(Un)Fairness Along the AI Pipeline

Problems and Solutions

Emily Black

CMU-CS-22-121

July 2022

Computer Science Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Matt Fredrikson, Chair
Alexandra Chouldechova
Rayid Ghani
Hoda Heidari
Solon Barocas (Microsoft Research)

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

This research was sponsored by the Defense Advanced Research Projects Agency under award number HR00112020006, the University of Virginia under award number GF14755PO2261588, PricewaterhouseCoopers, Amazon, and the National Science Foundation under award numbers 1350598, 1704845, and 1943016.

The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government or any other entity.

Keywords: Artificial Intelligence, Machine Learning, Deep Networks, Neural Networks, Deep Learning, Ethics, Fairness, Accountability, Explainability, Public Policy, machine learning pipeline, AI pipeline, stability, consistency, inconsistency, ensembling, counterfactual explanations, leave-one-out unfairness, tax policy, vertical equity, model multiplicity

First and foremost, for my mother and father, with love and gratitude.

In addition, for anyone who has felt that they couldn't go on, that it was too hard, that they weren't good enough, that everything was horribly wrong—and for all of those around them who take the time to shed some light onto their path, to help them see the roses around them, and a way forward.

Abstract

Artificial Intelligence (AI) systems now influence decisions impacting every aspect of people's lives, from the news articles they read, to whether or not they receive a loan. While the use of AI may lead to great accuracy and efficiency in the making of these important decisions, recent news and research reports have shown that AI models can act unfairly: from exhibiting gender bias in hiring models, to racial bias in recidivism prediction systems.

This thesis explores new methods for understanding and mitigating fairness issues in AI through considering how choices made throughout the process of creating an AI system—i.e., the modeling pipeline—impacts fairness behavior. First, I will show how considering a model's end-to-end pipeline allows us to expand our understanding of unfair model behavior. In particular, my work introduces a connection between AI system stability and fairness by demonstrating how instability in certain parts of the modeling pipeline, namely the learning rule, can lead to unfairness by having important decisions rely on arbitrary modeling choices.

Secondly, I will discuss how considering ML pipelines can help us expand our toolbox of bias mitigation techniques. In a case study investigating equity with respect to income in tax auditing practices, I will demonstrate how interventions made along the AI creation pipeline—even those not related to fairness on their face—can not only be effective for increasing fairness, but can often reduce tradeoffs between predictive utility and fairness.

Finally, I will close with an overview of the benefits and dangers of the flexibility that the AI modeling pipeline affords practitioners in the creation of their models, including a discussion of the the legal repercussions of this flexibility, which I call *model multiplicity*.

Acknowledgments

To start, I'd like to thank my advisor and thesis committee. First and foremost, I'd like to thank my advisor Matt Fredrikson, for his patience and for his generosity with his time and advice. Matt doesn't shy away from getting into the weeds on a project, from code to paper writing, and also took time to read over and listen to countless fellowship applications and research talks. Perhaps even more importantly, Matt has helped me discover, and let me follow, my interests as they developed, and has been consistently invested in my success. Thank you so much, Matt, for consistently supporting me and helping me become the researcher I am today. I'm grateful I've been lucky enough to work with you, and get to know you, your lovely family, and adorable pets.

For Alexandra Chouldechova, who practically served as my second advisor for my last year and a half of graduate school—Alex, your work has inspired many of the directions my research has taken, and your compassionate but honest mentorship helped me weather many storms. I can't begin to tell you how helpful our discussions—research related and otherwise—have been. I simply would not be where I am without you, so thank you.

I'd like to thank Rayid Ghani for his guidance and kind words over the years that imbued me with the the confidence I needed to push onward in my career, and the discussions that helped me find the narrative of this work. I'm grateful to have learned from his thoughtful approach to the problems that he pursues.

I'm grateful to Hoda Heidari, among other things, for making CMU in the mid-to-late pandemic again feel like a fun place full of people excited to share ideas. I'm inspired by her ability to create community, both in research projects and the department more generally, and I hope to emulate her in that way, as well as continue to be a part of the wonderful communities she fosters.

I'd like to thank Solon Barocas for introducing me to another style of work and another side of my ideas, which I enjoy thoroughly, and am so excited to pursue. I'm also very thankful for his praise and excitement for my talks and papers, from the (late for U.S. time) email cheering me on *right* after I finished a talk in Korea, to sending me papers relevant to my projects, to sharing my work with others. He is a consistent joy to work with, and I'm grateful for every meeting we have.

To Matt and my entire committee—I hope this is just the beginning of our work together.

I'd also particularly like to thank Klas Leino, my friend, collaborator, and mentor, without whom there is a large chance this thesis would not have been written because I may have dropped out of graduate school. When I came to CMU, not only did I not know what a deep neural network was, but I had never taken a probability class, and I didn't really know how to code. I vividly remember sitting in front of an empty Python coding environment after my first real research meeting where my advisor Matt had suggested I try some (retrospectively very easy) experiments, and not having the slightest clue how to start. So I swallowed any semblance of pride I may have had, and asked the only person I'd met at CMU, Klas, my lab-mate and office-mate, why the function I'd defined—after having just googled how to define functions in Python—was throwing an error.

Since then, Klas has poured endless hours into helping me gain the knowledge and skills I needed to succeed in graduate school—from what a deep network is, to how to become a numpy expert, to what linearity of expectation means. While over time my requests for help became much fewer and farther between, to this day he has always been there to bounce ideas off of for a paper, a proof, or even to read over an important email. Klas, buddy, I can never thank you enough.

I'm incredibly grateful for my collaborators, who have each contributed enormously to my journey, as well as made it a more pleasant one: Hadi Elzayn, Mikaela Meyer, Manish Raghavan, Zifan Wang, Sam Yeom, Priya Donti, Bonnie Fan, Michael Madaio, Joshua Williams, Jacob Goldin, Anupam Datta, and Daniel Ho.

Last but not least, I thank my friends and family. Sofia Bosch Gomez, Klas Leino, Pedro Parades, David Bernal, Mateo Dulce, Daniel de Roux, Mikaela Meyer, Ainesh Bakshi, Raj Jayaram, Goran Zuzic, Filipe Peres, Josh Williams, Ryan Kavanagh, Aymeric Fromherz, Sarah Birmingham, Niki Kennedy, Nick Snyder, Katy Frank, Leyla Wade, Shanti Kumar, Sonia Max, Maxine McGredy, Abigail Friedland, Thienthanth Trinh, Marya Friedman, Conor Hunt, and everyone else: thank you for the parties on the weekend that helped me get through some of those weeks, thank you for picking up the late-night phone calls, thank you for staying my friend when I would periodically drop off the face of the earth to make a deadline, thank you for making me laugh, thank you for 2 a.m. chocolate deliveries, thank you for the dancing, the bike rides, the hikes, the hugs, for "just one drink" on a weeknight, for the dinners—or even lunches—that went so long the day was gone before they were over, for the love and sunshine you bring to my life.

Katy, thanks for taking time off from your real job to come from Philly and cook all my meals for me the days leading up to my defense, and thanks for surprising me at the end with the flowers you sneaked in.

Sofia, thank you for Japonica Way and for that rainy day in Pittsburgh, thank you for the bus that never came. The rest won't fit.

Mom, thanks for the letters, books, and phone calls trying to help me figure it all out, and Dad, thanks, among other things, for reading (and proof reading) this whole thesis.

Contents

II Case Studies in Fairness Interventions along the Modeling Pipeline 59

III Policy and Legal Repercussions of Flexibility in Modeling Decisions 83

Chapter 1

Introduction

Artificial intelligence (AI) now touches almost every part of our lives. AI systems, whether by acting autonomously or by influencing human decisions, now affect everything from day-to-day choices as to the news articles we read, to potentially life-changing events such as whether we receive a loan [30] or whether and where we work [55], or even whether or not we are detained in jail [10].

Algorithms are often introduced to these decision processes in the hopes that they will improve on human decision-making, by increasing accuracy, consistency, and efficiency, and even decreasing bias [113]. Humans themselves are imperfect along many of these axes: human decision-making is biased, as is evident from studies showing that job applicants with black-sounding names are less likely to be called for an interview [19]; human decisions are inconsistent——for example, federal judges who decide the outcomes of asylum claims grant asylum at vastly different rates—from 8 to 98% [113]. Humans are also slow and inefficient—the wait to get an outcome from a benefits appeal claim for veterans can take between 5-7 years, causing complications for those whose health is on the line [113].

However, as AI systems have been adopted into a wider and wider range of applications, it has become apparent that there is cause for caution and concern. We now have ample evidence of AI's potential for significant harm and discriminatory impact. From gender bias in hiring models [55], to racial bias in recidivism prediction systems [10] it is clear that AI systems can exacerbate the same human behaviors they may have been intended to alleviate. In response to this reality, researchers have amassed a growing literature on how to detect and mitigate unfair behavior in AI models [8, 17, 72, 106, 142, 244, 263].

Fairness and Abstraction. Notably, most formalizations of fairness in the AI literature abstract the problem of unfair model behavior to model inputs, model outputs, and the ground truth label. The most common fairness definitions, such as demographic parity, equalized false and/or true positive rates, and equal accuracy, can be expressed only with "observed features V [i.e., input features], outcomes Y [i.e., ground truth labels] ... and decisions D [i.e., model outputs]" [177]. Even slightly less common notions, such as counterfactual fairness or individual fairness, are expressed as a constraint on model inputs and outputs in relation to additional information about the data distribution, such as a given similarity metric or causal graph [177].

This abstraction has many merits: abstraction leads to transportability, and so these definitions can be applied to a large variety of fairness problems. Further, these metrics can be easily computed and tested no matter the model implementation. Beyond these merits, in many contexts, such fairness definitions are also fundamentally important: for example, extreme demographic disparity is of legal consequence in areas such as hiring [235].

The AI Pipeline. However, there may be several more factors which influence the fairness of a model's behavior—for example, the process by which it makes a decision, and, relatedly, the process by which a model is created. The process of creating AI models for a given application is complex: myriad choices are made during model creation, from the profound to the seemingly inconsequential. For example, in order to construct a loan decision model for a bank, a decision has to be made about how to translate the bank's objective, i.e., deciding who to grant loans to, into a prediction target: e.g., predicting the probability of the client's default, or the amount of money gained by the bank per client, or something else entirely. Decisions also have to be made about how to use the available data, such as which features are to be used, and, if the data is unprocessed, how these features should be defined and constructed [197].On a lower level, model practitioners must decide what type of model will be used, the learning rule and objective function, the values of hyper-parameters determining normalization and training procedures, among many other decisions made continuously throughout model development [137, 153]. All of these decisions then impact how the model is constructed, and therefore, how it makes its own decisions.

In this thesis, we call this series of decisions model practitioners have to make the *AI creation pipeline*, or the pipeline. As the pipeline determines how models reach their decision-making schema, we will also refer to the *AI decision pipeline*, or the process by which a model makes a decision about a given input, as the pipeline, differentiating between the creation and decision pipelines when necessary.[1] While choices are often made along the creation pipeline in search of the most accurate possible model, as this thesis and other recent work shows, many of the potential models that could be created among these choices are equally viable (i.e. have equivalent performance) in the given context, while displaying considerably different fairness behavior.

This work aims to present the benefits of explicitly considering the creation and decision pipelines in our conceptualization and mitigation fairness problems in AI systems, which we call a *pipeline-based approach* to algorithmic fairness. We suggest that, while useful in many ways, the abstraction of fairness problems in the AI literature can narrow the field's perspective both on what can constitute unfair behavior, as well as on how unfair behavior can be mitigated in AI systems. By *also* considering a pipeline-based approach in our fairness conceptualization and mitigation, we can uncover and prevent a wider variety of undesirable model behavior, as well as learn to use a larger array of techniques to prevent and stop unfair model behavior in practice. We suggest the pipeline-based approach to algorithmic fairness not as a replacement or improvement upon the literature to date, but rather as an additional, largely unexplored, toolbox. We introduce how the pipeline-based approach can expand fairness conceptualization and mitigation below, before outlining the thesis in detail in Sections 1.2.1- 1.2.3.

[1] We will also interchangeably refer to the AI pipeline as the machine learning pipeline, training pipeline, or learning pipeline.

1.1 Incorporating the Pipeline in Fairness Conceptualization and Mitigation

Expanding Notions of Fairness: Procedural versus Outcomes-Based Fairness. By reducing the problem of unfairness to a question of inputs and outputs, we may narrow our perception of what unfair behavior means to only include those factors. By using a pipeline-based approach, we can uncover unfairness which may arise not only from a system's outcomes, but also by the process by which it reaches a given decision. For example, a decision process may rely on attributes that seem normatively or legally dubious— such as a recidivism prediction model which determines a defendant's ability to walk free between their arrest and their court date based on the criminal history of their parents and friends [10]. Or, as we will explore in depth in this thesis, a decision process may be unfair because it is highly arbitrary or unstable——such as a bank effectively flipping a coin to determine who gets a loan [24]. Even if such models performed admirably on outcomes-based fairness measures, they may still be seen as unfair by many. Further, while the transportability between contexts which comes from abstraction is useful, as we explore in Chapter 5, this generality can come at a cost to specificity, thus at times harming the applicability of common fairness metrics in real-world contexts.

Pipeline-based Fairness Interventions. This abstraction does not only exist in fairness definitions, but also in common unfairness mitigation techniques. Common fairness mitigation techniques are often constructed as *constraints* put on top of a model's decision or optimization process [8, 106, 262]. This approach has many merits: abstracting the problem of fairness to an optimization constraint over a problem with inputs and outputs is the only way to make a fully generalizable solution to a given fairness problem. However, mitigating unfair behavior in this way obfuscates the choices made along the AI pipeline and their effect on the fairness behavior of the model—and thus the potential fairness benefits of changing some of these choices. As we demonstrate in this thesis, decisions made along the AI pipeline have a huge impact on model behavior, from influencing a model's individual predictions, to changing the distribution of the individuals it selects [24, 28, 156]. While these generalized methods of alleviating fairness problems are extremely useful, attacking fairness problems *only* with these tools may leave some stones unturned.

We can, in addition to using commonly available bias mitigation techniques, also explicitly investigate the choices made along the AI pipeline—even seemingly fairness-unrelated ones—as intervention points to improve model fairness behavior. Given that the decisions made along the AI pipeline determine a model and its behavior, if a model is acting unfairly, one way to change that behavior may be to change some choices made along its creation pipeline, rather than adding an additional constraint on top of the model. For example, as we demonstrate in Chapter 3, changing model type to an ensemble model over a singleton model can be preferable for fairness-related stability concerns. Or, as we show in Chapter 5, we can modify a model's prediction target to acheive fairness gains: in that case, changing from a classification to a regression model. Given that, as we show in Chapter 7, there are often myriad equally viable (accurate) models for a given prediction task made with different pipelines, changing decisions made along the AI pipeline, as opposed to adding an additional constraint on top of a model's optimization or decision process, can also lead to fairness gains which come at a lower cost to performance. We demonstrate this phenomenon empirically in Chapter 5. Finally, as pipeline-based fairness interventions are not tied to any one definition of fairness, they can be more flexible to contextual model behavior requirements in real-world systems, as we explore in Chapter 5 as well.

1.2 Thesis Structure

In this thesis, we aim to demonstrate the importance of the AI pipeline in conceptualizing and managing unfairness risks in AI systems. We show how choices made along the AI pipeline—choices often made without fairness specifically in mind—can impact the fairness behavior of AI models. Choices made along the pipeline can lead to unfair behavior in and of themselves, and also, intervening along the AI pipeline can improve fairness behavior, often while reducing performance tradeoffs. The far-reaching impacts of each step in the model creation process, or AI pipeline, gives us an opportunity to leverage each choice we make to increase fairness alongside performance, providing a wealth of opportunities to address fairness issues. Symmetrically, however, by not explicitly considering the impacts of the choices we make along the pipeline, we may unwittingly choose models with suboptimal fairness properties, or worse yet, introduce fairness problems. This thesis contains three parts: Part I explores how choices made along the AI pipeline can lead to fairness problems, and how to mitigate certain such problems; Part II gives two examples of a real-world case study of AI pipeline interventions for fairness; and Part III concludes by investigating the legal repercussions of the flexibility that AI practitioners have when making decisions along the AI pipeline.

1.2.1 Instability and Unfairness: Problems and Solutions

Instability in Machine Learning Models. We start in Part I by showing how the AI pipeline–particularly, *instability* in the AI pipeline–can lead to fairness problems in and of itself. In contrast to many of the outcomes-based conceptions of fairness, however, the sense in which instability in the AI pipeline can lead to unfairness is by casting doubt on the quality and justifiability of the method by which an outcome was reached. In other words, instability in a model's creation pipeline can lead to a type of *procedural* unfairness.

We begin in Chapter 2 by showing that machine learning models—especially deep models—with nearly identical training pipelines, and similar aggregate performance metrics, such as accuracy, can exhibit substantially different behavior on individual points, both in terms of their predictions, and, as we discuss later in the chapter, their explanations.

We start by demonstrating instability in model predictions, and show that effectively equally accurate machine learning models which were trained with arbitrary differences in their creation pipelines, such as a one-point difference in the training set, or a change in random seed, can differ in their predictions or on individual points. In fact, as we show in Chapter 2 a substantial portion of a model's treatment population–between 3-85%—may have their prediction susceptible to change on the basis on such a seemingly arbitrary difference. We also show that in deep models, changes in prediction as a result of these seemingly minuscule changes can happen to points with confident predictions—meaning that the individual points which are subject to this instability are not borderline cases. Further, as we confirm theoretically later on in Chapter 7, we show empirically that accuracy and instability can increase together, particularly as more complex model classes are used to acheive higher accuracy—as instability, or what we term later *multiplicity*, is tightly related to a model's variance.

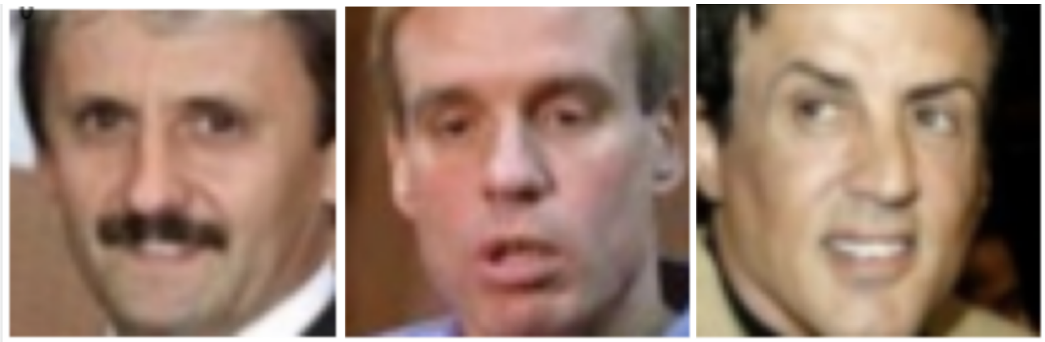


Figure 1.1: From left to right: Individual in a facial recognition model's training set (z), and two individuals in the test set (x, y). When z is included in the training set, the two individuals to the right (x, y) are labeled as a face match with confidence 0.84. When z is not included in the training set, x and y are predicted as *not* a face match with confidence 0.07.

This widespread instability on an individual level can be concerning from a fairness perspective in certain contexts: particularly, contexts which require justifiable decisions, such as credit lending, and government decision making. For example, consider a qualified individual applying for a loan at a bank that uses a machine learning model to determine creditworthiness. Suppose there are multiple equally accurate models, with differing outcomes for that individual (approved or not approved for a loan), only differing in seemingly insignificant ways, e.g. with different random seeds, or a one point difference in the training set. Depending on which model is chosen, the individual will face a different outcome–meaning that whether or not that individual is approved for a loan depends upon which random seed was randomly drawn at training time for the model which the bank ended up using, or perhaps, whether or not an individual who had applied to that same bank for a loan previously had signed a form releasing their data to be used for model development.

This behavior can lead to at least two distinct fairness-related problems, discussed below: a potential lack of quality in a model's decision, and the lack of justifiability of the individual's outcome. In recognition of these fairness problems, we call the phenomenon of an individual's outcome being susceptible to a seemingly arbitrary change in the creation pipeline, particularly a one-point change in the training set, *leave-one-out unfairness*. However, leave-one-out unfairness (LUF) is purposefully not construed as a fairness definition deemed necessary to eradicate for a model to be seen as fair. Instead, we suggest that LUF is a phenomenon that is undesirable in certain circumstances where model justifiability and consistency are of concern.

(1) **A potential lack of quality in a model's decision and/or decision process**

If an individual's decision is susceptible to change based on a change so seemingly insignificant as a random seed or a one-point change in the training set, this may indicate a lack of quality in the model's decision-making process, in that the pattern that was learned to reach that given outcome—even if it is correct—is not stable across small perturbations to the creation pipeline.[2] This sensitivity to seemingly arbitrary choices in the pipeline may indicate a model

[2] Note that we do not use the word robustness, as robustness concerns a model's stability in outcomes to perturbations to a model's *input space*, as opposed to its *creation pipeline*. As we show in Chapter 2, stability over these two types of perturbations do not seem to be strongly related, as models trained to satisfy robustness properties are in fact *less* stable across perturbations to the creation pipeline than deep models

compensating for uncertainty in the data distribution with the given input parameters—i.e. there may not be a strong signal to follow in the available data from which the model can make a generalizable observation. In such instances, more information may be required to reach a higher-quality solution.

Even beyond such concerns, consistency of model outcomes across extremely similar training pipelines may be a desirable property in deployment scenarios where a model will be retrained over time, as is often the case in real-world deployed systems. For example, in human-in-the loop systems where humans and models work closely together, variability in outcomes and explanations may be confusing to users and degrade trust even if aggregate accuracy is maintained. Further, as we explore and address in Chapter 4, consistency may also be desirable when providing instructions to individuals to achieve *recourse* from their algorithmic decisions (e.g. explanations of how a rejected loan applicant could change their application to get accepted in the future)—as in order for recourse to be successful across re-trainings, these instructions must remain valid across perturbations to the creation pipeline.

To address these concerns, in Chapter 2 we introduce a method of mitigating inconsistency in machine learning models—*selective ensembling*—which uses an ensemble of machine learning models drawn randomly from a distribution of similar training environments, coupled with a statistical test, to return the mode prediction of models over a given distribution of training environments, up to a statistical guarantee, or, if the guarantee cannot be met, abstain from prediction. Choosing the mode prediction allows us to provide a theoretical guarantee of the consistency of predictions across models over a given set of perturbations to the training environment. Additionally, the fact that selective ensemble models abstain from prediction on points where a consistency guarantee cannot be met serves as a method of flagging points which may require more information before reaching a high-quality decision—for example, by sending to a human for review. The fact that the mode is also the majority vote over all of the models from a given pipeline may also serve as a better justification for why an AI system returns a given outcome, which we discuss below.

(2) **A (lack of) justifiability of the individual's outcome**

The fact that there are several equally accurate models which lead to different outcomes for this applicant based on an arbitrary difference in training setup, begs the question, *why is this arbitrary choice determining a model's decision?* In other words, *why should one model be chosen over another?* Why should the model that rejects the applicant be chosen over one that does not, given their interchangeability from the perspective of aggregate accuracy? Given that myriad individuals are in this situation—individuals whose decision is up to an arbitrary choice, such as a random seed—the choice of any one model will inevitably mean rejecting certain individuals who would have been accepted in another, equally accurate, model. In situations where outcomes require justification, such as government decision-making, this instability can lead to *arbitrary* decisions: if no thought is given to why one equally accurate model may be chosen over others, despite the fact that the models give different predictions to a substantive portion of the treatment population, these individuals experience unjustified, or arbitrary, decisions. Thus, in order to ensure that despite this fact, everyone is getting justifiable, non-arbitrary outcomes, we must find a way to justify *why* the particular model that was chosen was in fact selected.

Importantly, we note that *arbitrary* and *random* are not interchangeable words in this work.

trained with standard techniques.

