring the Foundational Principles Behind Good Game Design*. Addison-Wesley Professional, 2014. 5.4.1

[22] Artlist.io. Artlist.io, 2016. URL https://artlist.io. Accessed: 2025-1-19. 6.1, 6.2.2, 6.4.3.3

[23] Stephen Bach, Victor Sanh, Zheng Xin Yong, Albert Webson, Colin Raffel, Nihal V. Nayak, Abheesht Sharma, Taewoon Kim, M Saiful Bari, Thibault Fevry, Zaid Alyafeai, Manan Dey, Andrea Santilli, Zhiqing Sun, Srulik Ben-david, Canwen Xu, Gunjan Chhablani, Han Wang, Jason Fries, Maged Al-shaibani, Shanya Sharma, Urmish Thakker, Khalid Almubarak, Xiangru Tang, Dragomir Radev, Mike Tian-jian Jiang, and Alexander Rush. PromptSource: An integrated development environment and repository for natural language prompts. In Valerio Basile, Zornitsa Kozareva, and Sanja Stajner, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational*

Linguistics: System Demonstrations, pages 93–104, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-demo.9. URL https://aclanthology.org/2022.acl-demo.9/. 6.2.1, 6.2.3

[24] Khaled Bachour, Richard Wetzel, Martin Flintham, Trung Dong Huynh, Tom Rodden, and Luc Moreau. Provenance for the people: An hci perspective on the w3c prov standard through an online game. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, page 2437–2446, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450331456. doi: 10.1145/2702123.2702455. URL https://doi.org/10.1145/2702123.2702455. 2.2

[25] Syed Balkhi. 2025's creator economy statistics that will blow you away, September 2023. URL https://www.wpbeginner.com/research/creator-economy-statistics-that-will-blow-you-away/. Accessed: 2025-1-19. 6.1

[26] Robert Ball. Don't search, just show me what i did: Visualizing provenance of documents and applications. *International Journal of Human–Computer Interaction*, 29(3):156–168, 2013. doi: 10.1080/10447318.2012.701569. URL https://doi.org/10.1080/10447318.2012.701569. 2.2

[27] Moshe Barak and Yaron Doppelt. Using portfolios to enhance creative thinking. *The Journal of Technology Studies*, 26(2):66–77, 2000. 7.1, 9.1

[28] Aadit Barua, Karim Benharrak, Meng Chen, Mina Huh, and Amy Pavel. Lotus: Creating Short Videos From Long Videos With Abstractive and Extractive Summarization. In *Proceedings of the 30th International Conference on Intelligent User Interfaces*, IUI '25, pages 967–981, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 979-8-4007-1306-4. doi: 10.1145/3708359.3712090. 7.4.1.1, **??**, 7.4.1.2, 7.4.2.2

[29] Patrick Baudisch, Henning Pohl, Stefanie Reinicke, Emilia Wittmers, Patrick Lühne, Marius Knaust, Sven Köhler, Patrick Schmidt, and Christian Holz. Imaginary reality basketball: A ball game without a ball. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '14, page 575–578, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450324748. doi: 10.1145/2559206.2574813. URL https://doi.org/10.1145/2559206.2574813. 3.2.1

[30] Studio Bean. *Choice Chamber*. Game [Steam], July 2015. Studio Bean. Last played October 2016. 5.2.1

[31] Jen Beeston, Christopher Power, Paul Cairns, and Mark Barlet. Accessible player experiences (APX): The players. In *Computers Helping People with Special Needs*, pages 245–253. Springer International Publishing, 2018. 4.4.2, 4.5.1

[32] Isaac William Bell. *Creators, Audiences, and New Media: Creativity in an Interactive Environment*. PhD thesis, University of Kansas, 2014. 1

[33] besthttp. Best HTTP/2. https://assetstore.unity.com/packages/tools/network/best-http-2-155981?aid=1101lfX8E&utm_source=aff, 2020. Accessed: 2021-4-7. 3.3.2

[34] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins, 2020. 3.6

[35] Po Bhattacharyya, Yein Jo, Ketki Jadhav, Radha Nath, and Jessica Hammer. Brick: A synchronous multiplayer augmented reality game for mobile phones. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI EA '19, page 1–4, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450359719. doi: 10.1145/3290607.3313257. URL https://doi.org/10.1145/3290607.3313257. 3.2.1, 3.5.1.1

[36] Po Bhattacharyya, Radha Nath, Yein Jo, Ketki Jadhav, and Jessica Hammer. Brick: Toward a model for designing synchronous colocated augmented reality games. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–9, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450359702. doi: 10.1145/3290605.3300553. URL https://doi.org/10.1145/3290605.3300553. 3.1, 3.2.1, 3.3.1, 3.3.3

[37] Blizzard. *Hearthstone*. Game [Multiple Platforms], March 2014. 5.1, 5.2.1

[38] Stephen Brade, Bryan Wang, Mauricio Sousa, Gregory Lee Newsome, Sageev Oore, and Tovi Grossman. SynthScribe: Deep Multimodal Tools for Synthesizer Sound Retrieval and Exploration. In *Proceedings of the 29th International Conference on Intelligent User Interfaces*, IUI '24, pages 51–65, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 979-8-4007-0508-3. doi: 10.1145/3640543.3645158. 7.4.2.2

[39] V Braun and V Clarke. Using thematic analysis in psychology. *Qual. Res. Psychol.*, 2006. 6.3.3, 8.3.3

[40] Alexandra Bremers and Wendy Ju. Can machines tell what people want? bringing situated intelligence to generative ai. In *Proceedings of the Halfway to the Future Symposium*, pages 1–6, 2024. 2.3, 7.5.2.3

[41] Blake Brittain. Music labels sue AI companies suno, udio for US copyright infringement. *Reuters*, June 2024. URL https://www.reuters.com/technology/artificial-intelligence/music-labels-sue-ai-companies-suno-udio-us-copyright-infringement-2024 6.7.1

[42] Louis L. Bucciarelli. Between thought and object in engineering design. *Design Studies*, 23(5):385–403, 2002. doi: 10.1016/S0142-694X(01)00052-4. 1.1, 7.1

[43] Jie Cai, Donghee Yvette Wohn, and Mashael Almoqbel. Moderation visibility: Mapping the strategies of volunteer moderators in live streaming micro communities. In *Proceedings of the 2021 ACM International Conference on Interactive Media Experiences*, IMX '21, pages 61–72, New York, NY, USA, June 2021. Association for Computing Machinery. 5.2.2

[44] Cambridge Dictionary. Vibe definition, 2025. URL https://dictionary.cambridge.org/dictionary/english/vibe. 6.3.3.2

[45] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose:

realtime multi-person 2d pose estimation using part affinity fields. *IEEE transactions on pattern analysis and machine intelligence*, 43(1):172–186, 2019. 3.3.2, 3.3.3

[46] Vincent Cavez, Catherine Letondal, Caroline Appert, and Emmanuel Pietriga. EuterPen: Unleashing Creative Expression in Music Score Writing. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 979-8-4007-1394-1. doi: 10.1145/3706598.3713488. 7.4.1.1, 7.4.1.3

[47] Tuhin Chakrabarty, Vishakh Padmakumar, Faeze Brahman, and Smaranda Muresan. Creativity support in the age of large language models: An empirical study involving emerging writers, 2024. URL https://arxiv.org/abs/2309.12570. 6.2.1

[48] Minsuk Chang, Ben Lafreniere, Juho Kim, George Fitzmaurice, and Tovi Grossman. Workflow graphs: A computational model of collective task strategies for 3d design software. In *Graphics Interface 2020*, 2020. 2.3

[49] Sven Charleer, Kathrin Gerling, Francisco Gutiérrez, Hans Cauwenbergh, Bram Luycx, and Katrien Verbert. Real-Time dashboards to support esports spectating. In *Proceedings of the 2018 Annual Symposium on Computer-Human Interaction in Play*, CHI PLAY '18, pages 59–71, New York, NY, USA, October 2018. Association for Computing Machinery. 4.5.1, 5.2.1

[50] D. Chatzopoulos, C. Bermejo, Z. Huang, and P. Hui. Mobile augmented reality survey: From where we are to where we go. *IEEE Access*, 5:6917–6950, 2017. doi: 10.1109/ACCESS.2017.2698164. 3.1

[51] Siddhartha Chaudhuri and Vladlen Koltun. Data-driven suggestions for creativity support in 3d modeling. In *ACM SIGGRAPH Asia 2010 papers*, pages 1–10. 2010. 2.3

[52] Ke Chen, Yusong Wu, Haohe Liu, Marianna Nezhurina, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Musicldm: Enhancing novelty in text-to-music generation using beat-synchronous mixup strategies. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1206–1210, 2024. doi: 10.1109/ICASSP48485.2024.10447265. 6.1, 6.2.3

[53] Tianying Chen, Jessica Hammer, and Laura Dabbish. Self-efficacy-based game design to encourage security behavior online. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI EA '19, page 1–6, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450359719. doi: 10.1145/3290607.3312935. URL https://doi.org/10.1145/3290607.3312935. 5.4

[54] Yan Chen, Walter S Lasecki, and Tao Dong. Towards supporting programming education at scale via live streaming. *Proceedings of the ACM on Human-Computer Interaction*, 4 (CSCW3):1–19, 2021. 5.2.2

[55] Zhilong Chen, Hancheng Cao, Yuting Deng, Xuan Gao, Jinghua Piao, Fengli Xu, Yu Zhang, and Yong Li. *Learning from Home: A Mixed-Methods Analysis of Live Streaming Based Remote Education Experience in Chinese Colleges during the COVID-19 Pandemic*. Association for Computing Machinery, New York, NY, USA, 2021. ISBN 9781450380966. URL https://doi.org/10.1145/3411764.3445428. 4.5.3

[56] Zhuo Chen, Wenlu Hu, Junjue Wang, Siyan Zhao, Brandon Amos, Guanhang Wu, Kiryong Ha, Khalid Elgazzar, Padmanabhan Pillai, Roberta Klatzky, Daniel Siewiorek, and Mahadev Satyanarayanan. An empirical study of latency in an emerging class of edge computing applications for wearable cognitive assistance. In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, SEC '17, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450350877. doi: 10.1145/3132211.3134458. URL https://doi.org/10.1145/3132211.3134458. 3.3.2

[57] Erin Cherry and Celine Latulipe. Quantifying the creativity support of digital tools through the creativity support index. *ACM Trans. Comput.-Hum. Interact.*, 21(4), June 2014. ISSN 1073-0516. doi: 10.1145/2617588. URL https://doi.org/10.1145/2617588. 6.5.3

[58] Gifford Cheung and Jeff Huang. Starcraft from the stands: understanding the game spectator. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 763–772, New York, NY, USA, May 2011. Association for Computing Machinery. 4.2.2, 4.4, 4.5.1, 5.6.2

[59] Jiin Choi, Seung Won Lee, and Kyung Hoon Hyun. GenPara: Enhancing the 3D Design Editing Process by Inferring Users' Regions of Interest with Text-Conditional Shape Parameters. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 979-8-4007-1394-1. doi: 10.1145/3706598.3713502. 7.4.1.4

[60] Jean-Peïc Chou, Alexa Fay Siu, Nedim Lipka, Ryan Rossi, Franck Dernoncourt, and Maneesh Agrawala. TaleStream: Supporting Story Ideation with Trope Knowledge. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 979-8-4007-0132-0. doi: 10.1145/3586183.3606807. 7.4.1.1, 7.4.1.4

[61] Sanjoy Chowdhury, Sayan Nag, K J Joseph, Balaji Vasan Srinivasan, and Dinesh Manocha. Melfusion: Synthesizing music from image and language cues using diffusion models, 2024. URL https://arxiv.org/abs/2406.04673. 6.2.3

[62] John Joon Young Chung and Eytan Adar. PromptPaint: Steering Text-to-Image Generation Through Paint Medium-like Interactions. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 979-8-4007-0132-0. doi: 10.1145/3586183.3606777. 7.4.2.4

[63] John Joon Young Chung, Shiqing He, and Eytan Adar. The intersection of users, roles, interactions, and technologies in creativity support tools. In *Proceedings of the 2021 ACM Designing Interactive Systems Conference (DIS '21)*, pages 1817–1833, Virtual Event, USA, 2021. ACM. doi: 10.1145/3461778.3462050. 7.2.1, 7.3.1

[64] John Joon Young Chung, Hijung Valentina Shin, Haijun Xia, Li-Yi Wei, and Rubaiat Habib Kazi. Beyond show of hands: Engaging viewers via expressive and scalable visual communication in live streaming. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, number Article 109 in CHI '21, pages 1–14, New

York, NY, USA, May 2021. Association for Computing Machinery. 4.2.1.1, 5.2.1

[65] John Joon Young Chung, Wooseok Kim, Kang Min Yoo, Hwaran Lee, Eytan Adar, and Minsuk Chang. Talebrush: Sketching stories with generative pretrained language models. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450391573. doi: 10.1145/3491102.3501819. URL https://doi.org/10.1145/3491102.3501819. 6.2.1

[66] Miguel Civit, Javier Civit-Masot, Francisco Cuadrado, and Maria J. Escalona. A systematic review of artificial intelligence-based music generation: Scope, applications, and future trends. *Expert Systems with Applications*, 209:118190, 2022. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2022.118190. URL https://www.sciencedirect.com/science/article/pii/S0957417422013537. 6.2.3

[67] M. Claypool and D. Finkel. The effects of latency on player performance in cloud-based games. In *2014 13th Annual Workshop on Network and Systems Support for Games*, pages 1–6, 2014. doi: 10.1109/NetGames.2014.7008964. 3.3.3

[68] Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. Simple and controllable music generation, 2024. URL https://arxiv.org/abs/2306.05284. 6.1, 6.2.3

[69] Nick Couldry. Liveness, "reality," and the mediated habitus from television to the mobile phone. *The Communication Review*, 7(4):353–361, October 2004. 4.2.2

[70] Alan B. Craig. Chapter 7 - mobile augmented reality. In Alan B. Craig, editor, *Understanding Augmented Reality*, pages 209–220. Morgan Kaufmann, Boston, 2013. ISBN 978-0-240-82408-6. doi: https://doi.org/10.1016/B978-0-240-82408-6.00007-2. URL https://www.sciencedirect.com/science/article/pii/B9780240824086000072. 3.1, 3.2.1

[71] Jenny Crawford. 2023 tiktok data report: 85% of videos on tiktok contain music, February 2024. URL https://pex.com/blog/2023-tiktok-data-report-85-of-videos-on-tiktok-contain-music/. Accessed: 2025-1-19. 6.1

[72] Ella Dagan, Elena Márquez Segura, Ferran Altarriba Bertran, Miguel Flores, and Katherine Isbister. Designing 'true colors': A social wearable that affords vulnerability. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–14, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450359702. doi: 10.1145/3290605.3300263. URL https://doi.org/10.1145/3290605.3300263. 3.5.1.3

[73] Ana Fiona Dalipi, Dongfang Liu, Xiaolei Guo, Yingjie Chen, and Christos Mousas. Vrpavib: The virtual reality pedestrian-autonomous vehicle interaction benchmark. In *12th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, AutomotiveUI '20, page 38–41, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450380669. doi: 10.1145/3409251.3411718. URL https://doi.org/10.1145/3409251.3411718. 3.5.1.3

[74] Ana Fiona Dalipi, Dongfang Liu, Xiaolei Guo, Yingjie Chen, and Christos Mousas. Vr-pavib: The virtual reality pedestrian-autonomous vehicle interaction benchmark. In *12th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, AutomotiveUI '20, page 38–41, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450380669. doi: 10.1145/3409251.3411718. URL https://doi.org/10.1145/3409251.3411718. 3.5.1.3

[75] Hai Dang, Lukas Mecke, Florian Lehmann, Sven Goller, and Daniel Buschek. How to prompt? opportunities and challenges of zero- and few-shot learning for human-ai interaction in creative applications of generative models, 2022. URL https://arxiv.org/abs/2209.01390. 6.1, 6.2.1, 6.2.3

[76] Nigel Davies, Nina Taft, Mahadev Satyanarayanan, Sarah Clinch, and Brandon Amos. Privacy mediators: Helping iot cross the chasm. In *Proceedings of the 17th International Workshop on Mobile Computing Systems and Applications*, pages 39–44, 2016. 3.2.2

[77] Nicholas Davis, Chih-PIn Hsiao, Kunwar Yashraj Singh, Lisa Li, and Brian Magerko. Empirically studying participatory sense-making in abstract drawing with a co-creative cognitive agent. In *Proceedings of the 21st international conference on intelligent user interfaces*, pages 196–207, 2016. 2.3, **??**, 7.4.2.2

[78] Nicholas Davis, Chih-Pin Hsiao, Kunwar Yashraj Singh, Brenda Lin, and Brian Magerko. Creative sense-making: Quantifying interaction dynamics in co-creation. In *Proceedings of the 2017 ACM SIGCHI Conference on Creativity and Cognition*, pages 356–366, 2017. 2.3, 7.4.2.3

[79] Adrian A. de Freitas, Michael Nebeling, Xiang 'Anthony' Chen, Junrui Yang, Akshaye Shreenithi Kirupa Karthikeyan Ranithangam, and Anind K. Dey. *Snap-To-It: A User-Inspired Platform for Opportunistic Device Interactions*, page 5909–5920. Association for Computing Machinery, New York, NY, USA, 2016. ISBN 9781450333627. URL https://doi.org/10.1145/2858036.2858177. 3.5.2.3

[80] B. Dhiman. A paradigm shift in the entertainment industry in the digital age: A critical review. *Global Media Journal*, 21(64):384, 2023. doi: 10.36648/1550-7521.21.64.384. URL https://www.globalmediajournal.com/open-access/a-paradigm-shift-in-the-entertainment-industry-in-the-digital-age-a-cr pdf. Accessed: 2025-10-09. 1

[81] Shangzhe Di, Zeren Jiang, Si Liu, Zhaokai Wang, Leyan Zhu, Zexin He, Hongming Liu, and Shuicheng Yan. Video background music generation with controllable music transformer. In *Proceedings of the 29th ACM International Conference on Multimedia*, MM '21, page 2037–2045, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450386517. doi: 10.1145/3474085.3475195. URL https://doi.org/10.1145/3474085.3475195. 6.2.4, 6.6.1

[82] Vaibhav Diwanji, Abigail Reed, Arienne Ferchaud, Jonmichael Seibert, Victoria Weinbrecht, and Nicholas Sellers. Don't just watch, join in: Exploring information behavior and copresence on twitch. *Comput. Human Behav.*, 105:106221, April 2020. 4.2.2

[83] Shachar Don-Yehiya, Leshem Choshen, and Omri Abend. Human learning by model

feedback: The dynamics of iterative prompting with midjourney. *arXiv preprint arXiv:2311.12131*, 2023. 2.3

[84] Chris Donahue, Antoine Caillon, Adam Roberts, Ethan Manilow, Philippe Esling, Andrea Agostinelli, Mauro Verzetti, Ian Simon, Olivier Pietquin, Neil Zeghidour, and Jesse Engel. Singsong: Generating musical accompaniments from singing, 2023. URL https://arxiv.org/abs/2301.12662. 6.2.3

[85] Zhikang Dong, Bin Chen, Xiulong Liu, Pawel Polak, and Peng Zhang. Musechat: A conversational music recommendation system for videos, 2024. URL https://arxiv.org/abs/2310.06282. 6.2.2

[86] Steven Dow, Julie Fortuna, Dan Schwartz, Beth Altringer, Daniel Schwartz, and Scott Klemmer. Prototyping dynamics: sharing multiple designs improves exploration, group rapport, and results. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 2807–2816, New York, NY, USA, May 2011. Association for Computing Machinery. 5.3.2

[87] S. Echeverría, D. Klinedinst, K. Williams, and G. A. Lewis. Establishing trusted identities in disconnected edge environments. In *2016 IEEE/ACM Symposium on Edge Computing (SEC)*, pages 51–63, 2016. doi: 10.1109/SEC.2016.27. 3.2.2

[88] Edgegap. Edgegap's documentation. https://docs.edgegap.com/, 2018. Accessed: 2021-4-7. 3.2.2

[89] M. S. Elbamby, C. Perfecto, M. Bennis, and K. Doppler. Toward low-latency and ultra-reliable virtual reality. *IEEE Network*, 32(2):78–84, 2018. doi: 10.1109/MNET.2018.1700268. 3.1, 3.2.1

[90] Chris Elsden, David S. Kirk, and Abigail C. Durrant. A quantified past: Toward design for remembering with personal informatics. *Human-Computer Interaction*, 31(6):518–557, 2016. doi: 10.1080/07370024.2015.1093422. 2.3

[91] Zach Evans, Julian D. Parker, CJ Carr, Zack Zukowski, Josiah Taylor, and Jordi Pons. Long-form music generation with latent diffusion, 2024. URL https://arxiv.org/abs/2404.10301. 6.2.3

[92] Zach Evans, Julian D. Parker, CJ Carr, Zack Zukowski, Josiah Taylor, and Jordi Pons. Stable audio open, 2024. URL https://arxiv.org/abs/2407.14358. 6.1, 6.2.3

[93] Travis Faas, Lynn Dombrowski, Alyson Young, and Andrew D Miller. Watch me code: Programming mentorship communities on twitch. tv. *Proceedings of the ACM on Human-Computer Interaction*, 2(CSCW):1–18, 2018. 4.1, 4.5.3

[94] Die Gutie Fabrik. *Johann Sebastian Jouse*. Game, June 2013. 3.5.1.1

[95] Triangle Factory. *AR Sports Basketball*. Game [Mobile], September 2017. Triangle Factory. 3.2.1

[96] Cong Fang, Yujie Zhu, Le Fang, Yonghao Long, Huan Lin, Yangfan Cong, and Stephen Jia Wang. Generative ai-enhanced human-ai collaborative conceptual design: A systematic literature review. *Design Studies*, 101:101300, 2025. doi: 10.1016/j.destud.2025.101300. URL https://www.sciencedirect.com/science/article/pii/

S0142694X25000122. 7.2.1

[97] Xiaoxuan Fang, Davy Tsz Kit Ng, Jac Ka Lok Leung, and Samuel Kai Wah Chu. A systematic review of artificial intelligence technologies used for story writing. *Education and Information Technologies*, 28(11):14361–14397, 2023. doi: 10.1007/s10639-023-11741-5. URL https://doi.org/10.1007/s10639-023-11741-5. 7.2.1

[98] Melanie Feinberg. A design perspective on data. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, page 2952–2963, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450346559. doi: 10.1145/3025453.3025837. URL https://doi.org/10.1145/3025453.3025837. 8.2.2

[99] Therese Fessenden. When remote workshops fail. https://www.nngroup.com/articles/remote-workshop-fail/, September 2020. Accessed: 2024-2-22. 5.3.2

[100] Claudia Flores-Saviaga, Jessica Hammer, Juan Pablo Flores, Joseph Seering, Stuart Reeves, and Saiph Savage. Audience and streamer participation at scale on twitch. In *Proceedings of the 30th ACM Conference on Hypertext and Social Media*, HT '19, pages 277–278, New York, NY, USA, September 2019. Association for Computing Machinery. 1, 4.1

[101] Colin Ford, Dan Gardner, Leah Elaine Horgan, Calvin Liu, A m Tsaasan, Bonnie Nardi, and Jordan Rickman. Chat speed OP PogChamp: Practices of coherence in massive twitch chat. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '17, pages 858–871, New York, NY, USA, May 2017. Association for Computing Machinery. 4.2.1.1, 4.2.2, 4.4

[102] C Ailie Fraser, Joy O Kim, Alison Thornsberry, Scott Klemmer, and Mira Dontcheva. Sharing the studio: How creative livestreaming can inspire, educate, and engage. In *Proceedings of the 2019 on Creativity and Cognition*, pages 144–155. ACM, 2019. 4.1, 4.5.3, 5.2.2

[103] Jonas Frich, Michael Mose Biskjaer, and Peter Dalsgaard. Why hci and creativity research must collaborate to develop new creativity support tools. In *Proceedings of the Technology, Mind, and Society*, TechMindSociety '18, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450354202. doi: 10.1145/3183654.3183678. URL https://doi.org/10.1145/3183654.3183678. 2.2