By an arbitrary model selection process, we mean a completely unconsidered decision—one that is made without thought or perhaps even without knowledge that a choice was being made. This is different from a random selection process, where a decision is *purposefully* left to chance. We draw this distinction to stress that a random selection process is predicated on a conscious choice to employ this selection method: as Perry and Zarsky [204] write,"the decision to opt for chance must be reasoned." Thus, once a reasoned justification is given as to why one model is selected, even if randomness remains, the element of unfairness which comes from arbitrariness will be mitigated. Beyond a normative desire for justifiable and non-arbitrary decisions, there may be legal precedent that makes this instability problematic from a legal standpoint in the US, particularly in applications such as credit-lending and government decision-making. While we begin discussion of the problem of arbitrariness in Chapter 2, we discuss these ideas in more detail at the end of this work, in Chapter 7, where we also explicitly consider legal repercussions of instability in credit lending, as well as introduce some suggestions for how to justify model selection in high-stakes contexts. In that chapter, we also explore circumstances where the randomness that comes from instability may be harmless, or even desirable—but in order to reap the benefits of randomness, arbitrariness must first be mitigated.

1.2.2 Pipeline Interventions for Bias Mitigation

In Part 2, we turn our focus to how modeling interventions along the AI pipeline can be used to improve fairness behavior—broadly defined—sometimes while even reducing fairness-performance tradeoffs.

In Chapter 5, we present a study which examines issues of algorithmic fairness in the context of systems that inform tax audit selection by the United States Internal Revenue Service (IRS). Income tax revenue is one of the main sources of government funding (51% of government revenue[39]); thus in order for continued government function, US taxpayers must comply with income tax. However, the US tax system does not function perfectly— the annual gross tax gap, or difference between taxes owed and taxes paid, is approximately $440 billion dollars based on data from 2010-2013 [125]. In order to recover lost revenue, and to enforce the law that requires Americans to pay income tax, the IRS audits individuals that it believes may not be paying their full owed tax. During audits, the IRS typically solicits additional information from taxpayers to support information reported on filed returns. For the taxpayer, audits can be time-consuming, stressful, and costly [136, 165]. Low-income taxpayers, for whom tax refunds can comprise a substantial part of income, may wait "on their refunds to pay day-to-day living expenses such as rent, car repairs, or healthcare, and any delay can cause taxpayers significant hardship" [6].

In light of these realities, it is important that the tax audit selection system allocate audits equitably across income lines. As the IRS looks to update their audit selection systems—which have used classification models since the 1970s—our study assesses how the adoption of modern machine learning methods for selecting taxpayer audits may affect equity in tax audit selection with respect to taxpayer income.

Most importantly, we find that simple pipeline interventions—namely, moving from a classification approach to a regression approach to predict the expected magnitude of tax underreporting in the taxpayer population—can improve equity, while sidestepping many performance tradeoffs. In fact, we find that switching from classification to regression models can improve revenue returned to the IRS in comparison to classification models, demonstrating how pipeline-based interventions can reduce the fairness-performance tradeoff by changing the modeling process itself, instead of by adding an additional constraint.

In addition, this case study highlights three important findings relevant beyond the IRS context:

(1) **Conventional fairness concepts are not always a good fit for real-world policy problems.**

In order to evaluate the equity of an audit allocation, we must have an idea of what constitutes an equitable allocation. We find that many of the conceptions of equity commonly used in the AI literature, such as statistical parity, or equalized false positive, false negative, or other error rates are not perfectly suited for understanding income equity in the tax enforcement context, as these algorithmic fairness definitions revolve around notions of treating like individuals or groups alike. This concept of treating like alike is commonly referred to as *horizontal equity* in the tax policy context. While horizontal equity is an important component of tax audit selection, particularly when considering equity across racial lines[78], we find these requirements may not sufficiently describe an equitable distribution of tax along income lines. Instead, we focus on the concept of *vertical equity* —appropriately accounting for relevant differences across individuals—in our work, which is a central component of fairness in many public policy settings.

Vertical equity is not a fairness definition but rather a framework through which to consider fairness desiderata. To become meaningful, vertical equity requires an understanding of "appropriate" accounting for relevant differences, and thus varies from context to context. An example of vertical equity is in the US' progressive income tax: individuals with higher incomes pay a higher percentage income tax. Differences in income between individuals–a relevant difference to tax policy—is accounted for by allowing higher-income individuals to pay a higher percentage tax, in an attempt to distribute more proportionate burdens across individuals, based on economic assumptions about the decreasing marginal utility of each dollar as income increases. We consider the relationship between vertical equity and other more common fairness notions such as individual fairness and proportionality in Chapter 6.

In order to begin to conceptualize vertical equity in the tax audit context, we analyzed incidence of recorded non-compliance in IRS data[3]. We found that the incidence of non-compliance is approximately monotonic in income. Motivated by this fact, our model of vertical equity in the tax audit context includes a preference for audit allocations with audit rates increasing monotonically in income, as well as those displaying high agreement with an *oracle* model which has perfect knowledge of tax misreporting, and selects individuals for audit in descending order of largest misreport.

While we show that common fairness notions, particularly equalized odds and equal true positive rate, are related to this notion of vertical equity in the IRS audit context under certain conditions, we find that enforcing satisfaction of these more common fairness definitions in the literature does not equate to improving more contextualized fairness desiderata in the audit allocation setting—such as reducing audit burden on lower-income taxpayers, or increased agreement with an oracle.

[3] As explained in more detail in Chapter 5, the data used to understand levels of tax non-compliance in the taxpaying population is a random sample of the US population, collected by the IRS for research purposes.

(2) **Conventional algorithmic fairness mitigation techniques are ill-fit for many real-world policy problems.**

Beyond the definitions themselves, we find that standard methods used to implement common notions of algorithmic fairness, are not an ideal fit for the IRS context. While these techniques can mitigate some disparities across income, these come at a steep cost to performance. Additionally, algorithmic fairness mitigation techniques may not perform as expected due to differences in problem set-up—namely, that the IRS audit selection problem is *budgeted*, i.e. only a portion of positive predictions from classification models are selected for audit due to resource constraints, and thus mitigation techniques which guarantee equitable distribution of *all positive predictions for misreporting* across income groups do not often translate into an *audit allocation* which is only a small percentage of the most confident predictions. As many policy problems involve the budgeted distribution of resources, this poses a new challenge for creating fairness mitigation techniques which are effective in such settings. For the time being, this may point to an additional benefit of pipeline-based interventions—they can be deeply contextualized, and are not inherently tied to any one definition of fairness. Thus, pipeline-based fairness interventions may be particularly helpful in on-the-ground bias mitigation projects in the public policy space.

(3) **Agency constraints can have a greater impact on the equity of system performance than any algorithmic attributes.**

We investigate the role of differential audit cost (to the IRS) in shaping the distribution of audits. Audits of lower income taxpayers, for instance, are typically conducted by mail and hence pose much lower cost to the IRS. These results show that the revenue-optimal distribution of audits overwhelmingly focuses on lower- and middle- income individuals, due to the fact that these audits provide a better return on investment. Political pressures to focus narrowly on return-on-investment in the audit context can undermine equity even if the underling algorithm used to select audits is in equitable on its own—an observation which may be relevant to a variety of policy contexts [196], and reinforces the narrative that less biased machine learning algorithms do not guarantee less bias in the decision system of which they are a part.

1.2.3 Legal and Policy Implications

Finally, in Part 3, we discuss in more detail the benefits, downsides, and legal and policy implications of the fact that there are myriad equally viable models which can be created for any given prediction task, as demonstrated in parts I and II of the thesis. We term this availability of equally viable models with different predictive and internal behavior *model multiplicity.* In Chapter 7, we identify two main benefits, and two main concerns stemming from multiplicity: namely, flexibility and improved possibility for recourse, and underspecification and lack of justifiability, respectively. On the one hand, we argue that multiplicity leads to immense flexibility in the model selection process. By demonstrating that there are many different ways of making equally accurate predictions, multiplicity gives practitioners the freedom to prioritize other values in their model selection process without having to abandon their commitment to maximizing accuracy. For example, it may often be possible to satisfy fairness properties on machine learning models at no cost to accuracy, as researchers have shown in increasingly many contexts. We argue that in certain contexts, as a result of the disparate impact doctrine, the fact that several equally accurate models exist for the same prediction task puts legal pressure on model makers to search the space of equally accurate models to find the one that is the least discriminatory.

However, multiplicity also brings to light a concerning truth: model selection on the basis of accuracy alone—the default procedure in many deployment scenarios—fails to consider what might be meaningful differences between equally accurate models. This means that such a selection process effectively becomes an *arbitrary choice*, as we also note in Chapter 2. This obfuscation of the differences between models on axes of behavior other than accuracy—such as fairness, robustness, and interpretability—may lead to unnecessary trade-offs, or could even be leveraged to mask discriminatory behavior. Beyond this, the reality that multiple models exist *with different outcomes for the same individuals* leads to a crisis in justifiability of model decisions: why should an individual be subject to an adverse model outcome if there exists an equally accurate model that treats them more favorably? To remedy this problem, we present methods of building machine learning models which preserve justifiability—in an attempt to answer the question, *how do we take advantage of the benefits model multiplicity provides, while addressing the concerns that it may raise*?

As a whole, this thesis aims to show the immense flexibility that the AI creation pipeline gives practitioners with respect to how to reach the goals they set out to acheive with machine learning models—but also, how the choices made along the AI pipeline have to be carefully considered in terms of their impact on desired model behaviors, as even seemingly small choices can have a large impact. By considering all the choices made along the AI pipeline—from feature selection, to model type, to the objective functions—as places where changes can be made to improve fairness behavior, we can greatly expand our arsenal of unfairness-fighting tools.

Additionally, this work showcases the utility of pipeline-based fairness interventions in deeply contextualized, real-world machine learning bias mitigation: given that pipeline-based interventions are not inherently tied to any one notion of fairness, pipeline interventions can be constructed for a wide variety of desired behaviors. The pipeline-based perspective of looking at fairness problems leads to myriad questions: it is not well understood how each part of the AI pipeline impacts fairness behavior, and to what extent these patterns remain consistent across contexts. In order for pipeline-based interventions to be used with the same efficacy as mainstream fairness interventions, much research must be done to map the choices made along the pipeline to fairness behaviors: this leads to rife opportunities for future work.

1.2.4 Roadmap

This thesis proceeds as follows: we start in Part I with Chapter 2, where we introduce *leave-one-out unfairness* and other forms of potentially undesirable instability in deep creation pipelines. We then address this instability problem through introducing *selective ensembles* in Chapter 3. We then introduce and address the problem of deep learning instability in model explanations in Chapter 4. We then move on to Part II, where we present a case study of income equity in machine learning models applied to the problem of IRS audit allocation in Chapter 5, and follow this with a brief discussion of vertical equity as it relates to other common fairness notions in Chapter 6. Following this, we move to Part III, where we discuss the legal and policy implications of *model multiplicity*. Finally, we conclude (8).

Part I

Inconsistency and Unfairness: Problems and Solutions

Chapter 2

Inconsistency in Deep Model Predictions

In the first part of this thesis, we show how investigating what effects choices made along the AI creation pipeline have on model performance can help widen our understanding of what constitutes unfair behavior. In particular, we demonstrate how instability over perturbations to the creation pipeline can impact individual people's predictions and explanations. We also introduce methods to mitigate these inconsistencies.

Specifically, in Chapter 2 we demonstrate how instability in a model's learning rule can lead to model predictions which change depending on seemingly insignificant changes in the model's training process, such as a one-point difference in the training set, or the random initializations of model parameters. As we expand upon in more detail in Part III, in high stakes contexts such as consumer credit and criminal justice, such decisions may be seen as unfair because they effectively dependent upon these arbitrarily chosen factors, making the model decisions themselves arbitrary. We also demonstrate that, perhaps surprisingly, models trained to satisfy robustness guarantees are in fact *more* sensitive to perturbations to the learning set-up than more standard deep learning models.

In Chapter 3, we introduce methods of alleviating inconsistency in predictions by creating models whose predictions are consistent across small changes to the creation pipeline, which we call *selective ensembles*. Additionally, we demonstrate that certain types of machine learning model explanations—specifically, gradient-based explanations—also suffer from inconsistency as a result of small perturbations to the creation pipeline. We show that both selective ensembles, and also traditional ensemble models, aid consistency in gradient-based explanations as well.

Lastly, in Chapter 4, we show that a different type of explanation technique, namely, *counterfactual examples*, display inconsistency over the same types of perturbations explored in Chapters 2 and 3.This can complicate methods of achieving *recourse* from machine learning model decisions. We introduce a method of generating counterfactual explanations which are much more stable across perturbations to the creation pipeline—*stable neighbor search.*

2.1 Instability and Unfairness

There are several definitions that aim to formalize fair behavior in machine learning contexts: group-based notions, such as demographic parity [83] and equalized odds [105], stipulate that different demographic groups should be treated similarly in aggregate; on the other hand, individualized notions focus on how each person is treated, such as individual fairness [72], which requires "similar" outcomes for similar people, and counterfactual fairness [142], which argues that people should be treated the same as their hypothetical counterpart, who takes a different protected attribute. Fundamentally, *these fairness criteria depend on a comparison of how one group or individual is treated versus another.* However, there are also situations where the decision-making mechanism is unfair not because of how its behavior varies across defined groups or individuals, but rather because its decisions cannot be justified by consistent, intelligible criteria. In other words, decisions may be unfair because they are arbitrary.

In this section, we study the extent to which instability can lead to such fairness issues. Intuitively, when a person's outcome hinges on an arbitrary factor—such as the presence of another, single individual in the training data—the outcome that follows may be viewed as unfair. Take for example a person in reasonable financial health who applies for an auto loan. Suppose that whether their application is approved or not depends on whether another *unrelated* person had applied for a loan from the same bank, and was subsequently included in the training data. Such a decision may be viewed as unfair, as it depends on the willingness and availability of another person to provide their data for training—a chance occurrence, rather than a well-justified set of criteria. Even beyond its potential unfairness, this behvaior may be especially undesireable in applications which come with a "right to explanation" [129].

Measuring leave-one-out Unfairness. To formalize this intuition, we introduce *leave-one-out unfairness* (LUF): the chance that an individual's outcome will change due to the presence of any one instance in the training data (Section 2.3, Definition 2.2). To the best of our knowledge, this is one of the first attempts to formalize unfairness as stemming from the *arbitrary* nature of decision rules, and in particular the stability of the underlying learning algorithm[1]. Importantly, we do not view leave-one-out *fairness*–i.e. the lack of leave-one-out unfairness—as a fairness definition that needs to be satisfied in order for a model to be deemed acceptable in any context. We consider leave-one-out unfairness to be undesirable dependent upon context, which we expand upon in Section 2.1, and again in Chapter 7.

While in this section, we focus on one-point changes to the training set, there are other random choices made during model development that may lead to an arbitrary change in model outcome for an individual—changes in the random initialization or architecture, for example, which we explore in Section 2.6. In Chapter 3, we generalize the problem of inconsistency in machine learning (ML) model predictions to be over any small perturbation—not only one-point changes to the training set— to the *training pipeline*, not only the learning rule. However, we focus on instability with respect to training data in this section due to its theoretical connections to other areas of machine learning literature such as stability, privacy, and robustness.

[1] Contemporary work Marx et. al [168] addresses similar behavior in linear models, but their definition of "predictive multiplicity" is focused on the prediction problem and not the individual.

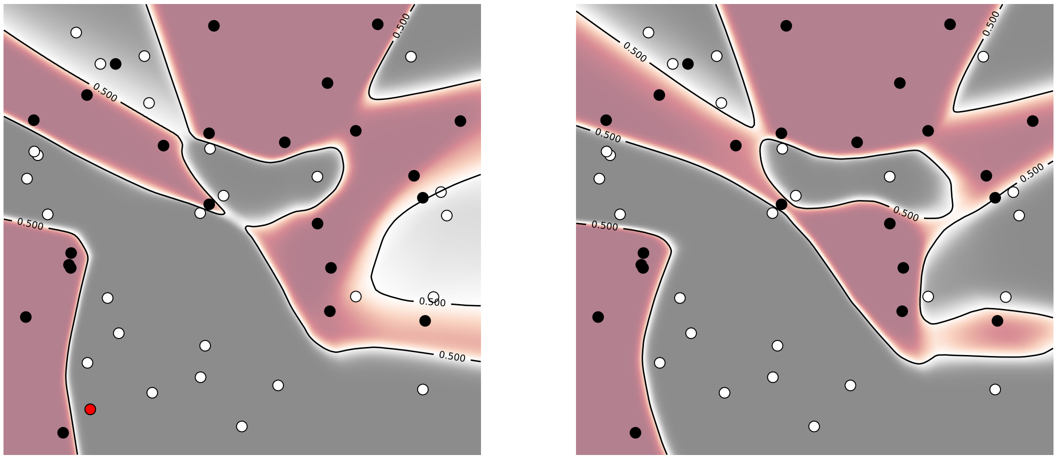


Figure 2.1: Classification boundaries of a deep model with three hidden layers, trained on two-dimensional data with uniform-random binary labels, before (left) and after (right) the point highlighted in red is removed from the training data. Lighter regions correspond to predictions with less confidence. While the model remains largely unchanged in the area around the left-out point, its boundary changes significantly in other, far-away areas. For example, the middle-right region assigns greater confidence to white points, even flipping its prediction on one such point.

We find that in many cases, the use of deep models can lead to this type of unfair outcome with surprising frequency, and can result in different outcomes for seemingly unrelated individuals. To gain an intuition for why this might be, Figure A.1 depicts the decision boundaries of two low-dimensional binary classifiers whose training data differs only on the presence of the point highlighted in red. Notice that the boundary near the left-out point remains fairly consistent, but there are non-trivial differences in both the boundary locations and the confidence of the model's predictions in regions away from the point. While this low-dimensional example provides some intuition, we systematically characterize the extent to which deep models behave as such on real data (Section 2.4). We find that it occurs often enough to be a concern in some settings (i.e., up to 7% of data is affected); that it occurs even on points for which the model assigns high confidence; and is not consistently influenced by dataset size, test accuracy, or generalization error (Figure 2.4, Table 3.1).

Connections. Leave-one-out unfairness has useful connections to other fields such as stability, privacy, and robustness. We show that while LUF is strictly stronger than some prior notions of leave-one-out stability [223] (Section 2.3.3, Proposition 2.2), it is *weaker* than differential privacy [70] (Proposition 2.3). Thus, one can achieve bounded levels of leave-one-out unfairness by satisfying differential privacy, but it may also be possible to do so via relaxations that allow greater flexibility in the selection of learning rules [176].

Recent work has related robust classification to desirable properties beyond mitigating adversarial examples [236], such as the encoding of more human-interpretable features [82, 119, 185, 238], and individual fairness on weighted ℓ_p metrics [260]. These results may seem to suggest that robust models would also be less susceptible to leave-one-out unfairness. Evaluating two common techniques for producing robust models, adversarial training [166] and randomized smoothing [46], we find that these methods in fact have *vastly different* effects on leave-one-out unfairness. Whereas randomized smoothing tends to have no effect, adversarial training amplifies the problem, resulting in up to a factor of five more affected points (Section 2.5). These results suggest that although LUF and robustness are not inherently tied to each other, certain types of models may prove beneficial for both.

Summary. In a similar vein to the oft-cited "lack of interpretability" [159], leave-one-out unfairness, and the wider observed pattern of prediction inconsistency, complicates the responsible application of deep models to sensitive decisions. Particularly in settings where a well-justified explanation is desirable, or even legally mandated [29, 44, 220] these complications may need to be weighed against the benefits that deep models provide over less complex alternatives. While we largely focus on presenting the phenomenon of prediction inconsistency in this section, we discuss the benefits and harms of this behavior, as well as its legal and policy repercussions, in Chapter 7.

To summarize, we present the following contributions:

1. We introduce and formalize *leave-one-out unfairness*, which characterizes a possible source of unfair, arbitrary outcomes in ML applications.
2. We relate leave-one-out unfairness to well-known prior notions of stability, shedding light on when models may suffer from leave-one-out unfairness, and techniques that might help to mitigate it.
3. Finally, we present an extensive evaluation of how prevalent LUF is when deep neural networks are trained on a variety of datasets, and compare it to other sources of instability such as random initialization and choice of architecture.

In Section 2.2, we provide two examples of machine learning applications where leave-one-out unfairness may lead to unjust model behavior, along with experimental results demonstrating that LUF indeed may occur in these contexts, as well as a brief discussion of when prediction inconsistency may in fact be beneficial. Following this, in Section 2.3, we formally define leave-one-out unfairness and explore its relationships to LOO-stability and differential privacy. In Section 2.4 and Section 2.5, we present our experimental results of the extent of leave-one-out unfairness on real datasets for conventional and robustly trained machine learning models.

2.2 Contextualizing Leave-one-out Unfairness

Leave-one-out unfairness may not pose a problem in all machine-learning applications. If the model's outcome is of little consequence to peoples' lives, or if the application context does not require consistency across data samples for adequate justification, then inconsistent or even arbitrary predictions may be acceptable. Determining whether or not leave-one-out unfairness leads to fairness issues requires considering this context. In this section, we motivate examples of how leave-one-out unfairness constitutes a fairness issue in two contexts: facial recognition use by law enforcement, and loan application decision models used by financial institutions. We also discuss situations where leave-one-out unfairness may be unimportant or even beneficial.

Prior to delving in to these examples, however, we clarify the difference between *arbitrariness* and *randomness*. By an arbitrary decision, we mean a completely unconsidered decision—one that is made without thought or perhaps even without knowledge that a choice was being

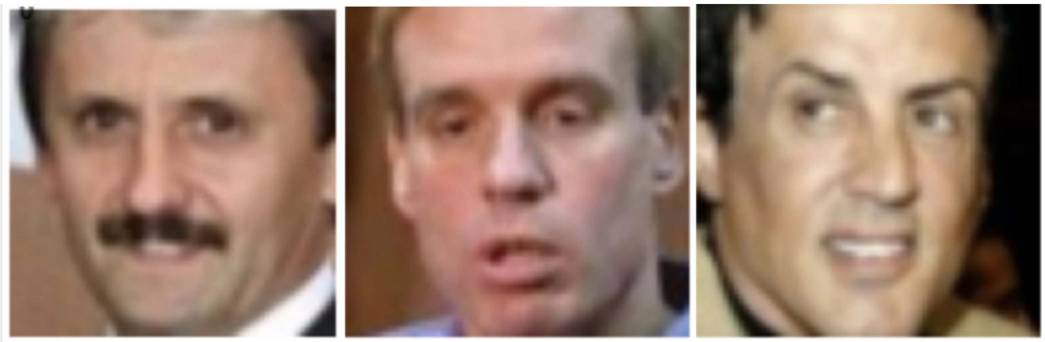


Figure 2.2: From left to right: Individual removed from the training set (z). When z is included in the training set, the two individuals to the right (x, y) are labeled as a match with confidence 0.84. When z is not in the training set, x and y are predicted as *not* a match with confidence 0.07.

made. This is different from a random decision, where a decision is *purposefully* left to chance. We draw this distinction to stress that a random selection process is predicated on a conscious choice to employ this selection method: as Perry and Zarsky [204] write,"the decision to opt for chance must be reasoned." This distinction is of importance in the case of leave-one-out unfairness; as in some cases, both randomness and arbitrariness pose a problem, in others, only arbitrariness, and sometimes, neither.

Consider again a situation where an individual's prediction outcome–such as their loan application decision—is changed as a result of a seemingly inconsequential change such as a one-point change in the training set. The crux of the problem of justifiability that arises from leave-one-out unfairness in deep models is how to answer the question: *why was one model chosen over another?* As Barocas et al. [220] have previously argued, in order to fully justify a model's decision, "one must seek explanations of the process behind a model's development, not just explanations of the model itself". If one train-test split occurred, or a random seed was chosen, without giving *thought or knowledge* to the fact that those choices alone could lead to changes in prediction, then decisions from the resultant models may be seen as arbitrary.

In situations where model explanations are desired or required, such as in loan application decisions [3, 93, 188], this arbitrariness may be problematic, as we discuss in more detail in Chapter 7. However, here the *randomness* from leave-one-out unfairness may not inherently be an issue—as long as there is a good justification for why a particular model was chosen in order to fully justify any given prediction. However for some types of decisions where machine learning models are being introduced, such as government decisions, and especially those related to criminal justice, the both the arbitrariness and the randomness may be a problem, as there is very little legal tolerance for randomness in these decisions [204]. We explore examples of leave-one-out unfairness in a criminal justice context, and a credit approval context, below.