[104] Jonas Frich, Lindsay MacDonald Vermeulen, Christian Remy, Michael Mose Biskjaer, and Peter Dalsgaard. Mapping the landscape of creativity support tools in hci. In *Proceedings of the 2019 CHI conference on human factors in computing systems*, pages 1–18, 2019. 2.1, 7.1, 7.2.1, 7.3.1.2, 7.5.1, 7.5.2.4

[105] Jonas Frich Pedersen, Michael Mose Biskjær, and Peter Dalsgaard. Twenty years of creativity research in human-computer interaction: Current state and future directions. In *Proceedings of the 2018 Designing Interactive Systems Conference*, pages 1235–1257, United States, 2018. ACM. doi: 10.1145/3196709.3196732. 2.1, 2.2, 7.2.1

[106] Emma Frid, Celso Gomes, and Zeyu Jin. Music creation by example. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–13, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450367080. doi: 10.1145/3313831.3376514. URL https://doi.org/10.1145/3313831.3376514. 6.1, 6.2, 6.2.1, 6.2.2, 6.2.3, 6.2.4, 6.3.3.1, 6.3.3.2, 6.3.3.4, 6.4.1, 6.5.3, 6.6.2

[107] Emma Frid, Celso Gomes, and Zeyu Jin. Music Creation by Example. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, pages 1–13, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 978-1-4503-6708-0. doi: 10.1145/3313831.3376514. 7.4.1.1, 7.4.1.2

[108] Alex Gabriel, Davy Monticolo, Mauricio Camargo, and Mario Bourgault. Creativity support systems: A systematic mapping study. *Thinking Skills and Creativity*, 21:109–122, 2016. doi: 10.1016/j.tsc.2016.05.009. URL https://www.sciencedirect.com/science/article/pii/S1871187116300293. 7.2.1

[109] Firaxis Games. *Sid Meier's Civilization VI*. Game [Multiple Platforms], October 2016. 5.4

[110] Matt Makes Games. Celeste. Windows, Nintendo Switch, PlayStation 4, Xbox One, 2018. 4.4.2

[111] Amrita Ganguly, Chuan Yan, John Joon Young Chung, Tong Steven Sun, Yoon Kiheon, Yotam Gingold, and Sungsoo Ray Hong. ShadowMagic: Designing Human-AI Collaborative Support for Comic Professionals' Shadowing. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, UIST '24, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 979-8-4007-0628-8. doi: 10.1145/3654777.3676332. 7.4.2.2, 7.4.2.4

[112] Aron Garst. Grounded Lets You Make The Spiders Less Terrifying If You Have Arachnophobia, 2020. URL https://www.gamespot.com/articles/grounded-lets-you-make-the-spiders-less-terrifying/1100-6480035/. 4.6

[113] S. George, T. Eiszler, R. Iyengar, H. Turki, Z. Feng, J. Wang, P. Pillai, and M. Satyanarayanan. Openrtist: End-to-end benchmarking for edge computing. *IEEE Pervasive Computing*, 19(4):10–18, 2020. doi: 10.1109/MPRV.2020.3028781. 3.1, 3.2.2, 3.3.2

[114] John S Gero. Fixation and commitment while designing and its measurement. *The Journal of Creative Behavior*, 45(2):108–115, 2011. 2.3

[115] Sarah Gibbons. Journey mapping 101. https://www.nngroup.com/articles/journey-mapping-101/, 2018. Nielsen Norman Group. Retrieved September 3, 2026. 8.4

[116] Seth Glickman, Nathan McKenzie, Joseph Seering, Rachel Moeller, and Jessica Hammer. Design challenges for livestreamed audience participation games. In *Proceedings of the 2018 Annual Symposium on Computer-Human Interaction in Play*, CHI PLAY '18, pages 187–199, New York, NY, USA, October 2018. Association for Computing Machinery. 4.1, 4.2.1.1, 4.2.2, 5.1, 5.2.1, 5.2.2, 5.4.3, 5.4.3.1

[117] Gabriela Goldschmidt. *Linkography: unfolding the design process*. Mit Press, 2014. 1.1, 2.3, 7.1, 9.2.1

[118] Alejandra Gomez Ortega, Peter Lovei, Renee Noortman, Romain Toebosch, Alex Bowyer, Albrecht Kurze, Mathias Funk, Sandy J. J. Gould, Samuel Huron, and Jacky Bourgeois. Sig on data as human-centered design material. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI EA '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450394222. doi: 10.1145/3544549.3583180. URL https://doi.org/10.1145/3544549.3583180. 8.2.2

[119] Google. Lightboard:an ar multiplayer game to demonstrate the cloud anchors api. https://opensource.google/projects/arcore-lightboard, 2018. Accessed: 2021-4-6. 3.2.1

[120] Google. Swiftshot: Creating a game for augmented reality. https://developer.apple.com/documentation/arkit/swiftshot_creating_a_game_for_augmented_reality, 2018. Accessed: 2021-4-6. 3.2.1

[121] Google LLC. Google translate api. https://cloud.google.com/translate, 2021. Accessed: April 5, 2023. 4.4.2

[122] Daniel Gros, Brigitta Wanner, Anna Hackenholt, Piotr Zawadzki, and Kathrin Knautz. World of streaming. motivation and gratification on twitch. In *Social Computing and Social Media. Human Behavior*, pages 44–57. Springer International Publishing, 2017. 4.2.2

[123] Aleksandra Gruszka and Min Tang. The 4p's creativity model and its application in different fields. In *Handbook of the management of creativity and innovation: Theory and practice*, pages 51–71. World Scientific, 2017. 1.1, 7.1

[124] Joy Paul Guilford. Creativity. *American psychologist*, 5(9), 1950. 2.2

[125] Rebecca Gulotta, Alex Sciuto, Aisling Kelliher, and Jodi Forlizzi. Curatorial agents: How systems shape our understanding of personal and familial digital information. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, page 3453–3462, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450331456. doi: 10.1145/2702123.2702297. URL https://doi.org/10.1145/2702123.2702297. 2.2, 2.3

[126] Fatih Gumus. Understanding time-based media: Examples, types, sources, and representation. https://dailyart.news/art-terms/understanding-time-based-media-examples-types-sources-and-representati 2024. URL https://dailyart.news/art-terms/understanding-time-based-media-examples-types-sources-and-representati Accessed: 2025-10-09. 1

[127] Kiryong Ha, Zhuo Chen, Wenlu Hu, Wolfgang Richter, Padmanabhan Pillai, and Mahadev Satyanarayanan. Towards Wearable Cognitive Assistance. In *Proceedings of the Twelfth International Conference on Mobile Systems, Applications, and Services*, Bretton Woods, NH, June 2014. 3.3.2

[128] Emily Halpin. Introducing extensions: A streaming revolution, 2020. URL https://blog.twitch.tv/en/2017/08/31/introducing-extensions-a-streaming-revolution-c31762addcd9/. 4.2.1.2, 4.3.5, 5.1, 5.2.1

[129] William A Hamilton, Oliver Garretson, and Andruid Kerne. Streaming on twitch: fostering participatory communities of play within live mixed media. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1315–1324, 2014. 1, 4.2.2, 4.4, 4.5.2, 5.6.2

[130] Noor Hammad, Erik Harpstead, and Jessica Hammer. The view from MARS: Empowering game stream viewers with metadata augmented real-time streaming. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, number Article 62 in UIST '23, pages 1–13, New York, NY, USA, October 2023. Association for Computing Machinery. 5.1, 5.2.2, 5.3.2, 5.3.3

[131] Noor Hammad, C. Ailie Fraser, Erik Harpstead, Jessica Hammer, and Mira Dontcheva. "It's more of a vibe I'm going for": Designing Text-to-Music Generation Interfaces for Video Creators. In *Proceedings of the 2025 ACM Designing Interactive Systems Conference*, DIS '25, pages 2738–2754, New York, NY, USA, July 2025. Association for Computing Machinery. ISBN 979-8-4007-1485-6. doi: 10.1145/3715336.3735814. 7.1, 7.4.2.4

[132] Hyunyoung Han, Kyungeun Jung, and Sang Ho Yoon. ChoreoCraft: In-situ Crafting of Choreography in Virtual Reality through Creativity Support Tool. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 979-8-4007-1394-1. doi: 10.1145/3706598.3714220. **??**, 7.4.1.2, 7.4.2.4

[133] John Harris and Mark Hancock. To asymmetry and beyond! improving social connectedness by increasing designed interdependence in cooperative play. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, number Paper 9 in CHI '19, pages 1–12, New York, NY, USA, May 2019. Association for Computing Machinery. 4.4.3

[134] John Harris, Mark Hancock, and Stacey D Scott. Leveraging asymmetries in multiplayer games: Investigating design elements of interdependent play. In *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play*, CHI PLAY '16, pages 350–361, New York, NY, USA, October 2016. Association for Computing Machinery. 4.4.3

[135] John Joseph Harris. *Leveraging asymmetry and interdependence to enhance social connectedness in cooperative digital games*. PhD thesis, May 2019. 4.4.3

[136] Gordon Hatcher, Will Ion, Ross Maclachlan, Mike Marlow, Bron Simpson, Nicky Wilson, and Alan Wodehouse. Using linkography to compare creative methods for group ideation. *Design Studies*, 61:179–198, 2018. doi: 10.1016/j.destud.2018.05.002. 2.3

[137] Lena Hegemann, Niraj Ramesh Dayama, Abhishek Iyer, Erfan Farhadi, Ekaterina Marchenko, and Antti Oulasvirta. CoColor: Interactive Exploration of Color Designs. In *Proceedings of the 28th International Conference on Intelligent User Interfaces*, IUI

’23, pages 106–127, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 979-8-4007-0106-1. doi: 10.1145/3581641.3584089. **??**, 7.4.2.2, 7.4.2.4

[138] Herzing. Media skills every content creator needs in 2025. https://blog.herzing.ca/media-skills-for-content-creators, 2025. URL https://blog.herzing.ca/media-skills-for-content-creators. Accessed: 2025-10-09. 1

[139] Zorah Hilvert-Bruce, James T. Neill, Max Sjöblom, and Juho Hamari. Social motivations of live-streaming viewer engagement on twitch. *Computers in Human Behavior*, 84:58–67, 2018. ISSN 0747-5632. doi: https://doi.org/10.1016/j.chb.2018.02.013. URL https://www.sciencedirect.com/science/article/pii/S0747563218300712. 4.2.2

[140] Josh Howarth. 30+ incredible creator economy statistics (2024), August 2022. URL https://explodingtopics.com/blog/creator-economy-stats. Accessed: 2025-1-19. 6.1

[141] Stacy Hsueh, Marianela Ciolfi, Sarah Fdili Alaoui, and Wendy E. Mackay. What counts as ‘creative’ work? articulating four epistemic positions in creativity-oriented hci research. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (CHI ’24)*. ACM, 2024. doi: 10.1145/3613904.3642854. 2.1, 7.1, 7.3.1.2

[142] Xi Hu, Yiwen Xing, Xudong Cai, Yihang Zhao, Michael Cook, Rita Borgo, and Timothy Neate. Designing interactions with generative ai for art and creativity: A systematic review and taxonomy. In *Proceedings of the ACM Designing Interactive Systems Conference (DIS ’25)*, page 30, Funchal, Portugal, 2025. ACM. doi: 10.1145/3715336.3735843. 7.2.1

[143] Forrest Huang, Eldon Schoop, David Ha, and John Canny. Scones: Towards conversational authoring of sketches. In *Proceedings of the 25th International Conference on Intelligent User Interfaces*, IUI ’20, pages 313–323, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 978-1-4503-7118-6. doi: 10.1145/3377325.3377485. **??**, 7.4.1.2, 7.4.2.3

[144] Qingqing Huang, Aren Jansen, Joonseok Lee, Ravi Ganti, Judith Yue Li, and Daniel P. W. Ellis. Mulan: A joint embedding of music audio and natural language, 2022. URL https://arxiv.org/abs/2208.12415. 6.2.2

[145] Zhanpeng Huang, Weikai Li, Pan Hui, and Christoph Peylo. Cloudridar: A cloud-based architecture for mobile augmented reality. In *Proceedings of the 2014 Workshop on Mobile Augmented Reality and Robotic Technology-Based Systems*, MARS ’14, page 29–34, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450328234. doi: 10.1145/2609829.2609832. URL https://doi.org/10.1145/2609829.2609832. 3.2.2

[146] Scott E. Hudson and Jennifer Mankoff. *Concepts, Values, and Methods for Technical Human–Computer Interaction Research*, pages 69–93. Springer New York, New York, NY, 2014. ISBN 978-1-4939-0378-8. doi: 10.1007/978-1-4939-0378-8_4. URL https://doi.org/10.1007/978-1-4939-0378-8_4. 3.5.1.3

[147] Linda Hughes. Rooie ryles. In Katie Salen and Eric Zimmerman, editors, *The Game*

Design Reader: A Rules of Play Anthology, pages 504–518. MIT Press, 2006. 3.5.1.1

[148] Mina Huh, Ding Li, Kim Pimmel, Hijung Valentina Shin, Amy Pavel, and Mira Dontcheva. VideoDiff: Human-AI Video Co-Creation with Alternatives. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 979-8-4007-1394-1. doi: 10.1145/3706598.3713417. 8.1

[149] Ian Hamilton. Game accessibility guidelines. https://gameaccessibilityguidelines.com/why-and-how/, 2021. Accessed: April 5, 2023. 4.4.2

[150] Naughty Dog Inc. The last of us part ii. PlayStation 4, 2020. 4.4.2

[151] Niantic Inc. Pokemon go: Battling at gyms. https://niantic.helpshift.com/a/pokemon-go/?s=gyms-battle&f=battling-at-gyms&l=en&p=web, 2017. Accessed: 2021-4-6. 3.2.1

[152] Niantic Inc. Niantic Planet-Scale AR alliance accelerates social AR future in codename: Urban legends - niantic. https://nianticlabs.com/blog/niantic-planet-scale-ar-5g-urban-legends/, 2021. Accessed: 2021-9-6. 3.2.2

[153] Alexander Ivanov, David Ledo, Tovi Grossman, George Fitzmaurice, and Fraser Anderson. MoodCubes: Immersive Spaces for Collecting, Discovering and Envisioning Inspiration Materials. In *Proceedings of the 2022 ACM Designing Interactive Systems Conference*, DIS '22, pages 189–203, New York, NY, USA, June 2022. Association for Computing Machinery. ISBN 978-1-4503-9358-4. doi: 10.1145/3532106.3533565. 7.4, 7.4.1.4

[154] Mikhail Jacob and Brian Magerko. Empirically Evaluating Creative Arc Negotiation for Improvisational Decision-making. In *Proceedings of the 13th Conference on Creativity and Cognition*, C&C '21, pages 1–10, New York, NY, USA, June 2021. Association for Computing Machinery. ISBN 978-1-4503-8376-9. doi: 10.1145/3450741.3465263. **??**, **??**, 7.4.1.4, 7.4.2.4

[155] Amir Jahanlou and Parmit K Chilana. Katika: An End-to-End System for Authoring Amateur Explainer Motion Graphics Videos. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 978-1-4503-9157-3. doi: 10.1145/3491102.3517741. **??**, 7.4.1.4, 7.4.2.4

[156] Mike Jia and Weifa Liang. Delay-sensitive multiplayer augmented reality game planning in mobile edge computing. In *Proceedings of the 21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, MSWIM '18, page 147–154, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450359603. doi: 10.1145/3242102.3242129. URL https://doi.org/10.1145/3242102.3242129. 3.2.2

[157] Mark R Johnson and Jamie Woodcock. The impacts of live streaming and twitch.tv on the video game industry. *Media Cult. Soc.*, 41(5):670–688, July 2019. 5.6.2

[158] Jeff W. T. Kan and John S. Gero. Acquiring information from linkography in protocol studies of designing. *Design Studies*, 29(4):315–337, 2008. doi: 10.1016/j.destud.2008.03.001. 2.3

[159] Jussi Kasurinen, Jukka-Pekka Strandén, and Kari Smolander. What do game developers expect from development and design tools? In *Proceedings of the 17th international conference on evaluation and assessment in software engineering*, pages 36–41, 2013. 1, 1.1

[160] Jasmine Kaur, Milka Trajkova, Andrea Knowlton, and Brian Magerko. Bringing LuminAI to Life: Studying Dancers' Perceptions of a Co-Creative AI in Dance Improvisation Class. In *Proceedings of the 2025 Conference on Creativity and Cognition*, C&C '25, pages 859–871, New York, NY, USA, June 2025. Association for Computing Machinery. ISBN 979-8-4007-1289-0. doi: 10.1145/3698061.3726925. 7.4.2.4

[161] Yubo Kou. Toxic behaviors in team-based competitive gaming: The case of league of legends. In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*, CHI PLAY '20, page 81–92, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450380744. doi: 10.1145/3410404.3414243. URL https://doi.org/10.1145/3410404.3414243. 3.5.2.2

[162] Max Kreminski and John Joon Young Chung. Intent elicitation in mixed-initiative co-creativity. In *IUI Workshops*, 2024. 2.3, 7.5.1

[163] Max Kreminski, Amy Smith, John Joon Young Chung, Kihoon Son, Qian Yang, Sang Won Lee, Noor Hammad, Eric Rawn, Shm Garanganao Almeda, and J.D. Zamfirescu-Pereira. Herding cats: Making sense of creative activity traces. In *Proceedings of the Extended Abstracts of the 2026 CHI Conference on Human Factors in Computing Systems*, CHI EA '26, New York, NY, USA, 2026. Association for Computing Machinery. ISBN 9798400722813. doi: 10.1145/3772363.3778743. URL https://doi.org/10.1145/3772363.3778743. 1.1, 2.3

[164] Google Labs. Musicfx, 2023. URL https://labs.google/fx/tools/music-fx. 6.2.3, 6.4.2, 6.5.1

[165] Tomas Lawton, Kazjon Grace, and Francisco J Ibarrola. When is a Tool a Tool? User Perceptions of System Agency in Human–AI Co-Creative Drawing. In *Proceedings of the 2023 ACM Designing Interactive Systems Conference*, DIS '23, pages 1978–1996, New York, NY, USA, July 2023. Association for Computing Machinery. ISBN 978-1-4503-9893-0. doi: 10.1145/3563657.3595977. 7.4.1.2

[166] Finchley Learning. Top 10 challenges faced by a videography editor and how to overcome them. https://www.finchley.co.uk/finchley-learning/top-10-challenges-faced-by-a-videography-editor-and-how-to-overcome-th 2025. URL https://www.finchley.co.uk/finchley-learning/top-10-challenges-faced-by-a-videography-editor-and-how-to-overcome-th Accessed: 2025-10-09. 1, 8.1, 8.2.1

[167] ChungHa Lee, DaeHo Lee, and Jin-Hyuk Hong. MVPrompt: Building Music-Visual Prompts for AI Artists to Craft Music Video Mise-en-scène. In *Proceedings of the 2025*

CHI Conference on Human Factors in Computing Systems, CHI '25, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 979-8-4007-1394-1. doi: 10.1145/3706598.3713876. 7.4.2.3

[168] Mina Lee, Katy Ilonka Gero, John Joon Young Chung, Simon Buckingham Shum, Vipul Raheja, Hua Shen, Subhashini Venugopalan, Thiemo Wambsganss, David Zhou, Emad A. Alghamdi, Tal August, Avinash Bhat, Madiha Zahrah Choksi, Senjuti Dutta, Jin L. C. Guo, Md Naimul Hoque, Yewon Kim, Simon Knight, Seyed Parsa Neshaei, Antonette Shibani, Disha Shrivastava, Lila Shroff, Agnia Sergeyuk, Jessi Stark, Sarah Sterman, Sitong Wang, Antoine Bosselut, Daniel Buschek, Joseph Chee Chang, Sherol Chen, Max Kreminski, Joonsuk Park, Roy Pea, Eugenia Ha Rim Rho, Zejiang Shen, and Pao Pao Siangliulue. A design space for intelligent and interactive writing assistants. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, page 1054, Honolulu, HI, USA, 2024. ACM. doi: 10.1145/3613904.3642697. 7.2.1, 7.3.1.4

[169] Seung Won Lee, Tae Hee Jo, Semin Jin, Jiin Choi, Kyungwon Yun, Sergio Bromberg, Seonghoon Ban, and Kyung Hoon Hyun. The impact of sketch-guided vs. prompt-guided 3d generative ais on the design exploration process. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pages 1–18, 2024. 2.3, 7.3.1, 7.5.2.1

[170] Elias Leight. Google trained its AI on copyrighted music, sources say — now it's trying to make deals, January 2024. URL http://www.billboard.com/pro/google-youtube-trained-ai-copyrighted-music-before-deals. Accessed: 2025-1-19. 6.7.1

[171] Pascal Lessel and Maximilian Altmeyer. Understanding and empowering interactions between streamer and audience in game live streams. *Interactions*, 27(1):40–45, December 2019. 4.1, 4.2.1.1, 4.2.2

[172] Pascal Lessel, Alexander Vielhauer, and Antonio Krüger. Expanding video game Live-Streams with enhanced communication channels: A case study. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 1571–1576, New York, NY, USA, May 2017. Association for Computing Machinery. 4.1, 4.2.1.2, 4.3.2

[173] Pascal Lessel, Alexander Vielhauer, and Antonio Krüger. Expanding video game Live-Streams with enhanced communication channels: A case study. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 1571–1576, New York, NY, USA, May 2017. Association for Computing Machinery. 5.1, 5.2.2, 5.4, 5.5, 5.5, 5.5

[174] Pascal Lessel, Alexander Vielhauer, and Antonio Krüger. CrowdChess: A system to investigate shared game control in Live-Streams. In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*, CHI PLAY '17, pages 389–400, New York, NY, USA, October 2017. Association for Computing Machinery. 5.2.1, 5.6.1

[175] Pascal Lessel, Maximilian Altmeyer, and Antonio Krüger. Viewers' perception of elements used in game Live-Streams. In *Proceedings of the 22nd International Academic Mindtrek Conference*, Mindtrek '18, pages 59–68, New York, NY, USA, October 2018.

Association for Computing Machinery. 4.1, 4.2.2, 5.2.2

[176] Jingyi Li, Eric Rawn, Jacob Ritchie, Jasper Tran O'Leary, and Sean Follmer. Beyond the artifact: power as a lens for creativity support tools. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 1–15, 2023. 1, 2.1, 7.1, 7.5.1, 7.6.1, 7.6.3

[177] Lingyuan Li, Guo Freeman, and Nathan J McNeese. Channeling End-User creativity: Leveraging live streaming for distributed collaboration in indie game development. *Proc. ACM Hum.-Comput. Interact.*, 6(CSCW2):1–28, November 2022. 5.2.2

[178] David Chuan-En Lin and Nikolas Martelaro. Learning personal style from few examples. In *Proceedings of the ACM Designing Interactive Systems Conference (DIS '21)*, pages 1566–1578, 2021. doi: 10.1145/3461778.3462115. URL https://doi.org/10.1145/3461778.3462115. 2.3, 8.2.1

[179] David Chuan-En Lin, Fabian Caba Heilbron, Joon-Young Lee, Oliver Wang, and Nikolas Martelaro. VideoMap: Supporting Video Exploration, Brainstorming, and Prototyping in the Latent Space. In *Proceedings of the 16th Conference on Creativity & Cognition*, C&C '24, pages 311–327, New York, NY, USA, June 2024. Association for Computing Machinery. ISBN 979-8-4007-0485-7. doi: 10.1145/3635636.3656192. 1.1

[180] Jen-Chun Lin, Wen-Li Wei, James Yang, Hsin-Min Wang, and Hong-Yuan Mark Liao. Automatic music video generation based on simultaneous soundtrack recommendation and video editing. In *Proceedings of the 25th ACM International Conference on Multimedia*, MM '17, page 519–527, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450349062. doi: 10.1145/3123266.3123399. URL https://doi.org/10.1145/3123266.3123399. 6.2.2