2.2.1 Facial Recognition

Facial Recognition Technology (FRT) has proliferated in recent years as a method of verifying identity at scale. Its use in law enforcement, and the potential harms that may follow, have gained particular attention due to the potentially dire consequences of misidentification:

facial recognition matches have been used as evidence for arrest [112, 246]. Moreover, the use of this technology in this context is becoming prevalent: according to a study from 2016 [92], at least one in four police agencies in the United States have made use of it.

Background. The use of FRT by law enforcement relies primarily on *face-matching* models, where two face images are provided as input to determine whether they depict the same individual. Note that this differs from *face classification* models, which aim to identify the person depicted in a face image from a pre-determined set of individuals. A typical workflow proceeds as follows: given an image of a suspect, law enforcement queries a face-matching model against a large set of images in a database, which also contains identifying information. The face-matching model provides a binary label, with a confidence score, and the most confident matches are provided to the operator for further review [219].

Many police agencies use ready-made, third-party models. For example, one such third-party, Clearview AI, reportedly contracts with approximately 2,400 law enforcement agencies [163]. Such third-party models are often trained on images obtained from public sources like the Internet, in particular by taking advantage of Creative-Commons licenses widely used on social media websites. [181]. The database of images on which these models are run during inference are often obtained from public records such as drivers license databases. Notably, these databases may largely consist of individuals with no prior criminal record [92].

Impact of Instability. The results of FRT are increasingly being used by law enforcement as evidence to justify arrest [112, 246]. If it is likely that a matching outcome can change due to the inclusion of a particular image–unrelated to the suspect or the potential match—out of tens of thousands in the model's training set, then it may be argued the evidence used to justify the eventual arrest is based on an arbitrary occurrence—the use of a particular randomly selected training set. In short, such an outcome would be unfair due to the arbitrary nature of the supporting evidence. Later in this Chapter, we formalize this behavior, and investigate its prevalence on models trained on real datasets, including face-matching models.

Experimental Confirmation. We trained a face-matching model on Labeled Faces in the Wild (LFW) [115], consisting of 13,000 unconstrained pictures of 1680 different individuals. To measure the effect of individual images on prediction outcomes, we trained models both with and without a randomly sampled individual, controlling for all sources of non-determinism (e.g., parameter initialization and GPU operations). We repeated this experiment for 25 different randomly sampled individuals, and measured the effects on prediction behavior. Further details of our methodology are given in Section 2.4.

We found that the predictions given by the face-matching model change across datasets with single-image differences, with surprising frequently. One such example of this behavior is shown in Figure 2.2. When person z is included in the dataset, persons x and y are labeled as a match; but when person z is removed, they are not. Persons x and y are clearly different from one another, and aside from gender, share few salient characteristics. More surprisingly, both predictions are made with high confidence—0.84 and 0.07–far from a baseline random guess. Such behavior was not limited to these images, but rather we observed that 12% of the model's predictions changed across datasets differing in one image, while the change in accuracy remained less than 2%. Moreover, this behavior was consistent across changes in

	age	*education*	*occupation*	*sex*	*capital gain*	*model conf.*
Affected point (x)	51	Bachelors	Self-employed	F	0	0.87
LOO point (z)	39	11th Grade	Service Industry	M	0	-

Table 2.1: Selected feature values for a point treated leave-one-out unfairly in a deep model on the Adult dataset, and the point z whose removal resulted in the change in prediction. Confidence refers to the raw output of the model's prediction in the model with z.

random initialization and choice of architectures, including a residual network resembling ResNet50.

2.2.2 Consumer Finance

Machine learning is also finding uses in consumer finance [12, 14, 225, 229]. Not surprisingly, the predictions made by these models, too, can greatly impact peoples' lives, potentially playing a decisive role in their ability to buy a car, a house, or start a business.

Impact of Instability Models used in this context may be expected to have consistent, justifiable reasons for the predictions that they make. The disquieting prospect that consumers' access to credit might rest on decisions made without adequate care was one of the main concerns that motivated the passage of FCRA and ECOA [3, 188], both of which target arbitrariness in lending decisions [1]. FCRA and ECOA provide a "right to explanation" in lending decisions, on the belief that having to justify their decisions will cause lenders to be less arbitrary in their decision making [220]. While the letter of these laws focus on the explanation of a given model, many have argued that true protections against arbitrariness also require a justification of the model building process [44, 50, 220].

Experimental Confirmation As with the face-matching model in the previous subsection, we conducted experiments on models trained to predict a proxy for creditworthiness using datasets differing in a single instance. We used the UCI Adult dataset [67], consisting of a subset of US census data, and trained one-hidden-layer neural networks with 200 internal units to predict income from demographic, education, and employment information (details in Section 2.4). Our results suggest that the predictions of these models are often sensitive to the presence of single instances, indicating the potential for *leave-one-out unfairness.*

Looking more closely at the results, one of these models was trained with the point z shown in Table 2.1 included in the training set: a 39-year-old man with an 11th-grade education who works in the service industry. This model predicts that a 51-year-old, college-educated, self-employed woman makes more than $50k (0.87 confidence), whereas a model trained on the same data *without* z made the opposite prediction. Mirroring our findings with the FRT models, there is no apparent connection between the features that represent these individuals (see Table 2.1), and the models predict the woman's outcome with high confidence. The removal of this one individual does not just affect this 51-year-old woman, but rather we find that approximately 2% of the entire data set, 603 predictions, are changed.

2.2.3 When LUF May Not Pose a Problem

In situations where the outcome of a machine learning model is not necessarily of critical importance, for example, ad distribution systems, instability in model predictions may not serve as a problem at all. Indeed, leave-one-out unfairness and related instability may in fact serve as a natural bulwark against *algorithmic monoculture* [51, 138], a phenomenon where all actors in a given application space converge on the same or extremely similar models. For example, if all online advertisers converge on one model (and thus show the same types of ads to each individual or demographic group), consumers may be systematically prevented from learning about products that may be useful to them, or even worse, kept in the dark about relevant employment or housing opportunities, due to quirks of that particular model. Leave-one-out unfairness demonstrates that even nearly identical models can differ substantially in their predictions—potentially preventing algorithmic monoculture in situations where it may be of concern. However, we note that the *justifiability* concerns which leave-one-out unfairness raises may still pose a problem even in situations impacted by algorithmic monoculture—and that solutions for both of these concerns do not oppose each other, as we discuss further in Chapter 7.

2.3 Leave-One-Out Unfairness

In this section, we introduce the definition of leave-one-out unfairness, and discuss its connections to prior notions of stability: leave-one-out stability [223], differential privacy [70], and individual fairness [72]. We prove that leave-one-out unfairness is a stronger notion than leave-one-out stability, and weaker than differential privacy. Our formalization of LUF allows us to measure its prevalence objectively on real data, and our investigation of its connections to other forms of stability suggest mitigation techniques as well potential middle ground for achieving gains in privacy.

2.3.1 Notation and Preliminaries

We assume a typical supervised learning setting. Let $z = (x, y) \in \mathbf{X} \times \mathbf{Y}$ be a data point, where x represents a set of features and y a response. Points z are drawn from a distribution $\mathcal{D}$, as are datasets S from the iid product of $\mathcal{D}$, i.e. $S \sim \mathcal{D}^n$. We assume that learning rules h are randomized mappings from datasets S to models h_S, which are functions mapping features to responses; in other words, $h_S : \mathbf{X} \to \mathbf{Y}$ is the model obtained by learning with h on data S. We use $U(m)$ to refer to the uniform distribution over the integers $\{1...m\}$. Given S sampled from $\mathcal{D}^n$ and index $i \sim U(m)$, we denote the sample S with the ith element removed as $S^{(\setminus i)}$.

2.3.2 Leave-one-out Unfairness

Leave-one-out unfairness is based on the notion that a model's treatment of an individual should not depend too heavily on the inclusion of any other single training point. This is related to the concept of algorithmic stability, which measures the effect that a small change in input has on an algorithm's output. For example, a machine learning algorithm is *stable* if a small change to its input (training set) causes limited change in its output (a trained model). Usually, the change in output is measured in the form of model error. Definition 2.1 formalizes this as *leave-one-out (LOO) stability*, but we note that there are several variants that quantify over pointwise *replacement* instead of leave-out, and use different types of aggregation in their bound [223].

Definition 2.1 (Leave-one-out (LOO) Stability [223])**.** *Let $\epsilon_{stable} : \mathbb{N} \to \mathbb{R}$ be a monotonically-decreasing function. Given a training set $S = (z_1, \ldots, z_m) \sim \mathcal{D}^n$, and a training set*

$S^{(\setminus i)} = (z_1, \dots, z_{i-1}, z_{i+1}, \dots, z_m)$ with $i \sim U(m)$, a learning rule h is leave-one-out-stable *(or* LOO-stable*) on loss function ℓ with rate $\epsilon_{stable}(m)$ if*

$$\frac{1}{m}\sum_{i=1}^{m} \mathop{\mathbb{E}}_{S\sim\mathcal{D}^n}[|\ell(h_S, z_i) - \ell(h_{S^{(\setminus i)}}, z_i)|] \leq \epsilon_{stable}(m)$$

LOO-stability records the average effect of removing an individual from the training set on the absolute loss on that individual's prediction. Quantifying the effect of learning rule instability on the fairness of predicted outcomes, however, calls for a definition focusing on different aspects of model behavior. LOO-stability is a predicate on a learning rule that can be satisfied in order to achieve an acceptable level of model stability, in expectation over all draws of a training set S. However, in this chapter, we are interested in quantifying the extent of arbitrariness in a particular individual's prediction—to capture this, we need a *metric* of unfairness, rather than a fairness guarantee. Pursuant of capturing an particular individual's real-life experience with a particular model, we are interested in a quantifying arbitrary behavior in relation to a particular model context–i.e., on a fixed training set S.

To focus the effect of instability on the experience of the population on which it is deployed, rather than a measure of model performance, we need a metric which accounts for the instability that arises for *any* person from the inclusion of a given point in the training set—rather than the impact that the changed point has on the error of its *own* prediction. Even with this expanded notion of what comprises instability to focus on the experience of the individuals, an aggregate calculation such as in LOO-stability may hide the experiences of an unlucky few who may encounter particularly high arbitrariness in their outcome. To ensure that model behavior on every individual is considered, a worst-case metric is more suitable. Further, appealing to the intuition that a model acts unfairly if it is arbitrary, the *consistency* of its prediction, rather than its loss, is the target; consistent predictions, even when incorrect, suggest that the model's decision is not arbitrary. Definition 2, below, reflects these considerations.

Definition 2.2 (Leave-one-out Unfairness (LUF))**.** *Let D be the distribution from which the training set S is drawn, and let x be in the support of D. We define the leave-one-out unfairness (LUF) experienced by x under learning rule h and training set $S \sim D$ to be:*

$$\mathrm{LUF}(h, S, x) = \max_{i,k} |\Pr[h_S(x) = k] - \Pr[h_{S^{(\setminus i)}}(x) = k]|$$

The randomness in this expression is over the choices made by h. Note that in cases of a deterministic learning rule, $\Pr[h_S(x) = k]$ *is* 0 *or* 1.

In other words, given a learning rule h and a training set S, the LUF experienced by a person x is the worst-case probability that x receives a different prediction in a model trained with h on S, and one trained with h on S with a single point removed. Intuitively, this is one way of quantifying the arbitrariness of the model's decision at x. If LUF is high, then the model's decision is brittle under small, potentially irrelevant changes, i.e., a one-point change in the model's training set–casting doubt on the reason behind the model's decision.

We note that leave-one-out unfairness may have less normative, and technical, significance if there is a noticeable difference in accuracy between the two models h_S and $h_{S^{(\setminus i)}}$. In practice, we find that models with such small changes between them most often have comparable accuracy, as shown in the variances of models trained with various leave-one-out differences in Section 2.4. Additionally, as we prove in Section 2.3.3, LOO-stable models can exhibit leave-one-out unfairness, so the assumption that the two models would have similar performance is supported by theoretical intuition as well. Thus, we assume throughout this chapter that the two models will have comparable accuracy. While the definition could be re-cast to include an accuracy constraint, we omit one to retain similarity to LOO-stability and differential privacy, in order to draw our theoretical connections.

In certain situations, such as when evaluating various models during development, it may be useful to understand the extent of leave-one-out unfairness across the entire population under a given learning rule: i.e. understanding how likely it is *any* individual in the distribution will experience an arbitrary decision. This motivates the concept of *expected* leave-one-out unfairness, defined below. As most of our experiments aim to measure the frequency and severity of arbitrary behavior across real datasets, we will focus most heavily on this definition throughout the chapter.

Definition 2.3 (Expected Leave-one-out Unfairness)**.** *Let D be the distribution from which the training set S is drawn, and let x be drawn randomly from D. We define the expected leave-one-out unfairness (LUF) experienced by x under learning rule h and training set $S \sim D$ to be:*

$$\mathbb{E}_x[\text{LUF}(h, S, x)] = \mathbb{E}_{x \sim D}[\max_{i,k} |\Pr[h_S(x) = k] - \Pr[h_{S^{(\setminus i)}}(x) = k]|]$$

Where the randomness in the expectation is taken over samples of x from D.

2.3.3 Connections to Existing Stability Notions

While our introduction of Definition 2.2 above is clearly motivated by LOO stability, in this section we explore the connections to this and other forms of stability in greater depth. Specifically, we demonstrate that while learning rules that are already known to be leave-one-out-stable may still be susceptible to leave-one-out unfairness, strategies for ensuring stronger notions of stability, such as differential privacy, can be used to mitigate LUF. We also explore the connection between LUF and other individual-based fairness notions, i.e. individual fairness.

LOO Stability. Leave-one-out stability is a coarser notion than leave-one-out unfairness, as it records the average change in a model's error on *a given point* when that same point is removed from the training set. Meanwhile, LUF focuses on how a certain point's model outcome can change as a result of *any* point in the training set being removed.

A LOO-stable model may still treat points leave-one-out unfairly: a model can exhibit similar error *on a given point* before and after that point is removed from the training set, but it may treat other points differently. We demonstrate this point on the simple learning rule and distribution in Figure 2.3. Additionally, the fact LOO-stability is averaged over the entire training set can obscure the fact that some individual points are strongly affected by a small change in the training set. Proposition 2.1 formalizes this, showing that LOO-stability is strictly weaker than LUF.

Proposition 2.1. *Let h be a learning rule, ℓ be 0-1 loss, and $\epsilon(m)$ be a montonically-decreasing function such that h is leave-one-out stable with rate $\epsilon(m)$ for all $S \sim \mathcal{D}^m$. Then there exists a training set S such that $\mathbb{E}_x[\text{LUF}(h, s, x)] > \epsilon_{stable}(m)$ and x.*

Proof. Consider a binary classification problem a discrete distribution D with three points, as pictured in Figure 2.3: $x_1, x_2 \in D$ are of class 0, and $x_3 \in D$ is of class 1, shown in red and blue. We define a learning rule, h, according to the different classifiers learned with each possible training set $S \sim D$, shown in Figure 2.3. Notice that this learning rule is LOO-stable with $\epsilon_{\text{stable}}(3)$=0, as when each point is removed, the classification error *on that point* remains the same: this is shown by construction in Figure 2.3 when $S = x_1, x_2, x_3$, and in all other cases, the learning rule is constant, as shown in the figure. Thus, $\frac{1}{3}\sum_{i=1}^{3} \mathbb{E}_{S\sim\mathcal{D}}[|\ell(h_S, z_i) - \ell(h_{S^{(\setminus i)}}, z_i)|] = 0 \leq 0$. However, notice that e.g., if $S = x_1, x_2, x_3$, and x_3 is removed, x_2 experiences a change in classification outcome. Thus, $\text{LUF}(h, S, x_2) = 1$. See that, in fact, every point is susceptible to a change in prediction as the result of different point being removed from the dataset—thus, $\mathbb{E}_x[\text{LUF}(h, S, x)] = 1$. □

Proposition 2.2 shows that models with bounded LUF are also LOO-stable; the proof is given in the supplementary material.

Proposition 2.2. *Let h be a learning rule, ℓ be 0-1 loss, and $\epsilon(m)$ be a montonically-decreasing function such that $\text{LUF}(h, S, x) \leq \epsilon(m)$ for all $S \sim \mathcal{D}^m$ and x. Then h is leave-one-out stable with rate $\epsilon(m)$.*

Differential Privacy. Privacy and fairness are related in various ways, as others have illustrated before [58, 72]. Like differential privacy, leave-one-out unfairness is a stability property of learning rules, but differential privacy is stronger. In particular, differential privacy (Definition 2.4) quantifies universally over all pairs of related training data, and limits the probability of any change in outcome. On the other hand, Definitions 2 and 3 fix a training set, and require stability of the model's response on points from the target distribution.

Definition 2.4 ((ϵ, δ) -Differential privacy)**.** *An algorithm $A : \mathbf{X} \to \mathbf{Y}$ satisfies (ϵ, δ)-differential privacy, for $0 < \epsilon$ and $\delta \in [0, 1]$, if for all $S \in \mathbf{X}^n$, $S' \in \mathbf{X}^{n-1}$ that differ in a single row and all $Y \subseteq \mathbf{Y}$, $\Pr[A(S) \in Y] \leq e^{\epsilon} \Pr[A(S') \in Y] + \delta$.*

Differential privacy is stronger than leave-one-out-unfairness, as any change to the model—even if it does not actually affect prediction of any point in the distribution—can potentially leak information, and is therefore a violation of differential privacy. This makes sense in the context of privacy, as it concerns an adversarial setting where an attacker is free to interact with a model as-needed to extract information. The focus of fairness is how people receiving an outcome from a model are treated, and thus leave-one-out unfairness focuses on the model's behavior on the data distribution, drawing attention to how changes in the model could affect those who are its likely subjects.

Leave-one-out unfairness does not require randomization in the model's learning rule, whereas differential privacy does. Figure 2.3 shows an intuitive example of this, where the deterministic learning rule may yield models with unstable outcomes, but only on points with vanishing probability; for points with non-zero probability, the model's predictions will remain consistent across unit changes to the training data. Moreover, because Definition 2.1 depends on $\mathcal{D}$, a learning rule may have little leave-one-out unfairness on some distributions, and more on others. However, as Proposition 3.2 shows, differential privacy implies bounded LUF. A proof can be found in the supplementary material.

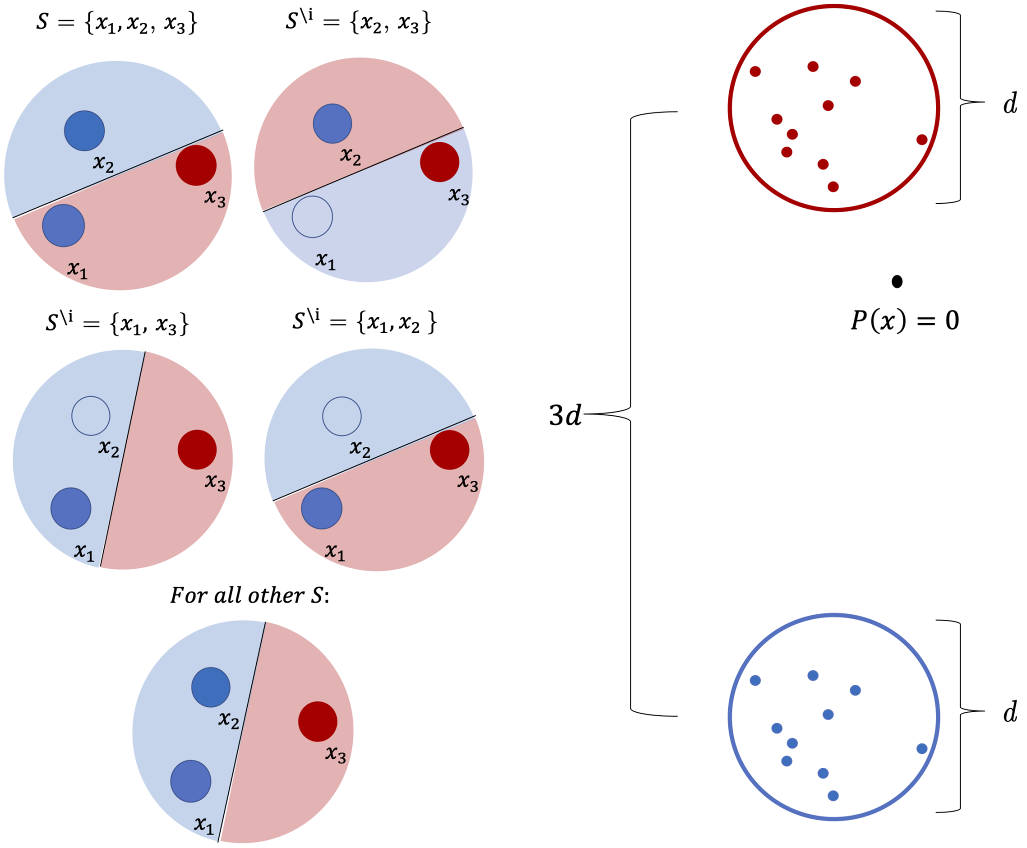

$S = \{x_1, x_2, x_3\}$
$S^{\setminus i} = \{x_2, x_3\}$
$S^{\setminus i} = \{x_1, x_3\}$
$S^{\setminus i} = \{x_1, x_2\}$
For all other S:
x_1
x_2
x_3
$3d$
d
d
$P(x) = 0$

Figure 2.3: Left: A learning rule h that satisfies LOO-stability, but not expected LUF, over the distribution D of the three points pictured. In each box, we see the decision boundary learned with a specified training set $S \sim D$, thus fully defining h. The proof is explained in Proposition 2.1. Right: Visual intuition for how a model can have $LUF = 0, \forall x \in D$ but not satisfy differential privacy. Consider a 1-KNN model on a binary classification problem over the distribution pictured above: two perfectly separated uniform distributions over circles. The diameter of each circle is d, and the distance between the centers of the two circles is $3d$. Consider any training set S drawn from this distribution that has at least two data points from each class. See that $\text{LUF}(h, S, x) = 0$ for all $x \in D$: removing any point from S cannot change the classification of any point *in the distribution*, i.e., within the circles pictured above. However, 1-KNN is not differentially private, as it is a deterministic, non-constant, learning rule. Specifically, see that adding or removing a point in S *can* shift the boundary sufficiently far to change the model's behavior on points *not* in D, (such as point x_2 pictured), which is a violation of differential privacy.

	Deep		**PGD**		**Trades**		**Smoothed**		**Linear**	
dataset	*base acc*	*gen err*	*base acc*	*gen err*	*base acc*	*gen err*	*base acc*	*gen err*	*base acc*	*gen err*
German Credit	0.7500	0.2500	0.7400	0.22	0.745	0.253	0.755	0.245	0.745	0.0175
Adult	0.8418	0.0344	0.8226	-0.0019	0.83217	0.0845	0.8390	0.0180	0.8400	0.000
Seizure	0.9736	0.0264	0.9770	0.000	0.9672	0.0083	0.9754	0.0246	0.8113	0.0043
FMNIST	0.9111	0.0211	0.7876	0.0099	0.9016	0.0700	0.8678	0.0269	0.8368	0.0145
LFW	0.8695	0.0597	-	-	-	-	-	-	0.5790	-0.0755

Table 2.2: Test accuracy and generalization error for all h_S models.

Proposition 2.3. *Let h be an (ϵ, δ)-differentially private learning rule, and $x \sim \mathcal{D}$ be a point. Then* $\mathrm{LUF}(h, S, x) \leq e^{\epsilon} - 1 + \delta$*.*

Individual Fairness. Individual Fairness is a Lipschitz condition that aims to formalize the maxim: "similar people ought to be treated similarly". Importantly, in the context of supervised learning this is typically construed as a constraint on *models* rather than learning rules. This stands in contrast to Definitions 2 and 3, which impose a constraint on the latter. Additionally, our definitions do not relate the treatment of individuals to others, but instead measure the degree to which one's treatment by the model may be arbitrarily decided by the composition of the training data. While there is no reason that individual fairness and leave-one-out fairness cannot coincide, there is no a priori reason to believe that they will. In Section 2.5, we present experimental results on models trained with random smoothing, which has been shown to guarantee individual fairness [260]; shedding further light on the relationship between these two fairness concepts.

We note that leave-one-out unfairness is also related to the definition of memorization introduced by Feldman [84], which we discuss in greater detail in Section 2.7.

2.4 LUF in Deep Models

We characterize the prevalence of leave-one-out unfairness across models trained on several types of data: tabular, time-series, and image data. Importantly, we find that a non-trivial fraction of data (from 3% to 77%) experiences LUF, and moreover, that the prevalence does not appear to depend on model generalization, test accuracy, or dataset size.

Datasets. We perform all of our experiments over five datasets: *UCI German Credit* [67], *Adult* [67], *Seizure* [67], *Fashion MNIST* [255], and Labeled Faces in the Wild [115]. The German Credit data set consists of individuals' financial data, with a binary response indicating their creditworthiness. The Adult dataset consists of a subset of publicly-available US Census data, with a binary response indicating annual income of > 50k. The Seizure dataset comprises time-series EEG recordings for 500 individuals, with a binary response indicating the occurrence of a seizure. Fashion MNIST contains images of clothing items, with a multilabel response of 10 classes. Labeled Faces in the wild consists of unconstrained pictures of individuals' faces, with labels connoting the identity of the individual in each picture. Further information about these datasets and the preprocessing steps we apply can be found in the supplementary material. Table 3.1 contains the accuracy and generalization error for each baseline model h_S for all datasets.

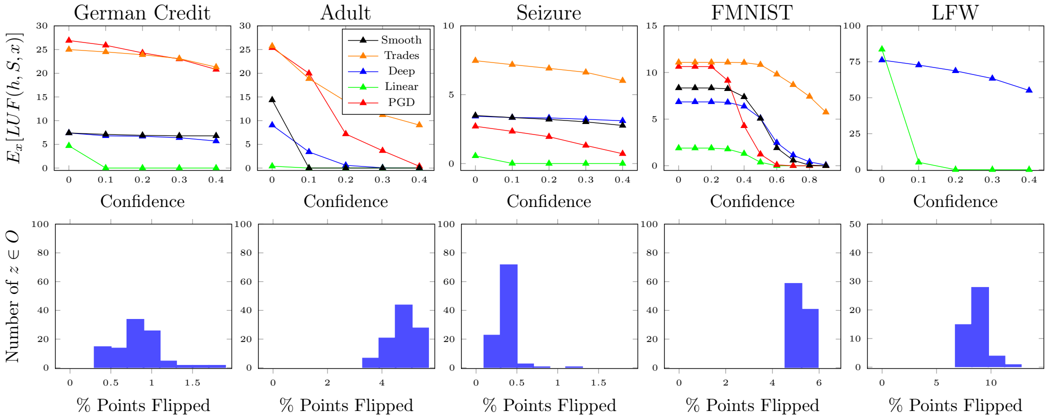


Figure 2.4: *Top row:* Prediction confidence on the horizontal axis, percentage of stable points experiencing LUF (i.e., $E_x[LUF(h,s,x)]$) on the vertical axis. For FMNIST, confidence is calculated as the absolute difference between the two most confidently predicted classes; for other datasets, confidence is $|h_S(x) - 0.5|$. Note the differences in scale between the graphs; adversarial German Credit and Adult models display especially high leave-one-out unfairness, as well as LFW. *Bottom Row*: A bar chart displaying what percentage of points in the dataset are affected by *each one* of the points taken out. Each bar shows the number of points in O (left-out points) whose absence changed the prediction of the percentage of points shown on the x axis. Notably, every single point that was taken out of the dataset affected at least one other individual's prediction. Note the difference in scale on the x axis.

Setup. For all experiments, we train models using Keras 2.4.3 with TensorFlow 2.0. In keeping with common practice, we set the random seeds used by Python, numpy, and Tensorflow. Beyond this, in order to isolate the effect of leave-one-out unfairness from other sources of instability, we use the same random initialization of model parameters across models in the same experiment, and we turn off non-determinism in GPU operations [237]. This effectively makes the learning rule h deterministic, so that when measuring LUF, the probabilities in Definition 2 are $\in \{0,1\}$. We note that, in the case of, LFW, an additional source of instability remains in the process that produces pairs of faces dynamically during training. This is necessary in order for the model to encounter a sufficiently high number of face pairs during training while being bound to memory constraints. We provide results of the same experiments over a smaller, static dataset in the supplementary material, with similar LUF behavior but lower accuracy.

As it would be prohibitively expensive to train $|S|$ models for the datasets S listed above, we instead measure differences over a fixed number of training sets obtained by randomly deriving from each dataset: a training set S, a set $O \subseteq S$ of size 100 that consists of points drawn randomly from test data (i.e. with which to create 100 different $S^{(\setminus i)}$), and a test set. We train a "baseline" deep model h_S with which to calculate the differences in prediction resulting from removing a point from O from S. For each $z_i \in O$, we train $h_{S^{(\setminus i)}}$ by removing z_i from S. For each $h_{S^{(\setminus i)}}$, we estimate LUF(h,S,x) for all x in the dataset by measuring the differences between $h_S(x)$ and $h_{S^{(\setminus i)}}(x)$, and taking the maximum difference over the sample of 100 leave-one-out points O. Since the distribution that each training set S comes from is a uniform distribution over the entire dataset, this is measuring $\mathbb{E}_x[\text{LUF}(h,S,x)]$ for each training set S and learning rule h. A step-by-step explanation of this calculation is given in the supplementary material. Due to the cost, for LFW we train 50 $h_{S^{(\setminus i)}}$ models, i.e., in this case we set $|O| = 50$.

To verify that the leave-one-out unfairness is a property of the models and not an unavoidable consequence of training a machine learning model on the presented datasets, we also train linear models on the same datasets with the same method, and compare the leave-one-out unfairness of these linear models to their deep counterparts.

The majority of our results displaying the extent of expected LUF in deep models center around the use of one architecture, seed, and set of hyper-parameters per dataset, in order to keep as many variables controlled as possible. To ensure that the behavior described is consistent, we present experiments displaying the effect of changing architecture and random seed on our main results in Figure 2.5. The main set of models for German Credit and Seizure datasets have three hidden layers, of size 128, 64, and 16. Models on the Adult dataset have one hidden layer of 200 neurons. The FMNIST model is a modified LeNet architecture [152]. This model is trained with dropout. The LFW face-matching model consists of a concatenation layer composing the two input images, a 4-layer convolutional stack, followed by a dense layer, and a Sigmoid output. German Credit, Adult, and Seizure models are trained for 100 epochs; FMNIST and LFW models are trained for 50. German Credit models are trained with a batch size of 32, FMNIST 64, and Adult, Seizure, and LFW used batch sizes of 128. German Credit, Adult, Seizure and LFW models were trained with Adam ($lr = 1.e^{-3}$), and FMNIST with SGD ($lr = 0.1$).

The experiments outlined above were also performed on models with two other architectures per dataset, in order to compare results across architecture, presented in Figure 2.5. For German Credit and Seizure datasets, one additional architecture was a shallower model of a 1-hidden layer model of size 100, and the other a narrower model of 3 hidden layers of sizes 64, 32, and 8. For the Adult dataset, the additional models were a narrower 1-hidden layer of size 100, and a deep model with the same architecture as the main German Credit models. For FMNIST, we trained a shallower model with one set of layers removed, as well as a model with no dropout. Finally, for LFW, we compare with a ResNet50 [109] model, pre-trained on ImageNet, and modified to take in two inputs and have a Sigmoid output, as well as a model whose filters are twice the size of the original model. For experiments comparing the extent of expected LUF across models seeded differently, we perform the main experiments outlined in the paragraphs above over 5 different random seeds for all tabular and time series datasets, and three different random seeds for image datasets. Further details on model construction can be found in the appendix.

LUF in Deep Models Figure 2.4 shows the prevalence of leave-one-out unfairness on all five datasets. The first row plots the percentage of individuals x experiencing LUF(h, S, x): i.e., $E_x[LUF(h, S, x)]$, ranging over the confidence of the baseline model's prediction. On every dataset examined, deep models display nontrivial expected LUF, ranging from ~4% to ~77%. The second row shows the number of points in $z_i \in O$ (out of 100) that lead to a given percentage of individuals x having their predictions changed when only z is removed from the dataset. The percentage per point on the X axis, and the number of points that change this percentage of outcomes is on the Y axis. Notably, the removal of each point sampled lead to an $h_{S\setminus i}$ model that changed the predictions of at least one other point, suggesting that leave-one-out unfairness is in fact very common.

The results show that leave-one-out unfairness cannot be reliably predicted given test accuracy, and more notably, generalization error (shown in Table 3.1). While it may seem natural that models with higher accuracies display less LUF, the deep model on the Adult dataset has an

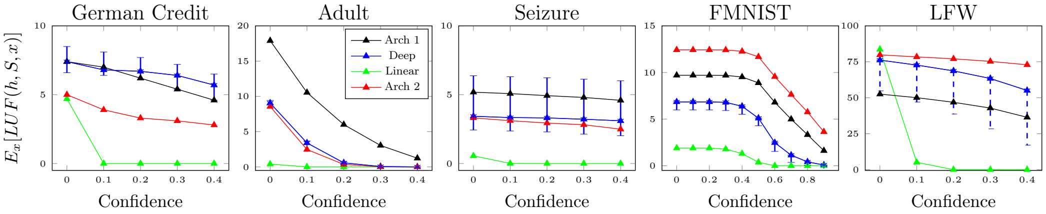


Figure 2.5: Effect of random seed and architecture on LUF results in deep models from Figure 2.4. The red and green plots show LUF for models of slightly different architecture, as described in the experimental setup, and the bars on the blue line show the minimum and maximum LUF values over 5 random seeds on the main architecture shown in main results. Notice the difference in scale across the graphs.

accuracy ~10% higher than the German Credit dataset, yet the German Credit dataset has approximately 2% fewer individuals experiencing LUF. Even more impressively, the LFW model has higher accuracy than both German Credit and Adult models, by 12% and 2% respectively, yet has a much higher expected LUF of ~77%, compared to 7% and 10%. Even for models on the same dataset, accuracy and LUF can increase together: for example, the FMNIST model has approximately an 8% accuracy difference between the linear and deep models, however, the deep model has approximately 7% of its treatment population susceptible to LUF, whereas the linear model has only 2%. In Chapter 7, we demonstrate how a more general notion of prediction instability—*predictive multiplicity*—can increase with accuracy, due to its strict ties to a model's variance.

Similarly, following intuitions from model stability, lower generalization error may naturally seem to coincide with lower levels of LUF. However, the German Credit model has a generalization error of ~25%, yet has lower LUF than both the Adult model, with generalization error of just ~3%, and the LFW model, with generalization error of ~5%. Indeed, while these results will be further discussed in the next section, it is worthy of note that the PGD model on the Adult dataset has essentially zero generalization error, yet has a very high percentage of individuals experiencing leave-one-out unfairness (~25%), while the deep model on the Adult dataset has generalization error of ~3.5% and has around 10% of individuals experiencing LUF. While we did not explicitly control for accuracy or generalization error, these results are evidence that LUF does not depend on these metrics.

Also of note is that LUF does not decrease with dataset size—FMNIST and German Credit are the largest and smallest datasets, with training set sizes of 60,000 and 800 respectively, yet FMNIST displayed similar LUF to German Credit (within 1%). The Adult dataset is also larger than German Credit (~$|S| = 15,000$) and displays more expected LUF.

Perhaps most importantly, confidently-predicted points are not immune from leave-one-out unfairness in deep models: on the majority of the datasets, a substantial portion of points with high LUF were predicted with confidence greater than 0.9 by the baseline model. This is illustrated by the fact that the curves displaying the number of points versus baseline model confidence do not drop off sharply in all models except for those on the Adult dataset. This is an interesting manifestation of miscalibration in deep models: some confident decisions may still be somewhat arbitrary, in that they are sensitive to the specific makeup of the training set.

Consistency Under Varying Conditions We provide calculations of expected LUF over all datasets in deep models where the architecture and random seed differ, in order to ensure that the results are consistent across different modeling choices.

The results are presented in Figure 2.5. While there is some variation in expected LUF, no modeling choice explored eradicates the behavior. Interestingly, certain architectures seem to exacerbate or diminish LUF: a deeper model increases LUF in the Adult dataset by nearly 10%, and removing dropout from the FMNIST model, as well as increasing the filter size on LFW, have a similar effect. This may warrant further study to find potential mitigation techniques through architecture selection, however, no pattern is immediately noticeable: for example, while a shallower model exhibited lower expected LUF on the German Credit dataset than the baseline model, the same shallow architecture exhibited more expected LUF than the baseline on the Seizure dataset, which shares the same architecture as the German Credit baseline model. Random seed also affects the prevalence of expected LUF, to a slightly lesser extent for all models but LFW. Broadly, however, the results show that LUF is not an artifact of any one particular set of training conditions.

Linear Models. We also provide the results for the same experiments on linear models to calibrate against a more stable learning rule that yields less complex models: observe the green line in Figure 2.4. These results show that LUF is not inherent to the data. While there are points that are treated leave-one-out unfairly, they are substantially fewer—with the exception of LFW, where the learning task is markedly more complex than the other datasets, and unsuitable for a linear model. Additionally, the overwhelming majority of points treated leave-one-out unfairly in linear models are not confidently predicted—in fact, in all models but FMNIST, there are no points treated leave-one-out unfairly that are predicted with a difference of more than 10% from 50% confidence.

This result agrees with intuition—linear boundaries are smooth, and linear regression is stable. If the introduction of a point does shift the boundary, it is likely that only points already close to the decision boundary (i.e., low-confidence points) are affected. Deep models can have arbitrarily complex decision boundaries, which appears to be closely-related to LUF. As the phenomenon of memorization [84, 264] suggests, and these results support, deep models have the capacity to "overreact" to the presence of individual entries in their training data. Figure A.1 illustrates this further in a low-dimensional setting. Not only can the region around the left-out point potentially change, but there are may also be far-reaching effects on the decision boundary beyond the neighborhood of the left-out point. These changes will affect not just the predicted label of new points, but also their assigned confidence score. While intuitions that are valid in low-dimensional settings do not always transfer to high dimension, this may nonetheless provide some intuition behind the factors that contribute to leave-one-out unfairness.

2.5 LUF and Robust Classification

Calls to mitigate adversarial examples [194, 236] have motivated a significant amount of research aimed at producing robust classifiers [46, 166, 254]. Recent results have shown that some of these techniques can even be repurposed to ensure individual fairness [260], and moreover, that they often produce deep models that admit more interpretable feature attributions [82, 119, 185]. Intuitively, these findings could suggest that robust prediction methods rely on “robust features” [119] that align more closely with human understanding of the problem domain, and whose presence in the model may be accordingly less dependent on individual points in the training data.

In this section, we explore this conjecture by measuring the incidence of leave-one-out unfairness with two robust classification methods: adversarial training, and randomized smoothing. We find that models trained adversarially using projected gradient descent (PGD) [166] as well as models trained with the TRADES algorithm [266] have significantly higher rates of LUF, in most cases approximately doubling the number of unstable points over standard training. On the other hand, models that are made robust by post-hoc smoothing with Gaussian noise [46] almost always have similar rates of expected LUF. Taken together, these results suggest that LUF and robustness are not inherently tied to one another, but that certain classes of models may provide beneficial properties for both, warranting further study.

Setup. We use the same experimental setup as in Section 2.4 for measuring leave-one-out unfairness. In these experiments, we only train deep models. For adversarial training, we use PGD with an ℓ_2 radius $\epsilon = 3.0$ and 10 PGD steps on FMNIST and Seizure datasets. For the Adult and German Credit datasets, we use radius $\epsilon = 1.0$. On the German Credit dataset, we use the ℓ_∞ norm. The radius remained the same between PGD and TRADES training. We determined the radius for adversarial training by finding the minimum distance (with respect to the adversarial norm) between any two points of different classes over a large sample of the dataset. If this was impossible because this distance was zero, we chose a distance smaller than that between over 99% of cross-class pairs of points in the sample. For TRADES training, we used all of the same hyperparameters as PGD training, with the addition of the TRADES parameter, which was 1 for Adult and German Credit, and 10 for Seizure and FMNIST. Notice that, for face-matching problems, the threat model for finding adversarial examples is less clear—e.g., it is not obvious if the attacker has access to individual images, or pairs of images. As we are unaware of an established threat model for face-matching, we do not evaluate LFW in this section. For randomized smoothing, we take 1,000 Gaussian samples with $\sigma^2 = 0.1$ for the Adult and Seizure datasets, 10,000 samples with $\sigma^2 = 0.05$ for FMNIST, and 2,000 samples with $\sigma^2 = 0.05$ for German Credit. While Cohen et al. [46] report needing more smoothing samples to achieve strong adversarial guarantees, our goal in these experiments is to measure LUF, which we found to be insensitive to additional samples beyond the numbers reported above. The accuracy of these models is shown in Table 3.1.

Results and Discussion. The results are shown in Figure 2.4. The most immediate trend is the degree to which PGD and TRADES adversarial training worsens LUF: approximately by a factor of two across all datasets, and by a factor of nearly three on the German Credit dataset. Seizure is a partial exception in that the PGD training does not worsen LUF, but TRADES training does. While adversarial training produces models that are more invariant to small changes in their inputs, these results show that the training procedure itself can be unstable. This may be related to prior work demonstrating that adversarially-trained models are more vulnerable to *membership inference* [232, 261], a privacy attack that exploits memorization to leak information about training data. While membership vulnerability does not necessarily imply greater LUF, these experiments show that in many cases the two phenomena may be related. We also note that these results do not necessarily contradict the

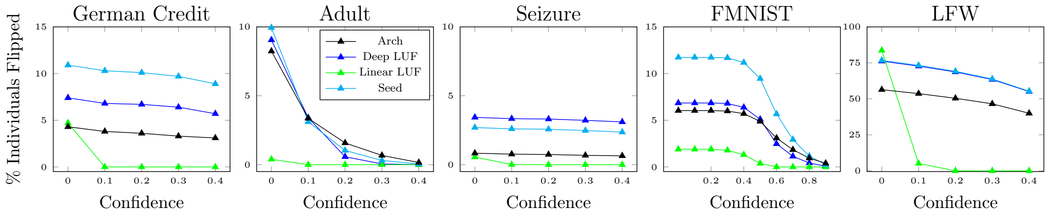


Figure 2.6: Arbitrariness in decision outcome as a result of changes in random seed, and small changes in architecture, are presented alongside expected LUF, i.e. arbitrariness from small changes in the training set. Calculation methods are described in 2.6. We present these results to motivate a wider connection between learning algorithm stability and fairness, beyond LUF. Notice the difference in scale across graphs.

"robust feature" hypothesis proposed by Ilyas et al. [119], as robust learned features need not generalize across large portions of the dataset.

Turning to the curves labeled "Smooth" in Figure 2.4, it is clear that randomized smoothing leads to qualitatively different leave-one-out unfairness results. On most datasets, smoothing had little effect ($< 1\%$ difference) on expected LUF. Beyond suggesting that leave-one-out unfairness is independent of robustness, these results also point to the fact that individual fairness and LUF are related, but separate notions. Randomized smoothing guarantees individual fairness for weighted ℓ_p metrics [260], but has a negligible effect on leave-one-out unfairness.

Looking at the geometry of these models can shed further light on the differences in results between PGD training and randomized smoothing. As suggested by Figure A.1, deep model decision boundaries have the potential to be very sensitive to individual points, and this sensitivity may affect regions of the decision boundary far beyond the local neighborhood of the point in question. This could contribute to leave-one-out unfairness, as the predictions of points in regions shifted by a training points' addition or removal will change. Adversarial training may in some cases intensify the boundaries' sensitivity to training points by penalizing inconsistent predictions in any direction within ϵ away.

Alternatively, a smoothed model returns the expected prediction over a continuous distribution centered at each point, rather than the value of the underlying model at only one point. While this does not remedy larger boundary changes stemming from instability, it likely does not exacerbate them, as evidenced by the effects in Figure 2.4.

2.6 Arbitrariness Beyond the Training Set

Our study focused on instability to changes in training data, as this type of stability is particularly well-studied due to its relevance to generalization and privacy. However, there are other potential sources of instability that may lead to arbitrary outcomes as well: for example, random initialization, batching order, and model architecture. If a difference in any of these choices results in a difference in outcome for an individual—e.g., if a change in random initialization frequently leads to a change in predicted credit risk for someone—then this too could be seen as unfair, as it would call into question the robustness of any supposed justification.

To establish a preliminary understanding of the degree to which these sources introduce changes in outcome similar to LUF, we experimentally investigate the percentage of changed outcomes resulting from varying the random seed prior to initializing and training models, as well as from the choice of model architecture. Figure 2.6 shows these results for all of the datasets studied in Section 2.4, alongside the corresponding measurements for LUF. The experimental setup largely follows that described in Section 2.4. We isolate the effect of each potential variable causing instability unfairness (architecture, random seed, and leave-one-out unfairness) in its own experiment; keeping other sources of instability controlled. For the random seed experiments, we train the same model with 100 different random seeds and calculated the effects of instability in the same manner as calculating LUF described in Section 2.4; for the experiments calculating the fairness effects of changes in architecture, we train the model on three different architectures, as described for the experiments verifying consistency in LUF in Section 2.4. Further information on the architectures considered can be found in the supplementary material.

As Figure 2.6 shows, any of these aspects in a model can affect model behavior over a substantial percentage of the overall dataset. While we focus on leave-one-out unfairness in particular for formal analysis of the intersection between model stability and fairness, all of these behaviors lead to an arbitrariness in decision procedure. In the following chapters, we consider the effects of, and remedies for broader notion of inconsistency and arbitrariness in model decision processes, beyond a leave-one-out change in the training set.

2.7 Related Work

Leave-one-out unfairness views the problem of learning instability [31, 32] from a fairness perspective. While deep learning is generally understood not to enjoy strong stability properties, our results are among the few systematic studies of the extent, and potential ramifications, of their instability. Hardt et al. show that even nonconvex models trained using Stochastic Gradient Descent remain stable over a small number of iterations, and that popular heuristics like dropout and ℓ_2 regularization help [107], and provide some experimental demonstrations. Towards achieving stability in deep learning, Kuzborskij et al. [143], develop a screening protocol for choosing random initalizations that improve stability.

Memorization, as defined by Feldman [84], is a symptom of model instability where a model predicts the correct output on a given point if it is in the training set, and incorrectly otherwise. There has been much recent work unearthing the potential for memorization in deep neural networks [264], discussion about the extent of the phenomenon in practice [11] as well as arguments for its usefulness [84]. Memorization is closely related to leave-one-out unfairness in it is a measure of stability, and crucially, focuses on how instability affects a given point, rather than an average. However, leave-one-out fairness is much broader than memorization. Memorization quantifies how much removing a given point from the training set affects that whether that particular point is predicted correctly. Leave-one-out fairness quantifies how the consistency, not the error, of a given point's prediction is affected by *any other point.*

A well-known meeting point of stability and privacy is differential privacy [70], which quantifies privacy risk in terms of a uniform, information-theoretic notion of stability. Leave-one-out fairness is related to, but weaker than, differential privacy, as shown in Section 2.3. Instability also worsens concrete privacy attacks: oversensitivity to the training set can affect a model's parameters, which can be leveraged to perform membership inference [154, 231, 259]. Our experiments in Section 2.5 may suggest that this phenomenon has a connection to leave-one-out unfairness, in that adversarial training increases both LUF and the potential for membership inference attacks [232, 261].

There is little work that connects *fairness* and stability. Notably, concurrent work by Marx et al. [168] also draws attention to the problem of instability, particularly in linear models with different regularization parameters, but does not center the definition of instability, i.e. *predictive multiplicity*, on fairness through the experience of a given individual. Leave-one-out fairness is an individual-based fairness notion. While there are several definitions of "individualized" fairness [71, 72, 127, 142], they are rarely operationalized in common fairness testing platforms, as they can be difficult to calculate. In addition to already-noted differences from prior notions of fairness, expected LUF can be effectively measured on real datasets to give insight into whether an individual may be subject to unfair treatment at inference time.