[181] Jianhui Liu and Qi Zhang. Edge computing enabled mobile augmented reality with imperfect channel knowledge. In *European Wireless 2019; 25th European Wireless Conference*, pages 1–6, 2019. 3.2.2

[182] Luyang Liu, Hongyu Li, and Marco Gruteser. Edge assisted real-time object detection for mobile augmented reality. In *The 25th Annual International Conference on Mobile Computing and Networking*, MobiCom '19, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450361699. doi: 10.1145/3300061.3300116. URL https://doi.org/10.1145/3300061.3300116. 3.2.2

[183] Qiang Liu and Tao Han. Dare: Dynamic adaptive mobile augmented reality with edge computing. In *2018 IEEE 26th International Conference on Network Protocols (ICNP)*, pages 1–11, 2018. doi: 10.1109/ICNP.2018.00011. 3.1

[184] Vivian Liu and Lydia B Chilton. Design guidelines for prompt engineering text-to-image generative models. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450391573. doi: 10.1145/3491102.3501825. URL https://doi.org/10.1145/3491102.3501825. 6.2.1

[185] Vivian Liu, Rubaiat Habib Kazi, Li-Yi Wei, Matthew Fisher, Timothy Langlois, Seth Walker, and Lydia Chilton. LogoMotion: Visually-Grounded Code Synthesis for Creating

and Editing Animation. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 979-8-4007-1394-1. doi: 10.1145/3706598.3714155. 7.4.1.1

[186] Wenliang Liu, Jinke Ren, Guan Huang, Yinghui He, and Guanding Yu. Data offloading and sharing for latency minimization in augmented reality based on mobile-edge computing. In *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, pages 1–5, 2018. doi: 10.1109/VTCFall.2018.8690922. 3.2.2

[187] Ryan Louie, Andy Coenen, Cheng Zhi Huang, Michael Terry, and Carrie J. Cai. Novice-ai music co-creation via ai-steering tools for deep generative models. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–13, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450367080. doi: 10.1145/3313831.3376739. URL https://doi.org/10.1145/3313831.3376739. 6.2.1, 6.2.3

[188] Ryan Louie, Jesse Engel, and Cheng-Zhi Anna Huang. Expressive communication: Evaluating developments in generative models and steering interfaces for music creation. In *Proceedings of the 27th International Conference on Intelligent User Interfaces*, IUI '22, page 405–417, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450391443. doi: 10.1145/3490099.3511159. URL https://doi.org/10.1145/3490099.3511159. 6.2.3

[189] Zhicong Lu, Seongkook Heo, and Daniel J Wigdor. StreamWiki: Enabling viewers of knowledge sharing live streams to collaboratively generate archival documentation for effective In-Stream and post hoc learning. *Proc. ACM Hum.-Comput. Interact.*, 2(CSCW): 1–26, November 2018. 4.2.1.2, 5.2.2

[190] Zhicong Lu, Michelle Annett, Mingming Fan, and Daniel Wigdor. " i feel it is my responsibility to stream" streaming and engaging with intangible cultural heritage through livestreaming. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2019. 1, 4.5.3, 5.2.1, 5.2.2

[191] Zhicong Lu, Rubaiat Habib Kazi, Li-Yi Wei, Mira Dontcheva, and Karrie Karahalios. StreamSketch: Exploring Multi-Modal interactions in creative live streams. *Proc. ACM Hum.-Comput. Interact.*, 5(CSCW1):1–26, April 2021. 4.2.1.2

[192] Chance Lytle, Parker Ramsey, Joey Yeo, Trace Dressen, Dong Hyun Kang, Brenda Bakker Harger, and Jessica Hammer. Toward live streamed improvisational game experiences. In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*, CHI PLAY '20, pages 148–159, New York, NY, USA, November 2020. Association for Computing Machinery. 4.2.2, 5.2.1

[193] Atefeh Mahdavi Goloujeh, Anne Sullivan, and Brian Magerko. Is it ai or is it me? understanding users' prompt journey with text-to-image generative ai tools. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2024. 2.3

[194] Moshe Mahler, Yuchen Liu, and Jodi Forlizzi. The evolution of artists' toolsets: A call for artist-driven design. In *Companion Publication of the 2025 ACM Designing Interactive*

Systems Conference, pages 335–339, 2025. 1, 1.1, 2.1, 7.1, 7.6.3

[195] Keri Mallari, Spencer Williams, and Gary Hsieh. Understanding analytics needs of video game streamers. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, number Article 337 in CHI '21, pages 1–12, New York, NY, USA, May 2021. Association for Computing Machinery. 1, 5.2.2, 5.6.1

[196] Ilaria Manco, Emmanouil Benetos, Elio Quinton, and György Fazekas. Contrastive audio-language learning for music, 2022. URL https://arxiv.org/abs/2208.12208. 6.2.2

[197] Daniel Manesh, Douglas Bowman Jr., and Sang Won Lee. SHARP: Exploring Version Control Systems in Live Coding Music. In *Proceedings of the 16th Conference on Creativity & Cognition*, C&C '24, pages 426–437, New York, NY, USA, June 2024. Association for Computing Machinery. ISBN 979-8-4007-0485-7. doi: 10.1145/3635636.3656195. 7.4.1.4, 7.4.2.4

[198] Viral Idea Marketing. How to create consistent branding across multiple videos through editing. https://www.viralideamarketing.com/post/how-to-create-consistent-branding-across-multiple-videos-through-editi 2025. URL https://www.viralideamarketing.com/post/how-to-create-consistent-branding-across-multiple-videos-through-editi Accessed: 2025-10-09. 8.1, 8.2.1

[199] Masahiro Matsuo, Fumi Masuda, Yukiyoshi Sumi, Masahiro Takahashi, Atsushi Yoshimura, Naoto Yamada, and Hiroshi Kadotani. Background music dependent reduction of aversive perception and its relation to P3 amplitude reduction and increased heart rate. *Front. Hum. Neurosci.*, 13:184, June 2019. URL https://pubmed.ncbi.nlm.nih.gov/31316359/. 6.1

[200] Daniel McKee, Justin Salamon, Josef Sivic, and Bryan Russell. Language-guided music recommendation for video via prompt analogies. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14784–14793, 2023. doi: 10.1109/CVPR52729.2023.01420. 6.2.2

[201] Matt Miesnieks. Multiplayer AR — why it's quite hard. https://medium.com/6d-ai/multiplayer-ar-why-its-quite-hard-43efdb378418, January 2018. Accessed: 2021-9-6. 3.1

[202] Matthew K Miller, John C Tang, Gina Venolia, Gerard Wilkinson, and Kori Inkpen. Conversational chat circles: Being all here without having to hear it all. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 2394–2404, New York, NY, USA, May 2017. Association for Computing Machinery. 4.2.1.1, 5.2.1, 5.2.2

[203] Piotr Mirowski, Kory Mathewson, Boyd Branch, Thomas Winters, Ben Verhoeven, and Jenny Elfving. Rosetta code: Improv in any language. In Tony Veale F. Amílcar Cardoso, Penousal Machado and João Miguel Cunha, editors, *Proceedings of the 11th International Conference on Computational Creativity*, pages 115–122, Coimbra, Portugal, 2020. Association for Computational Creativity. ISBN 978-989-54160-2-8. 7.4.2.2, 7.4.2.4

[204] Aditi Mishra, Frederik Brudy, Qian Zhou, George Fitzmaurice, and Fraser Anderson. WhatIF: Branched Narrative Fiction Visualization for Authoring Emergent Narratives using Large Language Models. In *Proceedings of the 2025 Conference on Creativity and Cognition*, C&C '25, pages 590–605, New York, NY, USA, June 2025. Association for Computing Machinery. ISBN 979-8-4007-1289-0. doi: 10.1145/3698061.3726933. **??**, 7.4.2.4

[205] Florian Mueller and Katherine Isbister. Movement-based game guidelines. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, page 2191–2200, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450324731. doi: 10.1145/2556288.2557163. URL https://doi.org/10.1145/2556288.2557163. 3.2.1

[206] Aaditya Naik, Jonathan Mendelson, Nathaniel Sands, Yuepeng Wang, Mayur Naik, and Mukund Raghothaman. Sporq: An interactive environment for exploring code using query-by-example. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, UIST '21, page 84–99, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450386357. doi: 10.1145/3472749.3474737. URL https://doi.org/10.1145/3472749.3474737. 7.3.1.2

[207] Niantic. *Pokémon Go*. Game [Mobile], July 2016. Nintendo. 3.2.1

[208] Zachary Novack, Julian McAuley, Taylor Berg-Kirkpatrick, and Nicholas Bryan. Ditto-2: Distilled diffusion inference-time t-optimization for music generation, 2024. URL https://arxiv.org/abs/2405.20289. 6.2.3, 6.7

[209] Zachary Novack, Julian McAuley, Taylor Berg-Kirkpatrick, and Nicholas J. Bryan. Ditto: Diffusion inference-time t-optimization for music generation, 2024. URL https://arxiv.org/abs/2401.12179. 6.2.3, 6.7

[210] Zachary Novack, Ge Zhu, Jonah Casebeer, Julian McAuley, Taylor Berg-Kirkpatrick, and Nicholas J. Bryan. Presto! distilling steps and layers for accelerating music generation, 2024. URL https://arxiv.org/abs/2410.05167. 6.1, 6.2.3

[211] William Odom and Tijs Duel. On the design of olo radio: Investigating metadata as a design material. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–9, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450356206. doi: 10.1145/3173574.3173678. URL https://doi.org/10.1145/3173574.3173678. 2.2, 2.3, 7.1

[212] Jeongseok Oh, Seungju Kim, and Seungjun Kim. LumiMood: A Creativity Support Tool for Designing the Mood of a 3D Scene. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 979-8-4007-0330-0. doi: 10.1145/3613904.3642440. **??**, 7.4.2.4

[213] OpenAI. Clip: Connecting text and images, 2021. URL https://openai.com/index/clip/. Accessed: 2025-1-19. 6.4.3.6

[214] Janne Paavilainen, Hannu Korhonen, Kati Alha, Jaakko Stenros, Elina Koskinen, and Frans Mayra. The pokémon go experience: A location-based augmented reality mobile

game goes mainstream. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, page 2493–2498, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450346559. doi: 10.1145/3025453.3025871. URL https://doi.org/10.1145/3025453.3025871. 3.2.1

[215] Pedro Pais, David Gonçalves, Daniel Reis, João Cadete Nunes Godinho, João Filipe Morais, Manuel Piçarra, Pedro Trindade, Dmitry Alexandrovsky, Kathrin Gerling, João Guerreiro, and André Rodrigues. A living framework for understanding cooperative games. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400703300. doi: 10.1145/3613904.3641953. URL https://doi.org/10.1145/3613904.3641953. 7.4, 7.6.1

[216] Srishti Palani, David Ledo, George Fitzmaurice, and Fraser Anderson. "i don't want to feel like i'm working in a 1960s factory": The practitioner perspective on creativity support tool adoption. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450391573. doi: 10.1145/3491102.3501933. URL https://doi.org/10.1145/3491102.3501933. 1.1, 7.1, 7.5.3.4

[217] Rui Pan, Lyn Bartram, and Carman Neustaedter. TwitchViz: A visualization tool for twitch chatrooms. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '16, pages 1959–1965, New York, NY, USA, May 2016. Association for Computing Machinery. 4.2.1.1, 5.2.1, 5.2.2

[218] Rock Yuren Pang, Hope Schroeder, Kynnedy Simone Smith, Solon Barocas, Ziang Xiao, Emily Tseng, and Danielle Bragg. Understanding the llm-ification of chi: Unpacking the impact of llms at chi through a systematic literature review. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–20, 2025. 2.1, 7.1, 7.2.1, 7.4

[219] Jeongeun Park, Hyorim Shin, Changhoon Oh, and Ha Young Kim. "is text-based music search enough to satisfy your needs?" a new way to discover music with images. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400703300. doi: 10.1145/3613904.3642126. URL https://doi.org/10.1145/3613904.3642126. 6.1, 6.2.2, 6.4.1