Chapter 3

Reducing Inconsistency with *Selective Ensembles*

We begin this section by demonstrating that not only are the predictions of related deep models often dissimilar, but their *feature attributions* [155, 227, 234] are as well (Section 3.2). In particular, we show that there is little connection between a model's gradients, which are the basis for many deep attribution methods, and the labels that it predicts—models with identical predictions can have arbitrarily different gradients almost everywhere (Theorem B.1). In practice, we show that this result occurs often on common datasets across closely-related models, leading to significant variation in attributions. This may be undesirable, as feature attributions are commonly used to provide explanations [155, 227, 234], debug model behavior [5], and diagnose problems related to privacy and fairness [56, 154]. Beyond these pragmatic concerns, this suggests that the salient factors behind these models' predictions on many points may have little in common, even when models appear to do comparably well on test data.

To address inconsistency in both prediction and attribution, we then turn to ensembling, a well-known approach for reducing predictive variance [91, 104, 141, 157, 173, 183]. We introduce *selective ensembles*, which leverage a recent result on multinomial rank verification [116]—which has also been used recently for making certifiably-robust predictions [46]—to efficiently mitigate the problem of inconsistency with a probabilistic guarantee. Given a point to classify, a selective ensemble returns the mode of the class labels predicted on that point, where the mode is sampled over models that vary according to a specified source of randomness in the training process. Importantly, if the mode cannot be inferred with sufficient confidence, then the selective ensemble *abstains* from prediction. This allows us to bound the probability that these ensembles do not return the true mode prediction (Theorem 3.2), and by extension, the rate of disagreement between selective ensembles (Corollary 3.2.2). In addition, we show that this also bounds the variance component in the ensembles' bias-variance error decomposition [62] (Corollary 3.2.1), providing guidance on how to effectively use them in practice.

Our experiments show that on seven benchmark datasets, selective ensembles of just ten models either *agree on the entire test data* across random differences in how their constituent models are trained, or abstain at reasonably low rates (1-5% in most cases; Section 3.4.1). Additionally, we show that simple ensembling doubles the agreement of attributions on key metrics on average, and when the variance of the constituent models is high that selective ensembling further enhances this effect (Section 3.2).

In summary, our contributions are: *(1)* we show that beyond predictions, feature attributions are not consistent across seemingly inconsequential random choices during learning (Section 3.2); *(2)* we introduce *selective ensembling*, a learning method that *guarantees* bounded inconsistency in predictions, (Section 3.3); and *(3)* we demonstrate the effectiveness of this approach on seven datasets, showing that selective ensembles consistently predict *all* points across models trained with different random seeds or leave-one-out differences in their training data, while also achieving low abstention rates and higher feature attribution consistency.

3.1 Notation and Preliminaries

Notation and Preliminaries We assume a supervised classification setting, with data points $(x, y) \in \mathbf{X} \times \mathbf{Y}$, drawn from data distribution, $\mathcal{D}$, where x represents a vector of features and y a response. In order to capture the effects of arbitrary random events on a learned model—ranging from randomness during training to randomness in the data selection process—we generalize the standard concept of a *learning rule* to that of a *learning pipeline*. Specifically, a learning pipeline, $\mathcal{P}$, is a procedure that outputs a model, $h : \mathbf{X} \to \mathbf{Y}$, taking as input a random state, $S \sim \mathcal{S}$, containing all the information necessary for $\mathcal{P}$ to produce a model (including the architecture, training set, random coin flips used by the learning rule, etc.). Intuitively, $\mathcal{S}$ represents a distribution over random events that might impact the learned model. For example, $\mathcal{S}$ might capture randomness in sampling of the training set, or nondeterminism in the optimization process, e.g., the initialization of parameters, the order in which batches are processed, or the effects of dropout.

In our experiments, we model $\mathcal{S}$ to capture two specific types of random choices, namely *(1)* the initial parameters of the model, and *(2)* leave-one-out changes to the training data. As the initial parameters of the model tend to be determined by a random seed, we will interchangeably refer to this as the selection of random seed. More generally, both of these types of choices instantiate a broader class of choices that could be considered *arbitrary*, despite the fact that they may impact the predictions [24, 168, 172] (Section 3.4.1) and explanations (Section 3.2) of the resulting model.

3.2 Instability of Feature Attributions in Deep Models

Before we consider mitigating predictive inconsistency with ensembling, we first demonstrate that models' inconsistency across random choices in training is exhibited not only through its predictions, but through its *feature attributions* as well. Feature attributions refer to numeric scores generated for some set of a model's features—most commonly the model's input features—which are meant to connote how important each feature is in generating the model's prediction. Feature attributions are commonly used as a tool for explaining model behavior [5, 155, 227, 234] localized to given set of inputs. Thus, inconsistent feature attributions between models suggest the models differ in the *process* by which they arrive at their predictions, even if the predictions are the same.

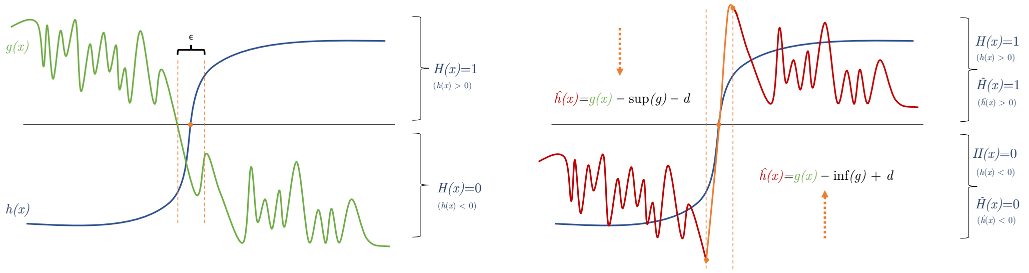


Figure 3.1: Intuitive illustration of how two models which predict identical classification labels can have arbitrary gradients. To show this, given a binary classifier H and an arbitrary function g, we construct a classifier H' that predicts the same labels as H, yet has gradients equal to g almost everywhere. We formally state this result in Theorem B.1.

In deep models, many of the most popular attribution methods are based on the model's gradients at or around a given point [227, 234]. Accordingly, we will focus on the stability of gradients, and show via analysis and experiment that they are not stable in conventional deep models. First, we motivate our results by showing that even two deep models that predict the same labels on all points may have arbitrarily different gradients almost everywhere. Later, in our empirical evaluation (Section 3.4), we demonstrate the extent of the differences between Saliency Maps [227] (i.e., input gradients) of deep networks even when the randomness of the learning pipeline is controlled to allow only one-point differences in the training set or differences in the random seed.

Predictions with Arbitrary Gradients. We show that even deep models that predict the exact same labels on all points cannot necessarily be expected to have the same, or even similar, gradients; in fact, given a binary classification model h, we can construct a model $\hat{h}$ which predicts the same labels as h, but has arbitrarily different gradients everywhere except an arbitrarily small region around the boundary of h (Theorem B.1).

Theorem 3.1. *Let $H : \mathbf{X} \to \{-1, 1\} = \text{sign}(h)$ be a binary classifier and $g : \mathbb{R}^n \to \mathbb{R}$ be an unrelated function that is bounded from above and below, continuous, and piecewise differentiable. Then there exists another binary classifier $\hat{H} = \text{sign}(\hat{h})$ such that for any $\epsilon > 0$,*

$$\forall x \in \mathbf{X}\ . \qquad \textit{1. } \hat{H}(x) = H(x) \qquad \textit{2. } \inf_{x' : H(x') \neq H(x)} \Big\{ ||x - x'|| \Big\} > {}^{\epsilon}\!/2 \implies \nabla\hat{h}(x) = \nabla g(x)$$

The proof of Theorem B.1 is given in Appendix B.0.1. The proof is by construction of $\hat{h}$; a sketch giving the intuition behind the construction is provided in Figure 3.1. In short, we first partition the domain into contiguous regions that are given the same label by H. We then construct $\hat{h}$ from g by adjusting g to lie above or below the origin to match the prediction behavior of h in each region. As these transformations merely shift g by a constant in each region, they do not change ∇g except near decision boundaries, where it is necessary to move across the origin.

Observations. The intuition stemming from Theorem B.1 is that a model's gradients at each point are largely disconnected from the labels it predicts on a distribution. As models that make identical predictions are likely to have similar loss on a given dataset, this theorem points to the possibility that models of similar objective quality may still have arbitrarily different gradients. In Section 3.2, we demonstrate that this outcome is not only *possible*, but that it occurs in real models—for example, on the German Credit dataset predicting credit risk, on average, individual models with similar accuracy agree on *less than two out of the five most important features* influencing their decision.

3.3 Selective Ensembling

We build on the approach of ensembling for variance reduction by showing how these differences in behavior can be bounded via *selective ensembling.* However, whereas prior work which finds that *more diversity* among the constituent networks is beneficial for reducing overall error [104, 141, 164, 192], our goal is to minimize, or at least place strict bounds on, the variance component. We show that ideas from robust classification, and in particular *randomized smoothing* [46], which stem from recent results on multinomial hypothesis testing [116], can be used to enforce such a bound.

Mode Predictor. We may view the image of the learning pipeline, $\mathcal{P}$, as a distribution over possible models induced by applying $\mathcal{P}$ to the random state, $S \sim \mathcal{S}$. The *mode prediction* on an input x, with respect to $\mathcal{S}$, is the expected label that would be predicted on x by models drawn from this distribution. More formally, we define the *mode predictor*, $g_{\mathcal{P},\mathcal{S}}$ for a pipeline, $\mathcal{P}$, and random state distribution, $\mathcal{S}$, as given by Equation 3.1.

$$g_{\mathcal{P},\mathcal{S}}(x) = \underset{y \in \mathbf{Y}}{\text{argmax}} \left\{ \underset{S \sim \mathcal{S}}{\mathbb{E}} \left[\mathbb{1}[\mathcal{P}(S \ ; \ x) = y] \right] \right\} \tag{3.1}$$

Note that while $g_{\mathcal{P},\mathcal{S}}$ is deterministic, and is therefore not sensitive to a specific state drawn from $\mathcal{S}$, it does not necessarily produce the ground truth label on all inputs—some learning pipelines may converge to a stable loss minimum that misclassifies certain points.

Approximation via Ensembling. An explicit representation of the true mode predictor is, of course, unattainable—the non-convex loss surface of deep models and the complex interactions between the learning pipeline and the distribution of random states makes the expectation in Equation 3.1 infeasible to compute analytically. However, we can approximate $g_{\mathcal{P},\mathcal{S}}(x)$ by computing the empirical mode prediction on x over a random sample of models produced by i.i.d. draws from $\mathcal{P}(S)$. But although ensembles with sufficiently many constituent models will more reliably output the mode prediction, for any fixed-size ensemble there will remain points on which the margin of the plurality vote is small enough to "flip" to runner-up in some set of nearby ensembles that differ on a subset of their constituents; in other words, these ensembles will not predict the mode prediction.

To rigorously bound the rate at which the ensemble will differ from the mode prediction, we allow the ensemble to *abstain* on points where the constituent predictions indicate a statistical toss-up between the two most likely classes. We call ensembles that may abstain in this way *selective ensembles*, borrowing the terminology from selective classification [75]. We can think of of abstention as a means of flagging unstable points on which the selective ensemble cannot accurately determine the mode prediction; whether this should be interpreted as a failed attempt at classification is an application-specific consideration.

Algorithm 1: Selective Ensemble Creation

```
def train_ensemble($\mathcal{P}$, $S \sim \mathcal{S}^n$, $n$):
    return $\{\mathcal{P}(S_i)$ for $i \in [n]\}$

def sample_ensemble($\mathcal{P}$, $\mathcal{S}$, $n$):
    $S \leftarrow$ sample_iid($\mathcal{S}^n$)
    return train_ensemble($\mathcal{P}$, $S$, $n$)
```

Algorithm 2: Selective Ensemble Prediction

```
def ensemble_predict($\hat{g}_n(\mathcal{P}, S)$, $\alpha$, $x$):
    $Y \leftarrow \sum_{h \in \hat{g}_n(\mathcal{P},S)}$ one_hot($h(x)$)
    $n_A$, $n_B \leftarrow$ top_2($Y$)
    if binom_p_value($n_A$, $n_A + n_B$, 0.5) $\leq \alpha$ then
        return argmax($Y$)
    else
        return ABSTAIN
```

Selective ensembles of n models predict according to the following procedure. First, the predictions of each of the n models in the ensemble are collected. The constituent models are derived from n i.i.d. samples of $\mathcal{P}(S)$ from $\mathcal{S}$, as described in Algorithm 1. From these predictions, we perform a two-sided statistical test to determine if the mode prediction was selected by a statistically significant majority of the constituent models. If the statistical test succeeds, we return the empirical mode prediction; otherwise we abstain from predicting. Pseudocode for this prediction procedure is given in Algorithm 2. We will denote by $\hat{g}_n(\mathcal{P}, S)$ (for $S \sim \mathcal{S}^n$) the output of `train_ensemble` in Algorithm 1, and by $\hat{g}_n(\mathcal{P}, S\ ;\ \alpha, x)$ prediction produced by `ensemble_predict` in Algorithm 2 on $\hat{g}_n(\mathcal{P}, S)$.

Because of their ability to abstain from prediction, we can prove that with probability at least $1 - \alpha$, a selective ensemble will either return the true mode prediction or abstain, where α is a chosen threshold for the statistical test to prevent prediction in the case of a toss-up. In other words, on any point on which it does not abstain, a selective ensemble will disagree with the mode predictor, $g_{\mathcal{P},\mathcal{S}}$, with probability at most α, as stated formally in Theorem 3.2.

The statement of Theorem 3.2 make use of the relation, $\overset{\texttt{ABS}}{\neq}$, where $y_1 \overset{\texttt{ABS}}{\neq} y_2$ if and only if $y_1 \neq \texttt{ABSTAIN}$ and $y_2 \neq \texttt{ABSTAIN}$ and $y_1 \neq y_2$. That is, $\overset{\texttt{ABS}}{\neq}$ captures disagreement between non-rejected predictions.

Theorem 3.2. *Let $\mathcal{P}$ be a learning pipeline, and let $\mathcal{S}$ be a distribution over random states. Further, let $g_{\mathcal{P},\mathcal{S}}$ be the mode predictor, let $\hat{g}_n(\mathcal{P}, S)$ for $S \sim \mathcal{S}^n$ be a selective ensemble, and let $\alpha \geq 0$. Then,*

$$\forall x \in \mathbf{X} \quad . \quad \mathbf{Pr}_{S \sim \mathcal{S}^n}\Big[\hat{g}_n(\mathcal{P}, S\ ;\ \alpha, x) \overset{\texttt{ABS}}{\neq} g_{\mathcal{P},\mathcal{S}}(x)\Big] \leq \alpha$$

The proof (Appendix C.1) relies on a result from Hung and Fithian [116] which bounds the probability that a set of votes does not return the true plurality outcome, and we apply it in a similar fashion to how it is used for making robust predictions in Randomized Smoothing [46].

Theorem 3.2 states that the probability that a selective ensemble makes a prediction that does not match the mode prediction is small. However, one possible means of ensuring this is by not providing a prediction in the first place, i.e., if the selective ensemble abstains. Thus, the *abstention rate* is necessary to quantify the fraction of points on which the mode prediction will actually be produced.

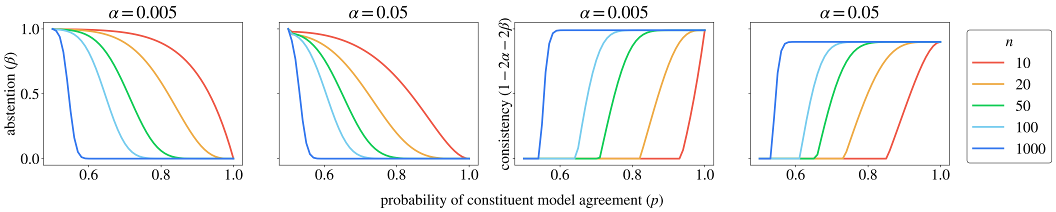


Figure 3.2: The left two plots show abstention rates as a function of the underlying probability of agreement among models over $\mathcal{S}$, i.e., the probability that any given model will return the mode prediction, with plots denoting varying numbers of constituent models. The right two graphs demonstrate the relationship between consistency of the ensemble models as given by Corollary 3.2.2.

In the 0-1 loss bias-variance decomposition of Domingos [62], the variance component of a classifier's loss is defined as the expected loss relative to the mode prediction (in our case, taken over the randomness in $\mathcal{S}$). Thus, Theorem 3.2 leads to a direct bound on this component, assuming a bound, β, on the abstention rate. This is formalized in Corollary 3.2.1.

Corollary 3.2.1. *Let $\mathcal{P}$ be a learning pipeline, and let $\mathcal{S}$ be a distribution over random states. Further, let $g_{\mathcal{P},\mathcal{S}}$ be the mode predictor, let $\hat{g}_n(\mathcal{P}, S)$ for $S \sim \mathcal{S}^n$ be a selective ensemble. Finally, let $\alpha \geq 0$, and let $\beta \geq 0$ be an upper bound on the expected abstention rate of $\hat{g}_n(\mathcal{P}, S)$. Then, the expected* loss variance*, $V(x)$, over inputs, x, is bounded by $\alpha + \beta$. That is,*

$$\mathop{\mathbb{E}}_{x \sim \mathcal{D}} \Big[V(x) \Big] = \mathop{\mathbb{E}}_{x \sim \mathcal{D}} \left[\Pr_{S \sim \mathcal{S}^n} \Big[\hat{g}_n(\mathcal{P}, S \ ; \ x) \neq g_{\mathcal{P},\mathcal{S}}(x) \Big] \right] \leq \alpha + \beta$$

Consistency of Selective Ensembles. Using the result from Theorem 3.2, we can also address the original problem raised: that deep models often disagree on their predictions due to arbitrary random events over the training pipeline. We show that, given a bound, β, on the abstention rate, the probability that two selective ensembles disagree in their predictions is bounded by $2(\alpha + \beta)$ (Corollary 3.2.2). Intuitively, this suggests that the predictions of selective ensembles are more stable over different instantiations of the random decisions captured by $\mathcal{S}$ compared to individual models.

Corollary 3.2.2. *Let $\mathcal{P}$ be a learning pipeline, and let $\mathcal{S}$ be a distribution over random states. Further, let $\hat{g}_n(\mathcal{P}, S)$ for $S \sim \mathcal{S}^n$ be a selective ensemble. Finally, let $\alpha \geq 0$, and let $\beta \geq 0$ be an upper bound on the expected abstention rate of $\hat{g}_n(\mathcal{P}, S)$. Then,*

$$\mathop{\mathbb{E}}_{x \sim \mathcal{D}} \left[\mathbf{Pr}_{S^1, S^2 \sim \mathcal{S}^n} \Big[\hat{g}_n(\mathcal{P}, S^1 \ ; \ \alpha, x) \neq \hat{g}_n(\mathcal{P}, S^2 \ ; \ \alpha, x) \Big] \right] \leq 2(\alpha + \beta)$$

Corollary 3.2.2 tells us that the agreement between any two selective ensembles is at least $1 - 2(\alpha + \beta)$. For a fixed n, decreasing α will lead to a higher abstention rate. Thus in order for α *and* β to both be small, as would be necessary for a high fraction of consistently-predicted points, we may require a large number of constituent models, n. Figure 3.2 illustrates the trade-off between α, β, and n, depending on the base level of agreement of the constituent models. In Section 3.4, we show empirically that even with small values of n, abstention rates of selective ensembles are reasonably low in practice.

	mean accuracy ± standard deviation						
Randomness	Ger. Credit	Adult	Seizure	Warfarin	Tai. Credit	FMNIST	Colon
RS	$.730 \pm .020$	$.842 \pm 1e-3$	$.973 \pm 2e-3$	$.686 \pm 3e-3$	$.820 \pm 1e-3$	$.916 \pm 3e-3$	$.927 \pm 2e-3$
LOO	$.729 \pm .012$	$.843 \pm 7e-4$	$.976 \pm 2e-3$	$.686 \pm 2e-3$	$.820 \pm 1e-3$	$.917 \pm 8e-4$	$.926 \pm 3e-3$

Table 3.1: Mean accuracy over 500 models trained over changes to random initialization and leave-one-out differences in training data. German Credit stands as an outlier due to its small sample size ($|D| = 800$).

		mean of portion of test data with $p_{flip} > 0$						
Randomness	n	Ger. Credit	Adult	Seizure	Tai. Credit	Warfarin	FMNIST	Colon
RS	1	.570	.087	.060	.082	.098	.061	.037
RS	(5, 10, 15, 20)	0.0	0.0	0.0	0.0	0.0	0.0	0.0
LOO	1	.262	.063	.031	.031	.033	.034	.042
LOO	(5, 10, 15, 20)	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 3.2: Percentage of points with disagreement between at least one pair of models ($p_{\text{flip}} > 0$) trained with different random seeds (RS) or leave-one-out differences (LOO) in training data, for single models ($n = 1$) and selective ensembles ($n > 1$). Results are averaged over 10 runs of creating 24 selective ensemble models, standard deviations are in Appendix B.0.3. Selective ensemble results are together, as there is no disagreement.

In summary, selective ensembles accomplish three primary things: (1) they identify points on which the mode prediction cannot be determined, (2) they bound the fraction of points that can be inconsistently predicted, and (3) they provide a means of reliably inferring the mode prediction when the abstention rate can be kept sufficiently low.

3.4 Evaluation

In this section, we demonstrate empirically that selective ensembles reduce instability in deep model predictions far below their theoretical bounds—to *zero* inconsistent predictions in the test set over 276 pairwise comparisons of model predictions for each of tabular datasets, and 40 for image datasets. Additionally, following Theorem B.1, we show that feature attributions of individual deep models are frequently inconsistent, and that ensembling effectively mitigates this problem.

Setup. To evaluate selective ensembling, we focus on two sources of randomness in the learning rule: *(1)* random initialization, and *(2)* leave-one-out changes to the training set. Our experiments consider seven datasets: UCI German Credit, Adult, Taiwanese Credit Default, Seizure, all from Dua and Karra Taniskidou [67]; the IWPC Warfarin Dosing Recommendation [120], Fashion MNIST [255], and Colorectal Histology [132]. All of these datasets are either related to finance, credit approval, or medical diagnosis, except for FMNIST, which we include as it is a common benchmark for image classification. Further details are in Appendix B.0.2.

All experiments are implemented in TensorFlow 2.3. For each tabular, we train 500 models from independent samples of the relevant source of randomness (e.g. leave-one-out data variations or random seeds), and for each image dataset, we train 200 models from independent samples of each source of randomness. Details about the model architecture and hyperparameters used are given in Appendix B.0.2. Table 3.1 reports the mean accuracy for each dataset, along with the standard deviation.

		accuracy (abstain as error) / abstention rate / non-selective accuracy						
$\mathcal{S}$	n	Ger. Credit	Adult	Seizure	Warafin	Tai. Credit	FMNIST	Colon
RS	5	0.0/1.0 /.745	0.0/1.0 /.842	0.0/1.0 /.975	0.0/1.0 /.688	0.0/1.0 /.822	0.0/1.0 /.919	0.0/1.0 /.927
RS	10	.576/.291/.746	.820/.043/.843	.960/.026/.975	.660/.050/.688	.800/.039/.822	.888/.059/.920	.914/.032/.928
RS	15	.636/.205/.750	.827/.032/.842	.965/.018/.975	.668/.037/.688	.807/.028/.822	.897/.042/.920	.919/.023/.928
RS	20	.664/.165/.747	.830/.024/.842	.967/.014/.975	.670/.031/.688	.810/.023/.822	.902/.036/.920	.921/.019/.938
LOO	5	0.0/1.0 /.728	0.0/1.0 /.844	0.0/1.0 /.978	0.0/1.0 /.685	0.0/1.0 /.821	0.0/1.0 /.918	0.0/1.0 /.927
LOO	10	.653/.151/.728	.827/.032/.844	.962/.027/.978	.677/.018/.685	.812/.017/.821	.909/.020/.918	.912/.036/.927
LOO	15	.678/.105/.733	.832/.012/.844	.968/.019/.979	.679/.013/.685	.814/.013/.821	.910/.016/.917	.916/.027/.927
LOO	20	.689/.079/.730	.834/.018/.843	.970/.015/.979	.680/.011/.685	.815/.010/.821	.912/.012/.918	.919/.023/.927

Table 3.3: Accuracy and abstention rate of selective ensembles, along with the accuracy of non-selective (traditional ensembles) with $n \in \{5, 10, 15, 20\}$ constituents. Results are averaged over 24 randomly selected models; standard deviations are given in Table B.4 in Appendix B.0.3

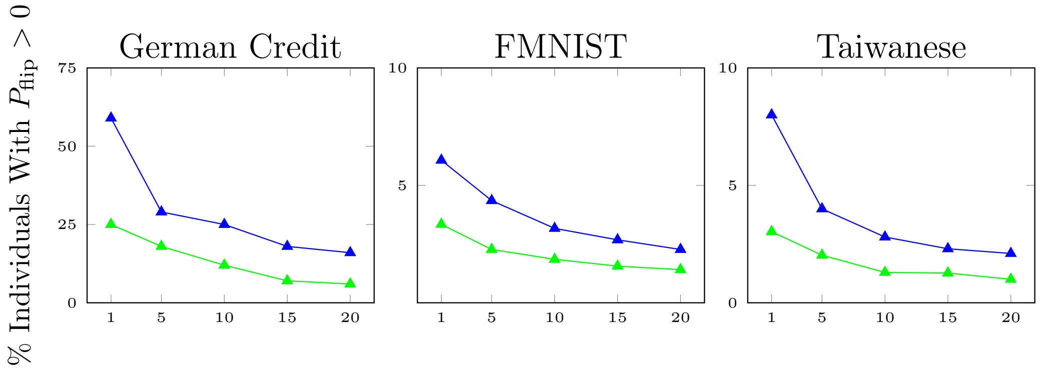


Figure 3.3: Percentage of test data with non-zero disagreement rate in normal (i.e., not selective) ensembles. Horizontal axis depicts ensemble size. While ensembling alone mitigates much prediction instability, it is unable to eliminate it as selective ensembles do.

For each non-image dataset we generate 24 random ensembles of size $n \in \{5, 10, 15, 20\}$ by selecting uniformly without replacement among the 500 pre-trained models, as well 24 "singleton" models drawn uniformly from the 500 to use as a point of comparison when measuring the stability of each ensemble. For image datasets, we generate 10 random ensembles of each size among 200 pre-trained models. We report ensemble predictions in the main document using $\alpha = 0.05$.

3.4.1 Selective Ensembles: Prediction Stability and Accuracy

To measure prediction instability over either selective ensembles or singleton models, we compare the predictions of each pair of models on each point in the test set, amounting to 276 comparisons for tabular datasets, and 40 comparisons for image datasets, in total for each point, and record the rate of disagreement, p_{flip}, across these comparisons. We report mean and variance of this disagreement over 10 random re-samplings of constituent models to create ensemble models.

The results in Table B.1 and Figure 3.3 show the percentage of points with disagreement rate greater than zero. We see that for singleton models, as many as 57% of test points have $p_{\text{flip}} > 0$, indicating that disagreement in prediction is in some cases the norm rather than the exception, although more commonly this occurs on 5-10% of the data. *Notably,*

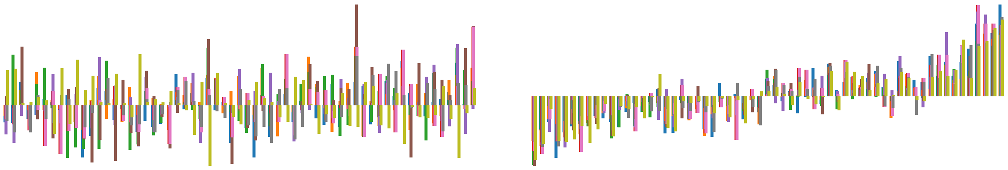


Figure 3.4: Inconsistency of attributions on the same point across an individual (left) and ensembled (right) model ($n = 15$). The height of each bar on the horizontal axis represents the attribution score of a distinct feature, and each color represents a different model. Features are ordered according to the attribution scores of one randomly-selected model.

selective ensembles completely mitigate this effect: even when *as few as ten* models are included in the ensemble, *no* points experienced $p_{\text{flip}} > 0$. Combined with the fact that abstention rates remain low (1-5%) in all cases except where p_{flip} was originally very high (e.g., German Credit), this shows that selective ensembling can be a practical method for mitigating prediction instability.

Table 3.3 shows the accuracy of selective ensembles, with abstention counted towards error, as well as accuracy of non-selective ensembles for comparison. Notably, in all six models, with the exception of German Credit, the abstention rate drops to below 4% with 20 models in the ensemble. Accordingly, the accuracy of the selective ensembles in these cases is comparable—typically within a few points—to that of the traditional ensemble. However, with just five models in the ensemble, the abstention rate is 100%; to achieve reasonable predictions with very few models, the threshold α needs to be increased accordingly. Disagreement of non-selective ensembles are pictured in Figure 3.3 (with exact numbers in Appendix B.0.3): while they do lower prediction inconsistency, they are unable to eliminate it as selective ensembles do.

3.4.2 Attribution Stability

Following up on the theoretical result given in Theorem B.1, we demonstrate that feature attributions, which are usually computed for deep models using gradients [155, 226, 234], are often inconsistent between similar models. We then show that, just as ensembling increases prediction stability, it also mitigates gradient instability, leading to more consistent attributions across models. For these experiments, we computed attributions using saliency maps [226], which are simply the gradient of the model's prediction with respect to its input, as a simple and widely-used representative of gradient-based attribution methods.

Metrics. Following previous work [61, 96], we measure the similarity between attributions using Spearman's Ranking Correlation (ρ) and the top-k intersection, with $k = 5$. For image datasets, we also display the Structural similarity metric (SSIM), discussed further in Appendix B.0.3. Spearman's ρ is a natural choice of metric as attributions induce an order of importance among features. We note that the top-k intersection is especially interesting in tabular datasets, as often only the most important features are of explanatory interest. To stay consistent with prior work, we also include Pearson's Correlation Coefficient (r). Note that r and ρ vary from -1 to 1, denoting negative, zero, and positive correlation. We compute these metrics over 276 pairwise comparisons of attributions for each size of ensemble (1, 5, 10, 15, and 20) for tabular datasets, and 40 pairwise comparisons for image datasets. For the top-k metric, we report the mean size of the intersection between each pair of attributions. More details are in Appendix B.0.2.

Baselines. To contextualize the difference of attributions across models trained from distinct randomness, we also include the attribution similarity between 24 randomly chosen points in the *same* model (Table 3.4). We also present a visual comparison of model attributions, for which we simply plot the attribution for the predicted class for a given point from nine randomly selected models out of the 24, and present the feature attributions in order of their magnitude according to another randomly selected model (Figure 3.4).

Singleton Models The left image in Figure 3.4 demonstrates the inconsistency of model attributions of individual German Credit models on a random point in the test set. Each bar on the x-axis represents the attributions for a feature, and each different-colored bar represents a different randomly selected model. Thus, the disagreement between the sizes of the bars of different colors shows the disagreement between models on which features should be deemed important. Notably, some of the bars on the graph depicting individual models even have different signs, which means that models disagree on whether that feature counts towards or against the same prediction. Similar graphs for all other datasets are included in Appendix B.0.4.

We demonstrate this inconsistency further in Table 3.4. We see that German Credit and Seizure models have particularly unstable attributions, as the top-k (and to a lesser extent, ρ and r) scores of attributions of varying points in both the *same* model, and *varying models on the same point*, are quite similar. Feature attributions of individual models are inconsistent even on highly weighted features: e.g., German Credit dataset has a top-k intersection of just over one attribute on average—suggesting that attributions generated through saliency maps on these sets of models may vary substantially over benign retrainings. Even on models where the metrics are higher, e.g. Taiwanese Credit, the baseline similarity between attributions is higher as well—thus, we see that attributions between models of the same point are usually only 2-3× more related than those of *random points within the same model*.

This instability suggests that salient variables used to inform predictions across models are sensitive to random choices made during training. As previous work has argued in similar contexts [53], this may be a result of a deep model's under-constrained search space with many local optima equivalent with respect to loss, with several minima corresponding to distinct *rationales* for making predictions.

Ensemble Models. We demonstrate that the similarity between saliency maps of ensembled models is greater than that of individual models, and that this similarity increases linearly with the number of models in the ensemble. For these experiments, we average each model's attributions toward the *majority* predicted class of the ensemble. On the right side of Figure 3.4, we see the feature attributions of various ensemble models of size 15 over the German Credit dataset. Note how the attributions of ensemble models are much more consistent than on the individual model.

We show this phenomenon more broadly in Figure 3.5, where we display graphs of average Spearman's Rank Coefficient (ρ) (y-axis) between saliency maps on a point in the test set. We see ρ increase as we increase the number of models in the ensemble (x-axis), for models generated over different random initializations and one-point differences in the training set.

Dataset	Random Seed				Leave-one-out			
	Top-5	ρ	r	SSIM	Top-5	ρ	r	SSIM
German Credit	0.20, .27	0.01, 0.25	0.02, 0.28	–	0.20, 0.49	0.01, 0.59	0.02, 0.60	–
Adult	0.46, 0.83	0.09, 0.83	0.07, 0.93	–	0.46, 0.89	0.15,0.91	0.14, 0.95	–
Seizure	0.14, 0.12	0.29, 0.32	0.30, 0.33	–	0.09, 0.25	0.23, 0.57	0.24, 0.59	–
Warfarin	0.37, 0.67	0.15, 0.72	0.12, 0.73	–	0.36, 0.92	0.12, 0.96	0.11, 0.97	–
Taiwanese Credit	0.55, 0.76	0.35, 0.75	0.36,0.83	–	0.56,0.91	0.35,0.95	0.37,0.96	–
FMNIST	0.00, 0.26	–	0.61, 0.61	0.50, 0.25	0.00, 0.57	–	0.90, 0.89	0.78, 0.62
Colon	0.00, 0.63	–	0.00, 0.92	0.18, 0.82	0.00, 0.61	–	0.00, 0.91	0.18,0.81

Table 3.4: Average top-5 intersection, Spearman's Rank Correlation Coefficient (ρ), and Pearson's Correlation Coefficient (r) to demonstrate attribution inconsistency on the *same* test points across *different* models. As a baseline, we compare against differences observed on *different* points in the *same* model. The baseline numbers are presented as: similarity baseline, similarity across models. For image models, we also report the Structural Similarity Index (SSIM). Standard deviations are included in Appendix B.0.4.

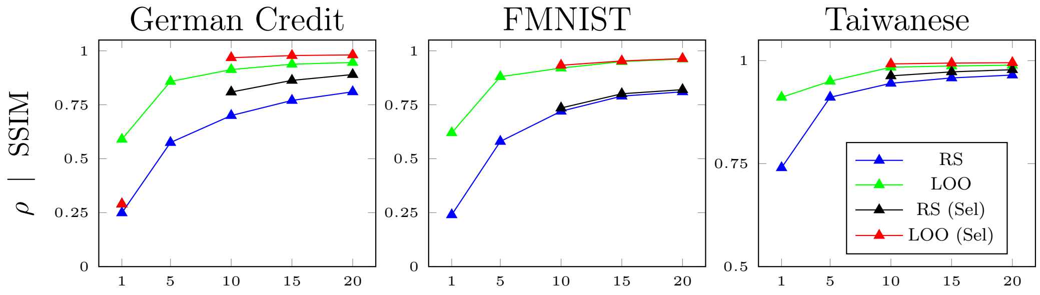


Figure 3.5: Average Spearman's Ranking coefficient, ρ, (For FMNIST, SSIM) between feature attributions for saliency maps generated for each individual test point (y-axis) over number of ensemble models (x-axis). The lines indicated with (Sel) in the legend are the same metrics for selective ensembles.

Selective ensembles can further increase stability of explanations by abstaining from unstable points, and this has a marked effect when the abstention rate is high (e.g. German Credit). Similar graphs for the rest of metrics calculated are presented in Appendix B.0.3.

3.5 Related Work

Prior work has shown that deep models are inconsistent in their predictions across arbitrary random changes in their training pipeline, such as initialization parameters and makeup of the training set [24, 53, 84, 140, 172]. The problem of model sensitivity, particularly to variability in the training set, can lead to an increase in generalization error [76] as well as to leaking training set information [70, 259]. Thus, stability-enhancing learning rules have received significant attention in order to bolster desirable properties, such as privacy [160, 195, 250].

One such approach is model ensembling, which has been used as a variance reduction method since the advent of statistical learning [40, 69, 89, 104, 108, 191, 203, 205, 218, 240, 242, 267]. However, to our knowledge, there is little work on providing guarantees about model disagreement using ensemble models that may *abstain* from prediction. We relate our approach to the classic bias-variance decomposition of error [62], showing that it certifiably bounds the variance component.

Selective ensembles can be seen as a way to flag points that are prone to inconsistency. Under this view, calibration and uncertainty estimation of deep model predictions [145, 193] is a related stream of work, and one could potentially use these techniques to determine when to abstain from prediction. However, preventing inconsistent predictions and abstaining from uncertain predictions are different goals: in our setting, the aim is to predict the mode across models drawn from a certain distribution, whereas calibration is measured against predicting the true label. Moreover, prior work has shown that confidence scores may not be correlated with prediction consistency across models with different random initializations [24]. Finally, while abstaining on points with low confidence scores may lead to greater consistency, it may not yield a guarantee, which this work provides.

Conformal inference [100, 158, 162], which rigorously assigns confidence to predictions in settings where the data may differ from training, is similarly related in that such a measure could be useful in identifying inconsistently predicted points. However, in this work, we aim to achieve consistent predictions across a *known* distribution of models, as prior work, as well as our results, suggest, even points conforming to past observations may still be predicted differently by different models.

In addition to inconsistent predictions, this work demonstrates how feature attributions can differ substantially between individual deep models with inconsequential differences. Prior works investigating instability of gradient-based explanation techniques focus on an *adversarial* context [61, 96, 111**?**]. For example, Anders et al. [9] develop attacks to create similar models that have differing gradient-based explanations. Contrastingly, this work focuses on the instability of counterfactual explanations between similar models that may occur naturally. As we demonstrate in Section 3.2, model gradients can be quite dissimilar without any adversary.

3.6 Summary

In this chapter, we show that similar deep models can have not only inconsistent predictions, but substantially different gradients as well. We introduce *selective ensembles* to mitigate this problem by bounding a model's inconsistency over random choices made during training. Empirically, we show that selective ensembles predict *all* points consistently over all datasets we studied. Selective ensembling may present a more reliable way of using deep models in settings where high model complexity *and* stability are required.

Chapter 4

Instability in Counterfactual Explanations: Problems and Solutions

In this chapter, we show that learning rule instability also impacts *counterfactual examples*, which are often used as model explanations in order to provide instructions for recourse. A counterfactual example for a given input is a nearby point in a model's input space which receives a different, often preferred, outcome. In this work, we find that counterfactual examples may not return the same prediction across small changes to their training environment—i.e. examples which promised a positive outcome may illicit different, less desirable outcomes across nearby models. This behavior may endanger recourse in situations where a model is retrained during deployment, as changes to its training set up may occur across retrainings. To remedy this problem, we introduce *Stable Neighbor Search*, a method of generating counterfactual examples which we demonstrate to be more stable across perturbations to the learning pipeline than state-of-the-art approaches used to combat instability.

4.1 Introduction

Deep Networks are increasingly being integrated into decision-making processes which require explanations during model deployment, from medical diagnosis to credit risk analysis [4, 12, 13, 14, 59, 81, 161, 233, 249]. *Counterfactual examples* [131, 133, 149, 167, 199, 206, 224, 241, 243, 245, 247] are often put forth as a simple and intuitive method of explaining decisions in such high-stakes contexts [169, 258]. A counterfactual example for an input x is a related point x' that produces a desired outcome y' from a model. Intuitively, these explanations are intended to answer the question, "Why did point x not receive outcome y'?" either to give instructions for *recourse*, i.e. how an individual can change their behavior to get a different model outcome, or as a check to ensure a model's decision is well-justified [241]. Counterfactual examples are particularly popular in legal and business contexts, as they may offer a way to comply with regulations in the United States and Europe requiring explanations on high-stakes decisions (e.g. Fair Credit Reporting Act [188] and General

Data Protection Regulation [93]), while revealing little information about the underlying model [16, 169].

Counterfactual examples are often viewed under the assumption that the decision system on which they will be used is static: that is, the model that *creates* the explanation will be the *same* model to which, e.g. a loan applicant soliciting recourse re-applies [16]. However, during real model deployments in high-stakes situations, models are not constant through time: there are often retrainings due to small dataset updates, or fine-tunings to ensure consistent good behavior [175, 208]. Thus, in order for counterfactuals to be usable in practice, they must return the same desired outcome not only for the model that generates them, but for similar models created during deployment.

This section investigates the consistency of model predictions on counterfactual examples between deep models with seemingly inconsequential differences, i.e. random seed and one-point changes in the training set. We demonstrate that some of the most common methods generating counterfactuals in deep models either are highly inconsistent between models or very costly in terms of distance from the original input. Recent work that has investigated this problem in simpler models [200] has pointed to increasing counterfactual cost, i.e. the distance between an input point and its counterfactual, as a method of increasing consistency. We show that while higher than *minimal* cost is necessary to achieve a stable counterfactual, cost alone is not a reliable signal to guide the search for stable counterfactuals in deep models (Section 4.3).

Instead, we show that a model's Lipschitz continuity and confidence around the counterfactual is a more reliable indicator of the counterfactual's stability. Intuitively, this is due to the fact that these factors bound the extent a model's local decision boundaries will change across fine-tunings, which we prove in Section 4.3.1. Following this result, we introduce *Stable Neighbor Search* (SNS), which finds counterfactuals by searching for high-confidence points with small Lipschitz constants in the generating model (Section 4.3.1). Finally, we empirically demonstrate that SNS generates consistent counterfactuals while maintaining a low cost relative to other methods over several tabular datasets, e.g. Seizure and German Credit from UCI database [67], in Section 4.4.

In summary, our main contributions are: 1) we demonstrate that common counterfactual explanations can have low consistency across nearby *deep* models, and that cost is an insufficient signal to find consistent counterfactuals (Theorem. 4.1); 2) to navigate this cost-consistency tradeoff, we prove that counterfactual examples in a neighborhood where the network has a small local Lipschitz constant are more consistent across changes to the last layer of weights, which suggests that such points are more stable across small changes in the training environment (Theorem. 4.2) ; 3) leveraging this result, we propose SNS as a way to generate consistent counterfactual explanations (Def. 4.5); 4) we empirically demonstrate the effectiveness of SNS in generating consistent and low-cost counterfactual explanations (Table 4.1). More broadly, this section further develops a connection between the geometry of deep models and the consistency of counterfactual examples. When considered alongside related findings that focus on attribution methods, our work adds to the perspective that *good explanations require good models to begin with* [52, 61, 226, 234, 251].

4.2 Background

Notation. We begin with notation, preliminaries, and definitions. Let $F(\mathbf{x};\theta) = \arg\max_i f_i(\mathbf{x};\theta)$ be a deep network where f_i denotes the logit output for the i-th class and θ is the vector of trainable parameters. If $F(\mathbf{x};\theta) \in \{0,1\}$, there is only one logit output so we write f. Throughout this section we assume F is piece-wise linear such that all the activation functions are ReLUs. We use $||\mathbf{x}||_p$ to denote the ℓ_p norm of a vector $\mathbf{x}$ and $B_p(\mathbf{x},\epsilon) \overset{\text{def}}{=} \{\mathbf{x}'|||\mathbf{x}'-\mathbf{x}||_p \leq \epsilon, \mathbf{x}' \in \mathbb{R}^d\}$ to denote a norm-bounded ball around $\mathbf{x}$.

Counterfactual Examples. We introduce some general notation to unify the definition of a counterfactual example across various approaches with differing desiderata. In the most general sense, a counterfactual example for an input $\mathbf{x}$ is an example $\mathbf{x}_c$ that receives the different, often targeted, prediction while minimizing a user-defined *quantity of interest* (QoI) (see Def. 4.1): for example, a counterfactual explanation for a rejected loan application is a related hypothetical application that was accepted. We refer to the point $\mathbf{x}$ requiring a counterfactual example as the *origin point* or *input* interchangeably. We note that there is a different definition of "counterfactual" widely used in the causality literature, where a counterfactual is given by an intervention on a causal model that is assumed to generate data observations [201]. This is a case of overlapping terminology, and is orthogonal to this work. We do not consider causality in this work.

Definition 4.1 (Counterfactual Example). *Given a model $F(\mathbf{x})$, an input $\mathbf{x}$, a desired outcome class $c \neq F(\mathbf{x};\theta)$, and a user-defined quantity of interest q, a counterfactual example $\mathbf{x}_c$ for $\mathbf{x}$ is defined as $\mathbf{x}_c \overset{\text{def}}{=} \arg\min_{F(\mathbf{x}';\theta)=c} q(\mathbf{x}',\mathbf{x})$ where the* cost *of $\mathbf{x}_c$ is defined as $||\mathbf{x}-\mathbf{x}_c||_p$.*

The majority of counterfactual generation algorithms minimize of $q_{\text{low}}(\mathbf{x},\mathbf{x}') \overset{\text{def}}{=} ||\mathbf{x}-\mathbf{x}'||_p$, potentially along with some constraints, to encourage low-cost counterfactuals [247]. Some common variations include ensuring that counterfactuals are attainable, i.e. not changing features that cannot be changed (e.g. sex, age) due to domain constraints [148, 241], ensuring sparsity, so that fewer features are changed [54, 99], or incorporating user preferences into what features can be changed [167]. Alternatively, a somewhat distinct line of work [128, 199, 243] also adds constraint to ensure that counterfactuals come from the data manifold. Other works still integrate causal validity into counterfactual search [131], or generate multiple counterfactuals at once [179].

We focus our analysis on the first two approaches, which we denote *minimum-cost* and *data-support* counterfactuals. We make this choice as the causal and distributional assumptions used in other counterfactual generation methods referenced are specific to a given application domain, whereas our focus is on the general properties of counterfactuals across domains. Specifically, we evaluate our results on minimum-cost counterfactuals introduced by Wachter et al. [247], and data-support counterfactuals from Pawelczyk et al. [199], and Van Looveren and Klaise [243]. We give the full descriptions of these approaches in Sec. 4.4.

Counterfactual Consistency. Given two models $F(\mathbf{x};\theta_1)$ and $F(\mathbf{x};\theta_2)$, a counterfactual example $\mathbf{x}_c$ for $F(\mathbf{x};\theta_1)$ is consistent with respect to $F(\mathbf{x};\theta_2)$ means $F(\mathbf{x}_c;\theta_1) = F(\mathbf{x}_c;\theta_2)$. Following Pawelczyk et al. [200], we define the *Invalidation Rate* for counterfactuals in Def. 4.2.

Definition 4.2 (Invalidation Rate). *Suppose $\mathbf{x}_c$ is a counterfactual example for $\mathbf{x}$ found in* *a model $F(\mathbf{x};\theta)$, we define the invalidation rate* IV*$(\mathbf{x}_c, \Theta)$ of $\mathbf{x}_c$ with respect to a distribution Θ of trainable parameters as* $\text{IV}(\mathbf{x}_c, \Theta) \overset{\text{def}}{=} \mathbb{E}_{\theta' \sim \Theta}\mathbb{I}[F(\mathbf{x}_c;\theta') \neq F(\mathbf{x}_c;\theta)]$.

Throughout this section, we will call the model $F(\mathbf{x};\theta)$ that creates the counterfactual the *generating* or *base* model. Recent work has investigated the consistency of counterfactual examples across similar linear and random forest models [200]. We study the invalidation rate with respect to the distribution Θ introduced by arbitrary differences in the training environment, such as random initialization and one-point difference in the training dataset. We also assume $F(\mathbf{x};\theta')$ uses the same set of hyper-parameters as chosen for $F(\mathbf{x};\theta)$, e.g. the number of epochs, the optimizer, the learning rate scheduling, loss functions, etc.

4.3 Counterfactual Invalidation in Deep Models

As we demonstrate in more detail in Section 4.4, counterfactual invalidation is a problem in deep networks on real data: empirically, we find that counterfactuals produce inconsistent outcomes in duplicitous deep models up to 94% of the time.