[220] Minjung Park, Jodi Forlizzi, and John Zimmerman. Exploring the innovation opportunities for pre-trained models. In *Proceedings of the 2025 ACM Designing Interactive Systems Conference*, pages 1973–2005, 2025. 2.1, 7.1, 7.2.1, 7.3.1.4

[221] Mengqi Peng, Li-yi Wei, Rubaiat Habib Kazi, and Vladimir G. Kim. Autocomplete Animated Sculpting. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, UIST '20, pages 760–777, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 978-1-4503-7514-6. doi: 10.1145/3379337.3415884. 7.4.2.4

[222] Kara Pernice. Affinity diagramming for collaboratively sorting UX findings and design

ideas. https://www.nngroup.com/articles/affinity-diagram/, February 2018. Accessed: 2024-2-21. 5.3, 5.3.2

[223] Karine Pires and Gwendal Simon. Youtube live and twitch: a tour of user-generated live streaming systems. In *Proceedings of the 6th ACM multimedia systems conference*, pages 225–230, 2015. 1, 4.1, 4.2

[224] Jonathan A Plucker, Ronald A Beghetto, and Gayle T Dow. Why isn't creativity more important to educational psychologists? potentials, pitfalls, and future directions in creativity research. *Educational psychologist*, 39(2):83–96, 2004. 1.1, 2.2, 7.1

[225] Laure Pretet, Gael Richard, and Geoffroy Peeters. Cross-modal music-video recommendation: A study of design choices, 2021. URL https://arxiv.org/abs/2104.14799. 6.2.2

[226] Laure Prétet, Gaël Richard, Clément Souchier, and Geoffroy Peeters. Video-to-music recommendation using temporal alignment of segments. *IEEE Transactions on Multimedia*, 25:2898–2911, 2023. doi: 10.1109/TMM.2022.3152598. 6.2.2

[227] Dennis Ramirez, Jenny Saucerman, and Jeremy Dietmeier. Twitch plays pokemon: a case study in big g games. In *Proceedings of DiGRA*, pages 3–6, 2014. 4.2.1.1, 5.2.1

[228] Xukan Ran, Carter Slocum, Maria Gorlatova, and Jiasi Chen. Sharear: Communication-efficient multi-user mobile augmented reality. In *Proceedings of the 18th ACM Workshop on Hot Topics in Networks*, HotNets '19, page 109–116, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450370202. doi: 10.1145/3365609.3365867. URL https://doi.org/10.1145/3365609.3365867. 3.2.2

[229] Eric Rawn, Jingyi Li, Eric Paulos, and Sarah E. Chasins. Understanding Version Control as Material Interaction with Quickpose. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 978-1-4503-9421-5. doi: 10.1145/3544548.3581394. 2.3, 7.1, **??**, 7.4.1.4, 7.4.2.1, 7.4.2.3, 7.4.2.4

[230] RealityEscape. What makes tower defence games good?, April 2019. 4.4.1

[231] Redis Technologies. Introduction to redis. https://redis.io/docs/about/. Accessed: 2023-4-5. 4.3.4

[232] Christian Remy, Lindsay MacDonald Vermeulen, Jonas Frich, Michael Mose Biskjær, and Peter Dalsgaard. Evaluating creativity support tools in hci research. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference (DIS '20)*, pages 457–476, Eindhoven, Netherlands, 2020. ACM. doi: 10.1145/3357236.3395474. 7.2.1

[233] Jinke Ren, Yinghui He, Guan Huang, Guanding Yu, Yunlong Cai, and Zhaoyang Zhang. An edge-computing based architecture for mobile augmented reality. *IEEE Network*, 33 (4):162–169, 2019. doi: 10.1109/MNET.2018.1800132. 3.2.2

[234] Microsoft Research. LLaVA: Large language and vision assistant, October 2023. URL https://www.microsoft.com/en-us/research/project/llava-large-language-and-vision-assistant/. Accessed: 2025-1-19. 6.4.3.6

[235] Mel Rhodes. An analysis of creativity. *The Phi delta kappan*, 42(7):305–310, 1961. 2.2

[236] Frans Rijnders, Günter Wallner, and Regina Bernhaupt. Live feedback for training through Real-Time data visualizations: A study with league of legends. *Proc. ACM Hum.-Comput. Interact.*, 6(CHI PLAY):1–23, October 2022. 5.2

[237] Raquel Robinson, Zachary Rubin, Elena Márquez Segura, and Katherine Isbister. All the feels: designing a tool that reveals streamers' biometrics to spectators. In *Proceedings of the 12th International Conference on the Foundations of Digital Games*, number Article 36 in FDG '17, pages 1–6, New York, NY, USA, August 2017. Association for Computing Machinery. 4.2.2, 5.2.1

[238] Erica Robles and Mikael Wiberg. Texturing the "material turn" in interaction design. In *Proceedings of the Fourth International Conference on Tangible, Embedded, and Embodied Interaction*, TEI '10, page 137–144, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781605588414. doi: 10.1145/1709886.1709911. URL https://doi.org/10.1145/1709886.1709911. 8.1, 8.2.2, 8.5.2.3, 8.6

[239] Stephen Robles. How much do YouTubers make? (earnings and examples), September 2024. URL https://riverside.fm/blog/how-much-do-youtubers-make. Accessed: 2025-1-19. 6.1

[240] Ana Rodrigues, Diogo Cabral, and Pedro Campos. Creativity support tools and convergent thinking: A preliminary review on idea evaluation and selection. In *Proceedings of the 15th Conference on Creativity & Cognition (C&C '23)*, pages 305–311, Virtual Event, USA, 2023. ACM. doi: 10.1145/3591196.3596821. 7.2.1

[241] Melissa Roemmele. Inspiration through observation: Demonstrating the influence of automatically generated text on creative writing. *arXiv preprint arXiv:2107.04007*, 2021. URL https://arxiv.org/abs/2107.04007. 2.1

[242] Vishnu Sarukkai, Lu Yuan, Mia Tang, Maneesh Agrawala, and Kayvon Fatahalian. Block and Detail: Scaffolding Sketch-to-Image Generation. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, UIST '24, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 979-8-4007-0628-8. doi: 10.1145/3654777.3676444. 7.4.1.2

[243] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies. The case for vm-based cloudlets in mobile computing. *IEEE Pervasive Computing*, 8(4):14–23, 2009. doi: 10.1109/MPRV.2009.82. 3.1, 3.2.2

[244] Mahadev Satyanarayanan. The Emergence of Edge Computing. *IEEE Computer*, 50(1), January 2017. 3.1

[245] Mahadev Satyanarayanan, Wei Gao, and Brandon Lucia. The Computing Landscape of the 21st Century. In *Proceedings of the 20th International Workshop on Mobile Computing Systems and Applications (HotMobile '19)*, Santa Cruz, CA, 2019. 3.2.2

[246] Oliver Schmitt and Daniel Buschek. CharacterChat: Supporting the Creation of Fictional Characters through Conversation and Progressive Manifestation with a Chatbot. In *Proceedings of the 13th Conference on Creativity and Cognition*, C&C '21, pages 1–10,

New York, NY, USA, June 2021. Association for Computing Machinery. ISBN 978-1-4503-8376-9. doi: 10.1145/3450741.3465253. 7.4.2.1

[247] R. Schmoll, S. Pandi, P. J. Braun, and F. H. P. Fitzek. Demonstration of vr / ar offloading to mobile edge cloud for low latency 5g gaming application. In *2018 15th IEEE Annual Consumer Communications Networking Conference (CCNC)*, pages 1–3, 2018. doi: 10.1109/CCNC.2018.8319323. 3.2.2

[248] Flavio Schneider, Ojasv Kamal, Zhijing Jin, and Bernhard Schölkopf. Moûsai: Text-to-music generation with long-context latent diffusion, 2023. URL https://arxiv.org/abs/2301.11757. 6.1, 6.2.3

[249] Donald A Schon and Vincent DeSanctis. The reflective practitioner: How professionals think in action, 1986. 1.1, 7.1, 8.2.2

[250] Patrick Schuck, Maximilian Altmeyer, Antonio Krüger, and Pascal Lessel. Viewer types in game live streams: questionnaire development and validation. *User Model. User-adapt Interact.*, 32(3):417–467, July 2022. 4.2.2

[251] Joseph Seering and Sanjay R Kairam. Who moderates on twitch and what do they do? quantifying practices in community moderation on twitch. *Proc. ACM Hum.-Comput. Interact.*, 7(GROUP):1–18, December 2022. 5.2.2

[252] Joseph Seering, Saiph Savage, Michael Eagle, Joshua Churchin, Rachel Moeller, Jeffrey P Bigham, and Jessica Hammer. Audience participation games: Blurring the line between player and spectator. In *Proceedings of the 2017 Conference on Designing Interactive Systems*, pages 429–440, 2017. 4.1, 4.2.2, 5.1, 5.2.1, 5.3.3, 5.4, 5.4.2, 5.4.3.1, 5.6.1

[253] Elena Márquez Segura, Katherine Isbister, Jon Back, and Annika Waern. Design, appropriation, and use of technology in larps. In *Proceedings of the 12th International Conference on the Foundations of Digital Games*, FDG '17, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450353199. doi: 10.1145/3102071.3106360. URL https://doi.org/10.1145/3102071.3106360. 3.5.1.3

[254] Ryan Shea, Andy Sun, Silvery Fu, and Jiangchuan Liu. Towards fully offloaded cloud-based ar: Design, implementation and experience. In *Proceedings of the 8th ACM on Multimedia Systems Conference*, MMSys'17, page 321–330, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450350020. doi: 10.1145/3083187.3084012. URL https://doi.org/10.1145/3083187.3084012. 3.2.2

[255] Ben Shneiderman. Creativity support tools: accelerating discovery and innovation. *Communications of the ACM*, 50(12):20–32, 2007. 2.1, 7.1, 7.5.2.1

[256] Ben Shneiderman. Creativity support tools: A grand challenge for hci researchers. In *Engineering the user interface: From research to practice*, pages 1–9. Springer, 2008. 2.1

[257] Momin N Siddiqui, Roy D Pea, and Hari Subramonyam. Script&Shift: A Layered Interface Paradigm for Integrating Content Development and Rhetorical Strategy with LLM Writing Assistants. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 979-8-4007-1394-1. doi: 10.1145/3706598.3714119. 7.1, 7.4.2.3

[258] Max Sjöblom and Juho Hamari. Why do people watch others play video games? an empirical study on the motivations of twitch users. *Computers in human behavior*, 75: 985–996, 2017. 4.2.2

[259] Amy Smith, Barrett R Anderson, Jasmine Tan Otto, Isaac Karth, Yuqian Sun, John Joon Young Chung, Melissa Roemmele, and Max Kreminski. Fuzzy linkography: Automatic graphical summarization of creative activity traces. In *Proceedings of the 2025 Conference on Creativity and Cognition*, pages 637–650, 2025. 1.1, 2.2, 2.3, 7.1, 9.2.1

[260] Olli Sotamaa. When the game is not enough: Motivations and practices among computer game modding culture. *Games and Culture*, 5(3):239–255, 2010. 4.5.2

[261] Epidemic Sound. Epidemic sound, 2009. URL https://www.epidemicsound.com. Accessed: 2025-1-19. 6.1, 6.2.2, 6.4.3.3

[262] Speechmatics. Speechmatics AI speech technology, 2023. URL https://www.speechmatics.com/. Accessed: 2025-1-19. 6.4.3.6

[263] Chris Speed and Jon Oberlander. Designing from, with and by data: Introducing the ablative framework. In Peter Lloyd and Edyta Bohemia, editors, *Future Focused Thinking, DRS International Conference 2016*, Brighton, United Kingdom, 2016. Design Research Society. doi: 10.21606/drs.2016.433. 2.3

[264] Samantha Stahlke, James Robb, and Pejman Mirza-Babaei. The fall of the fourth wall: Designing and evaluating interactive spectator experiences. *International Journal of Gaming and Computer-Mediated Simulations (IJGCMS)*, 10(1):42–62, 2018. 4.2.1.2, 5.2.1