Previous work investigating the problem of counterfactual invalidation [200, 211], has pointed to increasing counterfactual cost as a potential mitigation strategy. In particular, they prove that higher cost counterfactuals will lead to lower invalidation rates in linear models in expectation [211], and demonstrate their relationship in a broader class of well-calibrated models [200]. While this insight provides interesting challenge to the perspective that low cost counterfactuals should be preferred, we show that cost alone is insufficient to determine which counterfactual has a greater chance of being consistent at generation time in deep models.

The intuition that a larger distance between input and counterfactual will lead to lower invalidation rests on the assumption that the distance between a point x and a counterfactual x_c is indicative of the distance from x_c to the decision boundary, with a higher distance making x_c's prediction more stable under perturbations to that boundary. This holds well in a linear model, where there is only one boundary [211].

However, in the complex decision boundaries of deep networks, going farther away from a point across the *nearest* boundary may lead to being closer to a *different* boundary. We prove that this holds even for a one-hidden-layer network by Theorem 4.1. This observation shows that a counterfactual example that is farther from its origin point may be equally susceptible to invalidation as one closer to it. In fact, we show that the *only* models where ℓ_p cost is universally a good heuristic for distance from a decision boundary, and therefore by the reasoning above, consistency, are linear models (Lemma 1).

Theorem 4.1. *Suppose that H_1, H_2 are decision boundaries in a piecewise-linear network $F(\mathbf{x}) = sign\{w_1^\top ReLU(W_0\mathbf{x})\}$, and let $\mathbf{x}$ be an arbitrary point in its domain. If the projections of $\mathbf{x}$ onto the corresponding halfspace constraints of H_1, H_2 are on H_1 and H_2, then there exists a point $\mathbf{x}'$ such that:*

$$1)\ d(\mathbf{x}', H_2) = 0 \qquad 2)\ d(\mathbf{x}', H_2) < d(\mathbf{x}, H_2) \qquad 3)\ d(\mathbf{x}, H_1) \leq d(\mathbf{x}', H_1)$$

where $d(\mathbf{x}, H_*)$ denotes the distance between $\mathbf{x}$ and the nearest point on a boundary H_*.

Lemma 1. *Let H_1, H_2, F and $\mathbf{x}$ be defined as in Theorem 4.1. If the projections of $\mathbf{x}$ onto the corresponding halfspace constraints of H_1, H_2 are on H_1 and H_2, but there* does not *exist a point $\mathbf{x}'$ satisfying* (2) *and* (3) *from Theorem 4.1, then $H_1 = H_2$.*

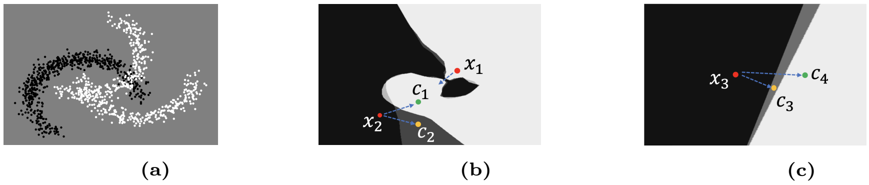


Figure 4.1: Illustration of the boundary change in a deep model (b) and a linear model (c) for a 2D dataset (a) when changing the seed for random initialization during the training. Shaded regions correspond to the area when two deep models in (b) (or two linear models in (c)) make different predictions.

Figure 4.1 illustrates the geometric intuition behind these results. The shaded regions of 4.1b correspond to two decision surfaces trained from different random seeds on the data in (a). The lighter gray region denotes where the models disagree, whereas the black and white regions denote agreement. Observe that counterfactuals equally far from a decision boundary may have different invalidation behavior, as demonstrated by the counterfactuals c_1 and c_2 for the point x_2. Also note that as shown with x_1, being far away from one boundary may lead one to cross another one in deep models. However, for two linear models shown in Fig. 4.1c, being far away from the boundary is indeed a good indicator or being consistent.

The discussion so far has demonstrated that there is not a strong theoretical relationship between cost and invalidation in deep models. In Section 4.4, we test this claim on real data, and show that higher-cost counterfactuals can have *higher* invalidation rates than their lower-cost relatives (c.f. Table 4.1). Further, we show that the coefficient of determination (R^2) between cost and invalidation rate is very small (with all but one around 0.05). Thus, while cost and invalidation are certainly related—for example, it may be necessary for a stable counterfactual to be more costly than the *minimum* point across the boundary—cost alone is not enough to determine which one will be the most consistent in deep models.

4.3.1 Towards Consistent Counterfactuals

In this section, we demonstrate that the Lipschitz continuity (Def. 4.3) of a neighborhood around a counterfactual can be leveraged to characterize the consistency of counterfactual explanations under changes to the network's parameters (Section 4.3.1). Our main supporting result is given in Theorem 4.2, which shows that a model's Lipschitz constant in a neighborhood around a x_c together with the confidence of its prediction on x_c serve as a proxy for the difficulty of invalidating x_c. We further discuss insights from these analytical results and introduce an effective approach, *Stable Neighbor Search*, to improve the consistency of counterfactual explanations (Section 4.3.1). Unless otherwise noted, this section assumes all norms are ℓ_2.

Definition 4.3 (Lipschitz Continuity)**.** *A continuous and differentiable function* $h : S \to \mathbb{R}^m$ *is* K-Lipschitz continuous *iff* $\forall \mathbf{x}' \in S, ||h(\mathbf{x}') - h(\mathbf{x})|| \leq K||\mathbf{x}' - \mathbf{x}||$. *We write h is K-Lipschitz in S.*

ReLU Decision Boundaries and Distributional Influence

We analyze the differences between models with changes such as random initialization by studying the differences that arise in their decision boundaries. In order to capture information about the decision boundaries in analytical form, we introduce *distributional* *influence*: a method of using a model's gradients to gather information on its local decision boundaries. We begin motivating this choice by reviewing key aspects of the geometry of ReLU networks.

ReLU networks have piecewise linear boundaries that are defined by the status of each ReLU neuron in the model [103, 126]. To see this, let $u_i(\mathbf{x})$ denote the pre-activation value of the neuron u_i in the network f at $\mathbf{x}$. We can associate a half-space A_i in the input space with the linear activation constraint $u_i(\mathbf{x}) \geq 0$ corresponding to the *activation status* of neuron u_i, and an *activation pattern* for a network at $\mathbf{x}$, $p(\mathbf{x})$, as the activation status of every neuron in the network. An *activation region* for a given activation pattern p, denoted $\mathcal{R}(p)$, is then a subspace of the network's input that yields the activations in p; geometrically, this is a polytope given by the convex intersection of all the half-spaces described by p, with facets corresponding to each neuron's activation constraint.

Note that for points in a given activation region $\mathcal{R}(p)$, the network f can be expressed as a linear function, i.e. $\forall \mathbf{x} \in \mathcal{R}(p).f(\mathbf{x}) = \mathbf{w}_p^\top \mathbf{x} + b_p$ where $\mathbf{w}_p$ is given by $\mathbf{w} = \partial f(\mathbf{x})/\partial \mathbf{x}$ [103, 126] Decision boundaries are thus piecewise-linear constraints, $f(\mathbf{x}) \geq 0$ for binary classifiers, or $f_i(\mathbf{x}) \geq f_j(\mathbf{x})$ between classes i and j for a categorical classifier, with linear pieces corresponding to the activation region of $\mathbf{x}$. This leads us to the following: *(1)* if a decision boundary crosses $\mathcal{R}(p)$, then $\mathbf{w}_p$ will be orthogonal to that boundary, and *(2)* if a decision boundary does not cross the region $\mathcal{R}(p)$, then $\mathbf{w}_p$ is orthogonal to an *extension* of a nearby boundary [90, 252]. In either case, the gradient with respect to the input captures information about a particular nearby decision boundary. Figure 4.2a summarizes this visually.

This analysis motivates the introduction of *distributional influence* (Definition 4.4), which aggregates the gradients of the model at points in a given *distribution of interest* (DoI) around $\mathbf{x}$.

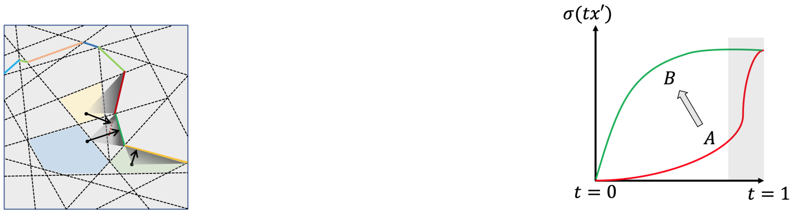
Definition 4.4 (Distributional Influence [155])**.** *Given an input* $\mathbf{x}$*, a network* $f : \mathbb{R}^d \to \mathbb{R}^m$*, a class of interest* c*, and a distribution of interest* $\mathcal{D}_\mathbf{x}$ *which describes a reference neighborhood around* $\mathbf{x}$*, define the distributional influence as* $\chi^c_{\mathcal{D}_\mathbf{x}}(\mathbf{x}) \overset{\text{def}}{=} \mathbb{E}_{\mathbf{x}' \sim \mathcal{D}_\mathbf{x}}[\partial f_c(\mathbf{x}')/\partial \mathbf{x}']$*. We write* $S(\mathcal{D}_\mathbf{x})$ *to represent the support of* $\mathcal{D}_\mathbf{x}$*. When* $m = 1$*, we write* $\chi_{\mathcal{D}_\mathbf{x}}(\mathbf{x}) \overset{\text{def}}{=} \mathbb{E}_{\mathbf{x}' \sim \mathcal{D}_\mathbf{x}}[\partial f(\mathbf{x}')/\partial \mathbf{x}']$*.*

In Leino et al. [155], distributional influence is used to attribute the importance of a model's input and internal features on observed outcomes. Following the connection between gradients and decision boundaries in ReLU networks, we leverage it to capture useful information about nearby decision boundaries as detailed in Section 4.3.1.

Consistency and Continuity

Characterizing the precise effect changes such as random initialization have on the outcome of training is challenging. We approach this by modeling the differences that arise from small changes such as a *fine-tuning* of the original model, where the top layer of the model is re-trained and the parameters of rest of the layers are frozen.

We now introduce Theorem 4.2, which bounds the change on distributional influence when the model is fine-tuned at its top layer in terms of the model's Lipschitz continuity on the support of $\mathcal{D}_\mathbf{x}$. This suggests that finding a high-confidence counterfactual example in a neighborhood with a lower Lipschitz constant may lead to lower invalidation after fine-tuning, given the relationship between nearby boundaries and influence described in the previous section.



(a)

(b)

Figure 4.2: (a) A geometric view of the input space in a ReLU network. Dashed lines correspond to activation constraints while the colorful solid lines are piece-wise linear decision boundaries. Taking gradient of the model's output with respect to the input returns a vector that is orthogonal to a nearby boundary (points in the blue and green regions) or an extension of a nearby boundary (the point in the yellow region). (b) Curves of the model's sigmoid output $\sigma(t\mathbf{x}')$ (y-axis) against interpolation parameter t (x-axis).

Theorem 4.2. *Let $f(\mathbf{x}) \overset{\text{def}}{=} \mathbf{w}^\top \cdot h(\mathbf{x}) + b$ be a ReLU network with a single logit output (i.e., a binary classifier), where $h(\mathbf{x})$ is the output of the penultimate layer, and denote $\sigma_\mathbf{w} = \sigma(f(\mathbf{x}))$ as the sigmoid output of the model at $\mathbf{x}$. Let $\mathcal{W} \overset{\text{def}}{=} \{\mathbf{w}' : ||\mathbf{w} - \mathbf{w}'|| \leq \Delta\}$ and $\chi_{\mathcal{D}_\mathbf{x}}(\mathbf{x}; \mathbf{w})$ be the distributional influence of f when weights $\mathbf{w}$are used at the top layer. If h is K-Lipschitz in the support $S(\mathcal{D}_\mathbf{x})$, the following inequality holds:*

$$\forall \mathbf{w}' \in \mathcal{W}, \quad ||\chi_{\mathcal{D}_\mathbf{x}}(\mathbf{x}; \mathbf{w}) - \chi_{\mathcal{D}_\mathbf{x}}(\mathbf{x}; \mathbf{w}')|| \leq K\sqrt{[d\sigma(\mathbf{x}; \mathbf{w})||\mathbf{w}|| + C_1]^2 + C_2}$$

where C_1 and C_2 are constants and $d\sigma(\mathbf{x}; \mathbf{w}) \overset{\text{def}}{=} \partial \sigma_\mathbf{w} / \partial f$.

Observations. Theorem 4.2 characterizes the extent to which a model's local decision boundaries, by proxy of influence, may change as a result of fine-tuning. This intuitively relates to the likelihood of a counterfactual's invalidation, as a point near a decision boundary undergoing a large shift is more likely to experience a change in prediction than one near a stable portion of the boundary. As the two key ingredients in Theorem 4.2 are the local Lipschitz constant and the model's confidence at $\mathbf{x}$, this suggests that searching for high-confidence points in neighborhoods with small Lipschitz constants will yield more consistent counterfactuals. While Theorem 4.2 does not provide a direct bound on invalidation, and is limited to changes only at the network's top layer, we characterize the effectiveness of this heuristic in more general settings empirically in Section 4.4 after showing how to efficiently operationalize it in Section 4.3.1.

Finding Consistent Counterfactuals

The results from the previous section suggest that counterfactuals with higher sigmoid output and lower Lipschitz Constants of the penultimate layer's output with respect to the DoI $\mathcal{D}_\mathbf{x}$ will be more consistent across related models. *Stable Neighbor Search* (SNS) leverages this intuition to find consistent counterfactuals by searching for those with a low Lipschitz constant and confident counterfactual. We can find such points with the objective in Equation 4.1, which assumes a given counterfactual point $\mathbf{x}$.

$$\mathbf{x}_c = \arg \max_{\mathbf{x}' \in B(\mathbf{x}, \delta)} [\sigma(\mathbf{x}') - K_{S'}] \quad \text{such that } F(\mathbf{x}_c; \theta) = F(\mathbf{x}; \theta) \tag{4.1}$$

In Eq. 4.1 above and throughout this section, we assume that F is a binary classifier with a single-logit output f, and sigmoid output $\sigma(f(\mathbf{x}))$. When f is clear from the context, we directly write $\sigma(\mathbf{x})$. The results are readily extended to multi-logit outputs by optimizing over the maximal logit at $\mathbf{x}$. $K_{S'}$ is the Lipschitz Constant of the model's sigmoid output over the support $S(\mathcal{D}_{\mathbf{x}'})$. We relax the Lipschitz constant K of the penultimate output in the Theorem 4.2 to the Lipschitz constant of the entire network, as in practice any parameter in the network, and not just the top layer, may change.

Leveraging a well-known relationship between the dual norm of the gradient and a function's Lipschitz constant [198], we can rephrase this objective as shown in Equation 4.2. Note that we assume ℓ_2 norms throughout, so the dual remains ℓ_2.

$$\mathbf{x}_c = \arg \max_{\mathbf{x}' \in B(\mathbf{x},\delta)} \left[\sigma(\mathbf{x}') - \max_{\hat{\mathbf{x}} \in S(\mathcal{D}_{\mathbf{x}'})} ||\frac{\partial \sigma(\hat{\mathbf{x}})}{\partial \hat{\mathbf{x}}}|| \right] \quad \text{such that } F(\mathbf{x}_c;\theta) = F(\mathbf{x};\theta) \tag{4.2}$$

Choice of DoI. The choice of DoI determines the neighborhood of points from which we gain an understanding of the local decision boundary [252]. Following prior work, we choose $\mathcal{D}$ as Uniform($\mathbf{0} \to \mathbf{x}$), a uniform distribution over a linear path between a zero vector and the current input [234]. That is, the set of points in $\mathcal{D}$ is $S(\mathcal{D}) \overset{\text{def}}{=} \{t\mathbf{x}, t \in [0,1]\}$. Equation 4.3 below updates the objective accordingly.

$$\mathbf{x}_c = \arg \max_{\mathbf{x}' \in B(\mathbf{x},\delta)} \left[\sigma(\mathbf{x}') - \max_{t \in [0,1]} ||\frac{\partial \sigma(t\mathbf{x}')}{\partial (t\mathbf{x}')}|| \right] \quad \text{such that } F(\mathbf{x}_c;\theta) = F(\mathbf{x};\theta) \tag{4.3}$$

While Equation (4.3) provides an objective that uses only primitives that are readily available in most neural network frameworks, solving the inner objective using gradient descent requires second-order derivatives of the network, which is computationally prohibitive. In the following, we discuss a sequence of relaxations to Eq. (4.3) that provides resource-efficient objective function.

Avoiding vacuous second-order derivatives. There exists a lower-bound of the term $\max_{t\in[0,1]} ||\partial\sigma(t\mathbf{x}')/\partial(t\mathbf{x}')||$ by utilizing the following Proposition 4.3, which allows us to relax Eq. 4.3 by maximizing a differentiable lower-bound of the gradient norm rather than the gradient norm itself.

Proposition 4.3. *Let q be a differentiable, real-valued function in $\mathbb{R}^d$ and S be the support set of Uniform($\mathbf{0} \to \mathbf{x}$). Then for $\mathbf{x}' \in S$, $||\partial q(\mathbf{x}')/\partial \mathbf{x}'|| \geq ||\mathbf{x}||^{-1} |\partial q(r\mathbf{x}')/\partial r|_{r=1}|$.*

Noting that the constant factor $||\mathbf{x}||$ is irrelevant to the desired optimization problem, Equation 4.4 below updates the objective by fitting σ into the place of q in Proposition 4.3. The absolute-value operator is omitted because the derivative of the sigmoid function is always non-negative.

$$\mathbf{x}_c = \arg \max_{\mathbf{x}' \in B(\mathbf{x},\delta)} \left[\sigma(\mathbf{x}') - \max_{t \in [0,1]} \frac{\partial \sigma(t\mathbf{x}')}{\partial t} \right] \quad \text{such that } F(\mathbf{x}_c;\theta) = F(\mathbf{x};\theta) \tag{4.4}$$

The second term in Equation 4.4, $-\max_{t\in[0,1]} \partial\sigma(t\mathbf{x}')/\partial t$, is interpreted by plotting the output score $\sigma(t\mathbf{x}')$ against the interpolation variable t as shown in Fig. 4.2b. This term encourages finding a counterfactual point $\mathbf{x}_c$ where the outputs of the model for points between the zero vector ($t = 1$) and itself ($t = 1$) form a smooth and flattened curve B in Fig. 4.2b. Therefore, by incorporating the graph interpretation of $-\max_{t\in[0,1]} \partial\sigma(t\mathbf{x}')/\partial t$ to find an solution of $\mathbf{x}_c$ that corresponds to curve B, we can instead try to increase the area under the curve of $\sigma(t\mathbf{x}')$ against t, which simplifies our objective function with replacing the inner-derivative with an integral shown in Equation 4.5.

$$\mathbf{x}_c = \arg \max_{\mathbf{x}' \in B(\mathbf{x},\delta)} \left[\sigma(\mathbf{x}') + \int_0^1 \sigma(t\mathbf{x}')dt \right] \quad \text{such that } F(\mathbf{x}_c;\theta) = F(\mathbf{x};\theta) \tag{4.5}$$

One observation of the objective defined by Equation 4.5 is that the first term $\sigma(\mathbf{x}')$ is redundant, as differentiating the second integral term already provides useful gradient information to increase $\sigma(\mathbf{x}')$. Equation 4.5 thus yields our approach, *Stable Neighbor Search.*

Definition 4.5 (Stable Neighbor Search (SNS))**.** *Given a starting counterfactual* $\mathbf{x}$ *for a network* $F(\mathbf{x})$*, its* stable neighbor $\mathbf{x}_c$ *of radius* ϵ *is the solution to the following objective:*

$$\arg \max_{\mathbf{x}' \in B(\mathbf{x},\delta)} \int_0^1 \sigma(t\mathbf{x}')dt$$

To implement Definition 4.3.1, the integral is replaced by a summation over a grid of points of a specified resolution, which controls the quality of the final approximation. The complexity of SNS is linear in the number of points in this grid.

4.4 Evaluation

In this section, we evaluate the extent of invalidation across five different counterfactual generation methods, including Stable Neighbor Search, over models trained with two sources of randomness in setup: *1)* initial weights, and *2)* leave-one-out differences in training data. Our results show that Stable Neighbor Search consistently generates counterfactuals with lower invalidation rates than all other methods, in many cases eliminating invalidation altogether on tested points. Additionally, despite not explicitly minimizing cost, SNS counterfactuals manage to maintain low cost relative to other methods that aim to minimize invalidation.

Setup

Data. Our experiments encompass several tabular classification datasets from the UCI database including: German Credit, Taiwanese Credit-Default, Seizure, and Cardiotocography (CTG). We also include FICO HELOC [85] and Warfarin Dosing [47]. All datasets have two classes except Warfarin, where we assume that the most favorable outcome (class 0) is the desired counterfactual for the other classes, and vice versa. Further details of these datasets are included in Appendix C.2.1.

Baselines. We compare SNS with the following baselines in terms of the invalidation rate. Further details about how we implement and configure these techniques are found in Appendix C.2.3. **Min-Cost ℓ_1/ℓ_2** [247]: we implement this by setting the appropriate parameters for the elastic-net loss [43] in ART [184]. **Min-Cost ϵ-PGD** [247]: We perform Projected Gradient Descent (PGD) for an increasing sequence of ϵ until a counterfactual is found. **Pawelczyk et al.** [200]: This method attempts to find counterfactual examples *on*

Invalidation Rate

Method	German Credit		Seizure		CTG		Warfarin		HELOC		Taiwanese Credit	
	LOO	RS	LOO	RS	LOO	RS	LOO	RS	LOO	RS	LOO	RS
Min. ℓ_1	0.41	0.56	-	-	0.07	0.29	0.44	0.35	0.30	0.43	0.30	0.78
+SNS	0.00	0.07	-	-	**0.00**	0.01	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.04
Min. ℓ_2	0.36	0.56	0.64	0.77	0.48	0.49	0.35	0.3	0.55	0.61	0.27	0.72
+SNS	0.00	**0.06**	**0.02**	0.13	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.04**
Min. ϵ PGD	0.28	0.61	0.94	0.94	0.04	0.09	0.10	0.12	0.04	0.11	0.04	0.24
+SNS	**0.00**	0.12	0.04	0.16	**0.00**	**0.00**	0.01	0.02	**0.00**	**0.00**	**0.00**	0.11
Looveren et al.	0.25	0.40	0.48	0.54	0.11	0.18	0.26	0.25	0.25	0.34	0.29	0.53
Pawelczyk et al.	0.20	0.35	0.16	**0.11**	**0.00**	0.06	0.02	0.01	0.05	0.15	0.02	0.20

Counterfactual Cost (ℓ_2)

Method	German Credit	Seizure	CTG	Warfarin	HELOC	Taiwanese Credit
Min. ℓ_1	1.33	-	0.17	0.50	0.24	1.56
Min. ℓ_2	4.49	8.23	0.06	0.54	0.11	2.65
Looveren et al.	5.37	8.40	0.11	1.03	0.45	2.82
Min. ϵ PGD	1.02	1.36	0.08	0.31	0.32	0.75
Min.ℓ_1 + SNS	3.40	-	0.25	0.80	1.71	3.50
Min.ℓ_2 + SNS	6.23	9.60	**0.21**	0.90	1.71	4.68
PGD + SNS	**3.03**	**3.60**	0.22	**0.50**	1.79	**2.78**
Pawelczyk et al.	7.15	13.66	1.07	2.62	**1.35**	4.24

IV - Cost Correlation

R^2	0.05	0.06	0.02	0.01	0.17	0.05

Table 4.1: The consistency of counterfactuals measured by invalidation rates. The average ℓ_2 cost of different methods are also included. Results are aggregated over 100 networks for each experiment (RS and LOO). Lower invalidation rates and cost are more desirable. For ℓ_2 cost, the best results are highlighted among three methods (separated by a line) with lower invalidation rates. If a method has significantly low success rate in generating counterfactual examples, we report '-'. In the last line, we present the R^2 correlation coefficient from a linear regression predicting invalidation percentage from cost. Small values indicate weak correlation.

the data manifold, that are therefore more resistant to invalidation, by searching the latent space of a variational autoencoder, rather than the input space. **Looveren et al.** [243]: This method minimizes an elastic loss combined with a term that encourages finding examples on the data manifold.