[265] Evgeny Stemasov, David Ledo, George Fitzmaurice, and Fraser Anderson. Immersive Sampling: Exploring Sampling for Future Creative Practices in Media-Rich, Immersive Spaces. In *Proceedings of the 2023 ACM Designing Interactive Systems Conference*, DIS '23, pages 212–229, New York, NY, USA, July 2023. Association for Computing Machinery. ISBN 978-1-4503-9893-0. doi: 10.1145/3563657.3596131. 7.4, **??**, **??**, 7.4.1.4, 7.4.2.4

[266] Constantine Stephanidis, Gavriel Salvendy, Margherita Antona, Jessie Y. C. Chen, Jianming Dong, Vincent G. Duffy, Xiaowen Fang, Cali Fidopiastis, Gino Fragomeni, Limin Paul Fu, Yinni Guo, Don Harris, Andri Ioannou, Kyeong ah Jeong, Shin'ichi Konomi, Heidi Kroemker, Masaaki Kurosu, James R. Lewis, Aaron Marcus, Gabriele Meiselwitz, Abbas Moallem, Hirohiko Mori, Fiona Fui-Hoon Nah, Stavroula Ntoa, Pei-Luen Patrick Rau, Dylan Schmorrow, Keng Siau, Norbert Streitz, Wentao Wang, Sakae Yamamoto, Panayiotis Zaphiris, and Jia Zhou. Seven hci grand challenges. *International Journal of Human-Computer Interaction*, 35(14):1229–1269, 2019. doi: 10.1080/10447318.2019.1619259. 2.1

[267] Sarah Sterman, Molly Jane Nicholas, and Eric Paulos. Towards creative version control. *Proc. ACM Hum.-Comput. Interact.*, 6(CSCW2), November 2022. doi: 10.1145/3555756. URL https://doi.org/10.1145/3555756. 1.1, 2.1, 2.2, 2.3, 7.1, 7.4.2.3, 7.5.1, 7.5.2.2, 7.6.1, 8.6

[268] Sarah Gimbert Sterman. *Process-Sensitive Creativity Support Tools*. PhD thesis, University of California, Berkeley, 2022. 7.5.2.2

[269] Adobe Stock. Adobe royalty free stock music and audio, 2020. URL https://stock.adobe.com/audio. Accessed: 2025-1-19. 6.4.3.3

[270] StoryLab. Identity building challenges: Defining your youtube brand. https://storylab.ai/identity-building-challenges-defining-youtube-brand/, 2025. URL https://storylab.ai/identity-building-challenges-defining-youtube-b Accessed: 2025-10-09. 8.1, 8.2.1

[271] Alina Striner, Andrew M Webb, Jessica Hammer, and Amy Cook. Mapping design spaces for audience participation in game live streaming. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, number Article 329 in CHI '21, pages 1–15, New York, NY, USA, May 2021. Association for Computing Machinery. 5.1, 5.2.2, 5.4, 5.4.3.1

[272] Bethesta Game Studios. *The Elder Scrolls V: Skyrim*. Game [Multiple Platforms], November 2011. Bethesda. Last played September 2023. 5.4

[273] Kun Su, Judith Yue Li, Qingqing Huang, Dima Kuzmin, Joonseok Lee, Chris Donahue, Fei Sha, Aren Jansen, Yu Wang, Mauro Verzetti, and Timo I. Denk. V2meow: Meowing to the visual beat via video-to-music generation, 2024. URL https://arxiv.org/abs/2305.06594. 6.2.4, 6.6.1

[274] Xiang Su, Jacky Cao, and Pan Hui. 5g edge enhanced mobile augmented reality. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, MobiCom '20, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450370851. doi: 10.1145/3372224.3417315. URL https://doi.org/10.1145/3372224.3417315. 3.2.2

[275] Zhida Sun, Zhenyao Zhang, Yue Zhang, Min Lu, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Creative Blends of Visual Concepts. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 979-8-4007-1394-1. doi: 10.1145/3706598.3713683. 7.4.2.4

[276] Petra Sundström and Kristina Höök. Hand in hand with the material: Designing for suppleness. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 463–472, New York, NY, USA, April 2010. Association for Computing Machinery. ISBN 978-1-60558-929-9. doi: 10.1145/1753326.1753396. 8.2.2

[277] Petra Sundström, Alex Taylor, Katja Grufberg, Niklas Wirström, Jordi Solsona Belenguer, and Marcus Lundén. Inspirational bits: towards a shared understanding of the digital material. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, page 1561–1570, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 9781450302289. doi: 10.1145/1978942.1979170. URL https://doi.org/10.1145/1978942.1979170. 1.2, 8.1, 8.2.2, 8.5

[278] Suno. Suno, 2023. URL https://suno.com. 6.2.3, 6.4.2, 6.5.1

[279] Dídac Surís, Carl Vondrick, Bryan Russell, and Justin Salamon. It's time for artistic correspondence in music and video. In *2022 IEEE/CVF Conference on Computer Vision*

and Pattern Recognition (CVPR), pages 10554–10564, 2022. doi: 10.1109/CVPR52688.2022.01031. 6.1, 6.2.2

[280] Streamer Tactics. List of games with twitch integration - streamer tactics. https://streamertactics.com/games-with-twitch-integration/. Accessed: 2023-4-5. 4.2.1.2

[281] The Oh Lab. Bloomwood stories. http://theohlab.org/bloomwood-stories, 2021. Accessed: April 5, 2023. 4.4.2

[282] Bekzat Tilekbay, Saelyne Yang, Michal Adam Lewkowicz, Alex Suryapranata, and Juho Kim. Expressedit: Video editing with natural language and sketching. In *Proceedings of the 29th International Conference on Intelligent User Interfaces*, IUI '24, page 515–536. ACM, March 2024. doi: 10.1145/3640543.3645164. URL http://dx.doi.org/10.1145/3640543.3645164. 8.1

[283] Bekzat Tilekbay, Saelyne Yang, Michal Adam Lewkowicz, Alex Suryapranata, and Juho Kim. ExpressEdit: Video Editing with Natural Language and Sketching. In *Proceedings of the 29th International Conference on Intelligent User Interfaces*, IUI '24, pages 515–536, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 979-8-4007-0508-3. doi: 10.1145/3640543.3645164. 7.4.1.1, **??**, 7.4.1.3, 7.4.1.4

[284] Klervie Toczé, Johan Lindqvist, and Simin Nadjm-Tehrani. Performance study of mixed reality for edge computing. In *Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing*, UCC'19, page 285–294, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450368940. doi: 10.1145/3344341.3368816. URL https://doi.org/10.1145/3344341.3368816. 3.2.2

[285] Zachary O Toups, Jessica Hammer, William A Hamilton, Ahmad Jarrah, William Graves, and Oliver Garretson. A framework for cooperative communication game mechanics from grounded theory. In *Proceedings of the first ACM SIGCHI annual symposium on Computer-human interaction in play*, CHI PLAY '14, pages 257–266, New York, NY, USA, October 2014. Association for Computing Machinery. 4.4

[286] TubeBuddy. Personal branding on youtube: Building a strong brand. https://www.tubebuddy.com/blog/personal-branding-on-youtube-building-a-strong-brand/, 2025. URL https://www.tubebuddy.com/blog/personal-branding-on-youtube-building-a-strong-brand/. Accessed: 2025-10-09. 8.1, 8.2.1

[287] TubeBuddy. How youtube creators can overcome challenges. https://www.tubebuddy.com/blog/how-youtube-creators-can-overcome-challenges/, 2025. URL https://www.tubebuddy.com/blog/how-youtube-creators-can-overcome-challenges/. Accessed: 2025-10-09. 8.1

[288] Twitch Tv. Twitch.tv - extensions. https://www.twitch.tv/p/en/

extensions/. Accessed: 2023-4-5. 4.3.5

[289] Udio. Udio, 2024. URL https://www.udio.com/. 6.2.3, 6.4.2, 6.5.1

[290] Unity Technologies. Tower defense template. https://assetstore.unity.com/packages/essentials/tutorial-projects/tower-defense-template-107692, 2018. Accessed: April 5, 2023. 4.4.1

[291] Unity Technologies. Game objects. https://docs.unity3d.com/Manual/GameObjects.html, 2021. Accessed: April 5, 2023. 4.3.3

[292] Unity Technologies. Occlusion culling. https://docs.unity3d.com/Manual/OcclusionCulling.html, 2021. Accessed: April 5, 2023. 4.6

[293] Anna Vallgårda. Giving form to computational things: developing a practice of interaction design. *Personal Ubiquitous Comput.*, 18(3):577–592, March 2014. ISSN 1617-4909. doi: 10.1007/s00779-013-0685-8. URL https://doi.org/10.1007/s00779-013-0685-8. 8.2.2

[294] Anna Vallgårda and Johan Redström. Computational composites. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 513–522, New York, NY, USA, April 2007. Association for Computing Machinery. ISBN 978-1-59593-593-9. doi: 10.1145/1240624.1240706. 8.2.2, 8.5.1.4, 8.6

[295] Anna Vallgårda and Tomas Sokoler. A material focus: Exploring properties of computational composites. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '09, pages 4147–4152, New York, NY, USA, April 2009. Association for Computing Machinery. ISBN 978-1-60558-247-4. doi: 10.1145/1520340.1520631. 8.2.2, 8.5.1.4, 8.6

[296] Valve. *Portal 2*. Game [Multiple Platforms], April 2011. 5.4

[297] Olga Vechtomova, Gaurav Sahu, and Dhruv Kumar. LyricJam: A system for generating lyrics for live instrumental music. In Andrés Gómez de Silva Garza, Tony Veale, Wendy Aguilar, and Rafael Pérez y Pérez, editors, *Proceedings of the 12th International Conference on Computational Creativity*, pages 122–130, México City, México (Virtual), 2021. Association for Computational Creativity. ISBN 978-989-54160-3-5. **??**

[298] Veera Vimpari, Annakaisa Kultima, Perttu Hämäläinen, and Christian Guckelsberger. "an adapt-or-die type of situation": Perception, adoption, and use of text-to-image-generation ai by game industry professionals. *Proc. ACM Hum.-Comput. Interact.*, 7(CHI PLAY), October 2023. doi: 10.1145/3611025. URL https://doi.org/10.1145/3611025. 6.7.1

[299] Günter Wallner and Michael Lankes. The ludic potentials of player data. In *Companion Proceedings of the Annual Symposium on Computer-Human Interaction in Play*, CHI PLAY Companion '23, pages 209–211, New York, NY, USA, October 2023. Association for Computing Machinery. 5.2

[300] Günter Wallner, Letian Wang, and Claire Dormann. Visualizing the Spatio-Temporal evolution of gameplay using storyline visualization: A study with league of legends. *Proc. ACM Hum.-Comput. Interact.*, 7(CHI PLAY):1002–1024, October 2023. 5.2

[301] Qian Wan and Zhicong Lu. GANCollage: A GAN-Driven Digital Mood Board to Facilitate Ideation in Creativity Support. In *Proceedings of the 2023 ACM Designing Interactive Systems Conference*, DIS '23, pages 136–146, New York, NY, USA, July 2023. Association for Computing Machinery. ISBN 978-1-4503-9893-0. doi: 10.1145/3563657.3596072. 7.4.2.4

[302] Bryan Wang, Yuliang Li, Zhaoyang Lv, Haijun Xia, Yan Xu, and Raj Sodhi. Lave: Llm-powered agent assistance and language augmentation for video editing. In *Proceedings of the 29th International Conference on Intelligent User Interfaces (IUI '24)*, pages –, Greenville, SC, USA, 2024. ACM. doi: 10.1145/3640543.3645143. 7.4.1.3, 7.5.1

[303] Bryan Wang, Yuliang Li, Zhaoyang Lv, Haijun Xia, Yan Xu, and Raj Sodhi. Lave: Llm-powered agent assistance and language augmentation for video editing. In *Proceedings of the 29th International Conference on Intelligent User Interfaces*, IUI '24, page 699–714, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400705083. doi: 10.1145/3640543.3645143. URL https://doi.org/10.1145/3640543.3645143. 6.2.1, 6.5.3, 8.1

[304] Bryan Wang, Yuliang Li, Zhaoyang Lv, Haijun Xia, Yan Xu, and Raj Sodhi. LAVE: LLM-Powered Agent Assistance and Language Augmentation for Video Editing. In *Proceedings of the 29th International Conference on Intelligent User Interfaces*, IUI '24, pages 699–714, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 979-8-4007-0508-3. doi: 10.1145/3640543.3645143. 7.1

[305] Dennis Wang, Yi-Chieh Lee, and Wai-Tat Fu. "i love the feeling of being on stage, but I become greedy": Exploring the impact of monetary incentives on live streamers' social interactions and streaming content. *Proc. ACM Hum.-Comput. Interact.*, 3(CSCW):1–24, November 2019. 5.6.2

[306] Huanchen Wang, Tianrun Qiu, Jiaping Li, Zhicong Lu, and Yuxin Ma. HarmonyCut: Supporting Creative Chinese Paper-cutting Design with Form and Connotation Harmony. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 979-8-4007-1394-1. doi: 10.1145/3706598.3714159. 7.4.2.2

[307] Piaohong Wang and Zhicong Lu. Let's play together through channels: Understanding the practices and experience of danmaku participation game players in china. *Proc. ACM Hum.-Comput. Interact.*, 7(CHI PLAY):1025–1043, October 2023. 5.2.1, 5.2.2, 5.6.1

[308] Qinyuan Wang, Bruce Gu, He Zhang, and Yunfeng Li. Fairness of large music models: From a culturally diverse perspective. In *2024 IEEE 9th International Conference on Data Science in Cyberspace (DSC)*, pages 706–712, 2024. doi: 10.1109/DSC63484.2024.00105. 6.7.1

[309] Sitong Wang, Zheng Ning, Anh Truong, Mira Dontcheva, Dingzeyu Li, and Lydia B Chilton. Podreels: Human-ai co-creation of video podcast teasers. In *Proceedings of the 2024 ACM Designing Interactive Systems Conference*, DIS '24, page 958–974, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400705830. doi: 10.1145/3643834.3661591. URL https://doi.org/10.1145/3643834.

3661591. 6.2.1

[310] Yunlong Wang, Priyadarshini Venkatesh, and Brian Y Lim. Interpretable Directed Diversity: Leveraging Model Explanations for Iterative Crowd Ideation. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 978-1-4503-9157-3. doi: 10.1145/3491102.3517551. 7.4

[311] Andrew M Webb, Chen Wang, Andruid Kerne, and Pablo Cesar. Distributed liveness: Understanding how new technologies transform performance experiences. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, CSCW '16, pages 432–437, New York, NY, USA, February 2016. Association for Computing Machinery. 4.2.2

[312] Richard Wetzel, Lisa Blum, Wolfgang Broll, and Leif Oppermann. Designing mobile augmented reality games. In *Handbook of augmented reality*, pages 513–539. Springer, 2011. 1

[313] Mikael Wiberg. Methodology for materiality: Interaction design research through a material lens. *Personal and Ubiquitous Computing*, 18(3):625–636, March 2014. ISSN 1617-4917. doi: 10.1007/s00779-013-0686-7. 8.2.2

[314] David Winter. Amplifying stories: The vital role of audio in video narratives, February 2024. URL https://lwks.com/blog/amplifying-stories-the-vital-role-of-audio-in-video-narratives. Accessed: 2025-1-19. 6.1, 6.2.2, 6.4.1

[315] Jacob O. Wobbrock and Julie A. Kientz. Research contributions in human-computer interaction. *Interactions*, 23(3):38–44, April 2016. ISSN 1072-5520. doi: 10.1145/2907069. URL https://doi.org/10.1145/2907069. 7.3.1.2

[316] Wofk, Diana and Ma, Fangchang and Yang, Tien-Ju and Karaman, Sertac and Sze, Vivienne. FastDepth: Fast Monocular Depth Estimation on Embedded Systems. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2019. 3.6

[317] Donghee Yvette Wohn and Guo Freeman. Audience management practices of live streamers on twitch. In *ACM International Conference on Interactive Media Experiences*, IMX '20, page 106–116, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450379762. doi: 10.1145/3391614.3393653. URL https://doi.org/10.1145/3391614.3393653. 4.1, 5.2.2, 5.6.2

[318] Minz Won, Justin Salamon, Nicholas J. Bryan, Gautham J. Mysore, and Xavier Serra. Emotion embedding spaces for matching music to stories, 2021. URL https://arxiv.org/abs/2111.13468. 6.2.2

[319] Shih-Lun Wu, Chris Donahue, Shinji Watanabe, and Nicholas J. Bryan. Music controlnet: Multiple time-varying controls for music generation. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 32:2692–2703, May 2024. ISSN 2329-9290. doi: 10.1109/TASLP.2024.3399026. URL https://doi.org/10.1109/TASLP.2024.3399026. 6.2.3, 6.7

[320] Tongshuang Wu, Michael Terry, and Carrie Jun Cai. Ai chains: Transparent and control-

lable human-ai interaction by chaining large language model prompts. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450391573. doi: 10.1145/3491102.3517582. URL https://doi.org/10.1145/3491102.3517582. 6.2.1, 6.2.3, 6.4.2

[321] Jason Wuertz, Sultan A Alharthi, William A Hamilton, Scott Bateman, Carl Gutwin, Anthony Tang, Zachary Toups, and Jessica Hammer. A design framework for awareness cues in distributed multiplayer games. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, number Paper 243 in CHI '18, pages 1–14, New York, NY, USA, April 2018. Association for Computing Machinery. 4.4

[322] Xiaotong (Tone) Xu, Srishti Palani, Azzaya Munkhbat, Tiffany Lee, and Steven P. Dow. Idea-centric search: Four patterns of information seeking during creative ideation. In *Proceedings of the 16th Conference on Creativity & Cognition*, pages 280–291, New York, NY, USA, 2024. ACM. doi: 10.1145/3635636.3656193. 2.3

[323] Qian Yang, Aaron Steinfeld, Carolyn Rosé, and John Zimmerman. Re-examining whether, why, and how human-ai interaction is uniquely difficult to design. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–13, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450367080. doi: 10.1145/3313831.3376301. URL https://doi.org/10.1145/3313831.3376301. 7.1

[324] Saelyne Yang, Changyoon Lee, Hijung Valentina Shin, and Juho Kim. Snapstream: Snapshot-based interaction in live streaming for visual art. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, pages 1–12, New York, NY, USA, April 2020. Association for Computing Machinery. 5.2.1

[325] Saelyne Yang, Jisu Yim, Juho Kim, and Hijung Valentina Shin. CatchLive: Real-time summarization of live streams with stream content and interaction data. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, number Article 500 in CHI '22, pages 1–20, New York, NY, USA, April 2022. Association for Computing Machinery. 4.2.2, 5.2.1

[326] Zhihao Yao, Qirui Sun, Beituo Liu, Yao Lu, Guanhong Liu, Xing-Dong Yang, and Haipeng Mi. InkBrush: A Sketching Tool for 3D Ink Painting. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 979-8-4007-0330-0. doi: 10.1145/3613904.3642128. 7.4.2.4

[327] Ryan Yen and Jian Zhao. Memolet: Reifying the reuse of user-ai conversational memories. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*, pages 1–22. ACM, 2024. doi: 10.1145/3654777.3676388. 2.3

[328] Ryan Yen, Li Feng, Brinda Mehra, Ching Christie Pang, Siying Hu, and Zhicong Lu. StoryChat: Designing a Narrative-Based viewer participation tool for live streaming chatrooms. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, number Article 795 in CHI '23, pages 1–18, New York, NY, USA, April 2023.

Association for Computing Machinery. 5.2.1

[329] Jing Yi, Yaochen Zhu, Jiayi Xie, and Zhenzhong Chen. Cross-modal variational auto-encoder for content-based micro-video background music recommendation. *IEEE Transactions on Multimedia*, 25:515–528, 2023. doi: 10.1109/TMM.2021.3128254. 6.2.2

[330] J.D. Zamfirescu-Pereira, Richmond Y. Wong, Bjoern Hartmann, and Qian Yang. Why johnny can't prompt: How non-ai experts try (and fail) to design llm prompts. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450394215. doi: 10.1145/3544548.3581388. URL https://doi.org/10.1145/3544548.3581388. 6.1, 6.2.1, 6.6.2

[331] Yongyi Zang and Yixiao Zhang. The interpretation gap in text-to-music generation models, 2024. URL https://arxiv.org/abs/2407.10328. 6.2.3, 6.4.2

[332] ZeroMQ. ZeroMQ. https://zeromq.org/, 2009. Accessed: 2021-4-7. 3.3.2

[333] Chao Zhang, Kexin Ju, Peter Bidoshi, Yu-Chun Grace Yen, and Jeffrey M. Rzeszotarski. Friction: Deciphering Writing Feedback into Writing Revisions through LLM-Assisted Reflection. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 979-8-4007-1394-1. doi: 10.1145/3706598.3714316. **??**, 7.4.2.3

[334] Lei Zhang, Andy Sun, Ryan Shea, Jiangchuan Liu, and Miao Zhang. *Rendering Multi-Party Mobile Augmented Reality from Edge*, page 67–72. Association for Computing Machinery, New York, NY, USA, 2019. ISBN 9781450362986. URL https://doi.org/10.1145/3304112.3325612. 3.2.2

[335] Lei Zhang, Ashutosh Agrawal, Steve Oney, and Anhong Guo. VRGit: A Version Control System for Collaborative Content Creation in Virtual Reality. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 978-1-4503-9421-5. doi: 10.1145/3544548.3581136. **??**, 7.4.2.4

[336] Lei Zhang, Daekun Kim, Youjean Cho, Ava Robinson, Yu Jiang Tham, Rajan Vaish, and Andrés Monroy-Hernández. Jigsaw: Authoring Immersive Storytelling Experiences with Augmented Reality and Internet of Things. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 979-8-4007-0330-0. doi: 10.1145/3613904.3642744. 7.4.1.3, 7.4.2.4

[337] Yixiao Zhang, Gus Xia, Mark Levy, and Simon Dixon. Cosmic: A Conversational Interface for Human-AI Music Co-Creation. In *NIME 2021*, apr 29 2021. URL https://nime.pubpub.org/pub/in6wsc9t. 6.2.3

[338] Yixiao Zhang, Yukara Ikemiya, Woosung Choi, Naoki Murata, Marco A. Martínez-Ramírez, Liwei Lin, Gus Xia, Wei-Hsiang Liao, Yuki Mitsufuji, and Simon Dixon. Instruct-musicgen: Unlocking text-to-music editing for music language models via instruction tuning, 2024. URL https://arxiv.org/abs/2405.18386. 6.2.1, 6.7

[339] Yixiao Zhang, Akira Maezawa, Gus Xia, Kazuhiko Yamamoto, and Simon Dixon. Loop copilot: Conducting ai ensembles for music generation and iterative editing, 2024. URL https://arxiv.org/abs/2310.12404. 6.2.1, 6.2.3

[340] Zheng Zhang, Jie Gao, Ranjodh Singh Dhaliwal, and Toby Jia-Jun Li. VISAR: A Human-AI Argumentative Writing Assistant with Visual Programming and Rapid Draft Prototyping. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 979-8-4007-0132-0. doi: 10.1145/3586183.3606800. 7.4.1.4

[341] Dora Zhao, Michelle S Lam, Diyi Yang, and Michael S Bernstein. Behavior latticing: Inferring user motivations from unstructured interactions. *arXiv preprint arXiv:2604.07629*, 2026. 9.2.1

[342] Mingxu Zhou, Dengming Zhang, Weitao You, Ziqi Yu, Yifei Wu, Chenghao Pan, Huiting Liu, Tianyu Lao, and Pei Chen. StyleFactory: Towards Better Style Alignment in Image Creation through Style-Strength-Based Control and Evaluation. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, UIST '24, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 979-8-4007-0628-8. doi: 10.1145/3654777.3676370. **??**, 7.4.2.4, 8.2.1

[343] Nianmei Zhou, Steven Devleminck, and Luc Geurts. Tangible affect: A literature review of tangible interactive systems addressing human core affect, emotions and moods. In *Proceedings of the 2024 ACM Designing Interactive Systems Conference (DIS '24)*, pages 1–17, IT University of Copenhagen, Denmark, 2024. ACM. doi: 10.1145/3643834.3661608. 7.2.1