We note that PGD was originally proposed in the context of adversarial adversarial examples [236]. As has been noted in prior work, the problem of finding adversarial examples is mathematically identical to that of finding counterfactual examples [36, 88, 230, 247]. While solution sparsity is sometimes noted as a differentiator between the two, we note that techniques from both areas of research can be used with various ℓ_p metrics. We measure cost in terms of both ℓ_1 and ℓ_2 norms, providing ℓ_2 in the main body and ℓ_1 in Appendix C.2.6.

Implementation of SNS. SNS begins with a given counterfactual example as mentioned in Def. 4.5, which we generate with Min. ℓ_1/ℓ_2 and Min. ϵ PGD. We use the sum of 10 points to approximate the integral.

Retraining Controls. We prepare different models for the same dataset using Tensorflow 2.3.0 and all computations are done using a Titan RTX accelerator on a machine with 64 gigabytes of memory. We control the random seeds used by both numpy and Tensorflow, and enable deterministic GPU operations in Tensorflow [237]. We evaluate the invalidation rate of counterfactual examples under changes in retraining stemming from the following two sources (see Appendix C.2.4 for more details on our training setup). **Leave-One-Out (LOO):** We select a random point (without replacement) to remove from the training data. Network parameters are initialized with the same values across runs. **Random Seed (RS):**

Network parameters are initialized by incrementing the random seed across runs, while other hyperparameters and the data remain fixed.

We note that these sources of variation do not encompass the full set of sources that are relevant to counterfactual invalidation, such as fine-tuning and changes in architecture or other hyperparameters. However, they are straightforward to control, produce very similar models that nonetheless tend to invalidate counterfactuals, and they are not dependent on any deployment or data-specific considerations in the way that fine-tuning changes would be. While we hope that our results are indicative of what might be observed across other sources, exploring invalidation in more depth in particular applications is important future work.

Metrics. To benchmark the consistency of counterfactuals generated by different algorithms, we compute the mean invalidation rate (Def. 4.2) over the validation split of each dataset. To calculate the extent of correlation between cost and invalidation, as discussed in Section 4.3, we perform a linear regression (`scipy.linregress`) between the costs for each valid counterfactual, across all five methods, with its invalidation rate across both LOO and RS differences. Table 4.1 reports the resulting R^2 for each dataset.

Methodology. For each dataset, we train a "base" model and compute counterfactual examples using the five methods for each point in the validation split. For each set of experiments (LOO or RS), we train 100 additional models, and compute the invalidation rate between the base model and the 100 variants. The results are shown in Table 4.1.

4.4.1 Results

Looking at the invalidation results in Table 4.1, the most salient trend is apparent in the low invalidation rates of SNS compared to the other methods. SNS achieves the lowest invalidation rate across all datasets in both LOO and RS experiments, except for on the Seizure dataset with RS variations, where there is a two-point difference in the invalidation rate. SNS generates counterfactuals with *no* invalidation on CTG, Warfarin, and Heloc, and no invalidation over LOO differences on German Credit and Taiwanese Credit.

Notably, this is down from invalidation rates as high as 61% from other methods on Heloc, and $\approx 10 - 50\%$ on others. On Seizure, which had IV rates as high as 94% from other methods, SNS achieves just 2% (LOO) invalidation. The closest competitor is the method of Pawelczyk et al. [200], which achieves zero invalidation in one case (CTG under LOO), but at significantly greater cost –in five out of six cases, SNS produced less-costly counterfactuals, and in nearly every case the margin between the two is greater than 2×.

As discussed in Section 4.3, while increasing cost is not a reliable way to generate stable counterfactuals for deep models, our results do show that stable counterfactuals tend to be more costly. The data suggests that greater-than-minimal cost appears to be necessary for stability. While SNS counterfactuals are much less costly than those generated by Pawelczyk et. al, they are consistently more costly than other methods that aim to minimize cost without other constraints. To investigate the relationship between counterfactual cost and invalidation more closely, we report the R^2 coefficient of determination of a linear regression between the cost of each valid counterfactual generated and its invalidation rate in Table 4.1. Recall that a R^2 ranges from zero to one, with scores closer to zero indicating no linear relationship. Notably, Table 4.1 shows that the correlation between cost and invalidation is quite weak: the *maximum* R^2 over all datasets is 0.17 (Heloc), while most of the other datasets report coefficients that are much smaller–at or below 0.05.

4.5 Related Work

Counterfactual examples enjoy popularity in the research literature [54, 60, 99, 133, 167, 199, 230, 243, 245, 247, 258], especially in the wake of legislation increasing legal requirements on explanations of machine learning models [93, 129]. However, recent work has pointed to problems with counterfactual examples that could occur during deployment [16, 150, 200, 211]. For example, Barocas and Selbst [15] point to the tension between the usefulness of a counterfactual and the ability to keep the explained model private. Previous work investigating the problem of invalidation, has pointed to cost as a heuristic for evaluating counterfactual invalidation at generation time [200, 211]. We demonstrate that cost is not a reliable metric for predicting invalidation in *deep* models, and show how the Lipschitz constant and confidence of a model around a counterfactual can be a more faithful guide to finding stable counterfactual examples.

While in this work, we address the problem of multiplicitious deep models producing varying outputs on *counterfactual examples*, recent work has shown that there are large differences in model prediction behavior on *any* input across small changes to the model [24, 53, 168]. Instability has also been shown to be a problem for gradient-based explanations, although this is largely studied in an adversarial context [61, 96, 111].

Within the related field of adversarial examples, there is a recent interest in *adversarial transferability* [64, 119, 256], where adversarial attacks are induced to transfer between models. In general, adversarial transferability concerns transferring attacks between extremely different models—e.g., trained on disjoint training sets. Meanwhile, in this work, we decrease counterfactual invalidation between very *similar* models, in order to preserve recourse and explanation consistency. Interestingly, Goodfellow et al. [97] suggest that transferability of adversarial examples is due to local linearity in deep networks. This supports our motivation: we find stable counterfactuals in more Lipschitz regions of the model, i.e. where it behaves (approximately) linearly. We note, however, that as linearity does not imply Lipschitzness, this insight does not provide a clear path to generating stable counterfactuals. We look forward to exploring the potential overlap between these two areas as future work.

4.6 Conclusion

In this chapter, we characterize the consistency of counterfactual examples in deep models, and demonstrate that counterfactual cost and consistency are not strongly correlated. To mitigate the problem of counterfactual inconsistency, we introduce *Stable Neighbor Search*, which finds stable counterfactuals by leveraging the connection between the Lipschitz constant and confidence of the network around a counterfactual, and its consistency. At a high level, our work adds to the growing perspective in the field of explainability that creating good explanations requires good models to begin with.

Part II

Case Studies in Fairness Interventions along the Modeling Pipeline

Chapter 5

Case Study: Machine Learning Pipeline Fairness Interventions in IRS Tax Audit Selection

In this chapter, we present a case study of using pipeline-based fairness interventions in a public policy application: the allocation of tax audits by the Internal Revenue Service (IRS). While the field of algorithmic fairness has developed primarily around notions of treating like individuals alike, we instead explore the concept of *vertical equity*—appropriately accounting for relevant differences across individuals—which is a central component of fairness in many public policy settings. Applied to the design of the U.S. individual income tax system, vertical equity relates to the fair allocation of tax and enforcement burdens across taxpayers of different income levels. Through a unique collaboration with the IRS, we use access to detailed, anonymized individual taxpayer microdata, risk-selected audits, and random audits from 2010-14 to study vertical equity in tax administration. In particular, we assess how the adoption of modern machine learning methods for selecting taxpayer audits may affect vertical equity. This work makes four contributions.

First, we show how the adoption of more flexible machine learning (classification) methods—as opposed to simpler models —shapes vertical equity by shifting audit burdens from high to middle-income taxpayers.

Second, given concerns about high audit rates of low-income taxpayers, we investigate how existing algorithmic fairness techniques would change the audit distribution. We find that such methods can mitigate some disparities across income buckets, but that these come at a steep cost to performance. This result points to the imperfect fit many common conceptions of fairness, and fairness mitigation techniques, have with certain public policy problems. Firstly, common fairness definitions may not align exactly with contextual fairness desiderata. Beyond this, bias mitigation techniques aimed at enforcing these definitions on machine learning systems often are not built to take into account the realities of many public policy problems: for example, the existence of an agency *budget.* The differences in problem setup between common ML solutions and the realities of some of the applications where ML systems are used can lead to bias mitigation techniques not performing as expected, which we explore in Section 5.5.

Third, and most importantly, we show that pipeline-based interventions, namely, changing the prediction of risk of underreporting from a classification task to a regression task, is successful at improving model performance according to vertical equity. Changing the risk estimation model's prediction target shifts the audit burden substantially toward high income individuals, while increasing revenue. This result showcases the utility of pipeline-based interventions for real-world machine learning fairness problems: not only does changing the prediction task of the model from classification to regression increase fairness desiderata with little performance tradeoff, but using a pipeline-based approach allowed for the intervention to conform to context-specific fairness goals.

Last, we investigate the role of differential audit cost in shaping the distribution of audits. Audits of lower income taxpayers, for instance, are typically conducted by mail and hence pose much lower cost to the IRS. These results show that a narrow focus on return-on-investment can undermine vertical equity. Our results have implications for ongoing policy debates and the design of algorithmic tools across the public sector.

The annual tax gap, namely the difference between taxes owed and taxes paid, is estimated to be $440B in the United States [125]. Audits are the principal mechanism by which the Internal Revenue Service (IRS), the agency responsible for tax collection, verifies tax compliance and deters non-compliance. IRS resources are limited and the agency must use audits judiciously. During audits, the IRS typically solicits additional information from taxpayers to support information reported on filed returns. For the taxpayer, audits can be time-consuming, stressful, and costly [136, 165]. Low-income taxpayers, for whom refunds can comprise a substantial part of income, may wait “on their refunds to pay day-to-day living expenses such as rent, car repairs, or healthcare, and any delay can cause taxpayers significant hardship" [6].

Since the 1970s, the IRS has used classification models as part of its audit selection process to detect which individuals are most likely to have misreported their tax liability. While the use of both classical and modern machine-learning models is foundational to many government agencies' efforts to modernize predictive and allocative tasks [79], the adoption of such tools comes with considerable risks. The algorithmic fairness literature has amply documented how disparate impact and other negative outcomes can arise from the uncritical adoption and application of such models [10, 55, 151]. Given the scale and impact government decisions may have, mitigating these risks is a key priority for researchers and policy [190, 207]. In this work we study the impact of, and safeguards for, fairness of machine learning models in the IRS tax audit context.

Specifically, our analysis focuses on fairness defined in terms of vertical equity, namely, appropriately accounting for relevant differences across individuals. This notion is central to public finance and public policy. By contrast, the algorithmic fairness literature has developed many formal definitions of fairness and techniques to satisfy notions of horizontal equity (treating like individuals alike) [72, 105, 142]. The applicability of these techniques to improve vertical equity has been little-explored. More generally, the literature on how to apply algorithmic fairness techniques to improve real-world systems remains in a nascent stage, especially in high-stakes policy settings where direct data and systems access can be challenging. Using anonymized IRS microdata, our work (i) examines the applicability of existing methods for promoting vertical equity in the tax audit context, (ii) introduces new algorithmic fairness problems motivated by vertical equity considerations, and (iii) provides a case study of addressing vertical equity concerns in a real-world algorithmic decision system. By introducing vertical equity to algorithmic fairness, we follow in the footsteps of others [21, 110, 118] that situate fairness in broader frameworks.

Our point of departure and the key motivation for our study is summed up in two key observations that, taken together, point to a discrepancy between the distribution of misreporting compared to the distribution of audits: (1) the audit rate for lower-to-middle income earners is often as high or higher in recent recent years than that of high income earners; yet (2) an analysis of randomly conducted audits reveals that the amount of misreported tax liability (which we refer to, interchangeably, as the "misreport amount" or "adjustment") is highest among the highest income earners and the rate of misreporting—defined as misreporting above $200—increases roughly monotonically with income. With this context, our key research questions are as follows:
(1) **To what extent does the choice of audit selection algorithm affect the noted discrepancy?** Given the discrepancy between ground truth misreporting and audit allocations, we might expect that introducing a more accurate model may mitigate the issue. However, we observe empirically that more flexible models, while indeed increasing accuracy, have the effect of even *further* concentrating of the audit burden on the lower-to-middle income taxpayers.
(2) **Can existing algorithmic fairness methods, originally designed to promote horizontal equity, be applied to improve vertical equity?** In our context, one conception of vertical equity consists of monotonicity of the audit rate with respect to income. We show that, under some conditions, a selection process[1] that satisfies the well-known fairness metrics of equal true positive rates and equalized odds also requires monotonicity of the audit rate with respect to the misreport rate. Given our empirical findings, this also implies monotonicity with respect to income. We thus divide taxpayers into *income buckets* and explore to what extent conventional fairness methods applied to such buckets can resolve the apparent discrepancy between the audit rate and misreporting. We show that such methods come at a steep cost to revenue.
(3) **What techniques can we use to more directly address vertical equity in the IRS audit allocation context?** We implement a direct approach to achieve monotonicity by imposing allocation constraints on model outputs, and find that this approach results in a modest cost to revenue. However, we find that switching the prediction task from classification to regression not only also achieves a roughly monotonic shape, closely matching the audit distribution of an *oracle* with knowledge of the true misreport amount, but also obtains *significantly more revenue* than even unconstrained classification. This is because regression shifts focus to taxpayers likely to have high amounts of underreporting rather than simply high probabilities of a misreport.
(4) **Can differential audit costs explain the status quo mismatch?** We show that fully optimizing for return-on-investment with respect to the IRS' *audit costs* concentrates audits nearly exclusively on lower income taxpayers, even when using predictions arrived at via regression. This suggests that IRS budgetary constraints may play an important role in shaping the agency's ability to more equitably allocate audits without sacrificing the detection of under-reported taxes. A narrow focus on return-on-investment can seriously undermine vertical equity goals.

[1] By 'selection process,' we mean the prediction model and the process by which predictions are used to allocate audits together.

A major contribution of this work is that we conduct all our experiments on real, detailed, audit data collected by the IRS. We view this collaboration as an important case study to assess and mitigate disparities in real-world, public sector settings that operate subject to binding operational constraints [see 35, 95, 114, 146, 186]. Our primary dataset consists of a stratified random sample of taxpayers collected as part of the IRS' National Research Program (NRP), allowing us to avoid the selective labels problem [144], to draw inferences on a representative dataset, and to directly measure the risk of misreporting. Our work also connects to work that emphasizes the choice of prediction task [180, 186] and problem formulation [197] for algorithmic fairness. In addition, our results speak to current policy debates about the fairness of tax administration [135] and appropriate funding levels for the IRS [7].

The chapter proceeds as follows. Section 5.1 provides background on the U.S. tax system and spells out the motivating stylized facts, setting up the question of what the IRS's turn to machine learning may portend for vertical equity. Section 5.2 provides background on data and key definitions. Section 5.3 formally describes the audit problem, introduces notation, and discusses how extant fairness metrics might apply to the IRS context. Our main investigation is presented in four parts. First, Section 5.4 examines the impact of more powerful classifiers on audit distribution. Second, Section 5.5 presents the results of applying established algorithmic fairness techniques in our setting. Third, Section 5.7 studies the incorporation of monotonicity constraints as well as the simple but fundamental change of switching from classification to regression. Fourth, Section 5.8 examines the implications of accounting for audit costs. Section 5.9 concludes.

5.1 Background on the US Tax System

We examine individual federal income taxes in the US system. Taxes are assessed based on self-reported liability statements called *tax returns*, which can be time consuming and complicated to prepare; many taxpayers use commercial software or paid preparers. The tax rate on income is progressive, with marginal tax rates increasing in income.

As the tax code is very complicated, taxpayers (and their preparers [171]) often make errors when calculating the amount they owe and are thus inadvertently non-compliant; others are willfully non-compliant, i.e., evade paying taxes. The annual gross tax gap, which measures total noncompliance, is approximately $440B [125]. In order to recover lost revenue, and to promote compliance with the income tax law, the IRS audits individuals that it believes may not be paying their full owed tax—due to, e.g., erroneously claiming credits or under-reporting income.

The IRS' audit selection system is complex, with many parts. It principally relies on: (i) algorithmic methods to predict which taxpayers are most likely to underreport taxes, which serves as our main focus, (ii) a combination of simple rules that flag returns automatically; and, to a lesser extent, (iii) tips and other third party information, such as from whistleblowers. We focus on the algorithmic component of the IRS audit selection process, which has historically been a classification algorithm predicting individual taxpayer misreport [117]. The details of existing modeling approaches are confidential, but historically, the basic approach involves a form of linear discriminant analysis.

Audits are conducted in different ways depending on the size and scope of issues identified.

Some audits, including most involving the Earned Income Tax Credit (EITC), are conducted by mail at relatively low cost to the IRS. More complicated and extensive audits may be conducted by interview or by IRS examiner field visits. The timing of an audit relative to the processing of a return also varies. For instance, audits may be conducted on taxpayers claiming refunds before a check is sent out; this is known as revenue protection, and such audits are called "pre-refund". Audits occurring after a check has been sent out to, or received from, the taxpayer are known as "post-refund." These timing distinctions create differential impact on taxpayers, and may also affect the ease with which the IRS conducts audits.

Over the last eight years, budget cuts have decreased the audit rate, from an overall rate of 1% of individual filings receiving audits in 2010 to just 0.5% in 2016 [123]. The audit rate has decreased most significantly for individuals earning between $1-5M. Such individuals were audited at a rate of ≈8% in 2010 but just 2.2% in 2016 [123]. These changes in audit rates correspond to disproportionate reductions in examiners with more specialized expertise: while there was a 15% reduction in examiners conducting correspondence audits (i.e. audits by mail) from 2010 to 2019, there was a 25-40% reduction in examiners conducting in-person audits, which are utilized more for higher-income individuals [122].

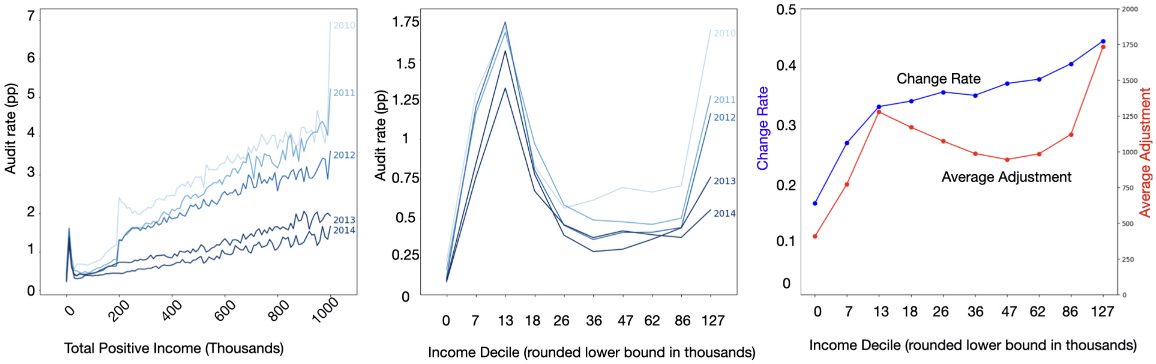


Figure 5.1: Left two graphs: Audit Rate vs. Total Positive Income over time. Both of these graphs are calculated on operational audit (OP) data. Each line of a different color represents a different year, from 2010 to 2014. The x-axis indicates income binned into buckets of income, while the y-axis is the fraction of taxpayers in each bucket audited. On the leftmost, we have reported income buckets of $10,000, up $1m, while on the second graph, we show the same analysis over reported income deciles. Note that as the 10th income decile starts at 127K, this graph is comparatively compressed. Right: Ground truth rates of misreporting (over $200) (left) and average amount of misreporting conditional on misreport, aka average adjustment (over $ 200), (right) over income. The results here are presented over five years of NRP data 2010-2014, adjusted to 2014 dollars. The x-axis denotes income deciles, and the y-axis denotes rate of misreporting and average amount of misreporting in dollars, respectively. Taken together, we can see that there is a mismatch between audit allocation and ground truth noncompliance.

5.1.1 Motivating Facts

We highlight two motivating facts relevant to our investigation. First, in the most recent years, the lowest income earners have been audited at the same rate as the highest income earners. The left panel of Figure 5.1 plots income in $10K bins from $0 to $1M on the x-axis against the audit rate on the y-axis. Each line represents one year, from 2010 in lightest to 2014 in darkest blue. This panel shows the clear trend of the declining overall audit rate over time, which affects higher income groups most acutely. In addition, while audit rates generally increase in income, there is a large spike of audits in the lowest income groups. In 2014, the lowest earners are audited at a higher rate than all other income groups, except for those earning nearly $1M. The middle panel depicts the same data using income deciles. After 2010, low-to-middle income taxpayers (i.e. those in the 2nd-4th income deciles from $6.7K to $26K), were audited at a higher rate than all higher income deciles. This reflects the particular focus on pre-refund audits done principally by mail.

Second, the rate at which taxpayers understate their tax liability increases monotonically with income and average adjustments are highest in the highest income decile. The right panel presents audit outcomes estimated on the NRP data (described in Section 5.2 below). The blue line in this panel depicts the estimated fraction of audits in each decile with a true misreport of at least $200, while the red line depicts the average adjustment by decile. Because this is a stratified random sample with corresponding sampling weights, it is free of the selection bias inherent in measuring outcomes among risk-selected audits, and can thus be used to construct consistent estimates of *population* non-compliance.

These facts raise the motivating questions of this work: if adjustments are highest in the highest income decile, and the misreport rate increases monotonically with income, then why are audits so highly concentrated on lower-to-middle income taxpayers? To what extent can such patterns be exacerbated or mitigated by machine learning techniques? And are there opportunities for improving vertical equity given this mismatch?

5.2 Data and Key Terminology

We address these questions through a unique collaboration with the Treasury Department and IRS, which provides us access to two data sets previously unexplored in the computer science literature: (1) the NRP data, which consists of line-by-line audits of a stratified random sample of the US population (n=71.9K) from 2010-14 [121]; and (2) all Operational Audits (OP) for 2014 (n=791.9K), which are risk-selected audits to identify tax evasion. Each observation contains information filed in a tax return. All dollar amounts are adjusted for inflation to 2014 dollars.

We train and evaluate our machine learning models on NRP data, as this data is a random, representative sample of the US population and does not suffer from selection bias [144].[2] We note that the OP audit data includes observations that were selected for audit not solely through machine learning tools, but also through rule-based flags such as internal inconsistencies, and other methods of selecting audits. We use the OP data to display the status quo of audit selection in the IRS as of 2014, for example, in the left-most graphs in Figure 5.1.

In this data, three concepts are particularly important. First, by *income*, we mean the taxpayer's reported *total positive income* (TPI). TPI captures all positive income an individual receives, gross of any losses.[3] We focus on reported (rather than audit-adjusted) income because that is what the IRS observes at the time it selects taxpayers for audit, and we focus on TPI (rather than taxable income) because it represents a simple measure of earnings that

[2] That is, when a return is selected for OP audit, the IRS has reason to believe that the return represents a misreport. Hence, the return is likelier to have a large adjustment than a randomly selected return from the population, and may be more generally non-representative as well. That said, one limitation is that prior work has found that NRP data under-reports higher income tax evasion [**?**].

[3] Not all this income is taxable—for instance, tax deductions for losses or charitable contributions may reduce the total amount of taxable income.

is less likely to be affected by audit determinations. Many of the analyses in this chapter will be over binned income, i.e. discretized income into equal-sized buckets, typically taken to be deciles of the income distribution.[4]

Second, we refer to the amount by which a taxpayer's return understates true tax liability as the *misreport amount*. If a taxpayer overstates their tax liability, then their misreport amount is negative. Throughout, we use the terms "adjustment" and misreport amount interchangeably. For classification, we define a *significant misreport* as whether the taxpayer's understated tax liability exceeds a de minimis amount ($200). For brevity, we refer to these simply as *misreports*. Our findings are consistent across different choices of threshold (see Appendix D.3).

Third, we define the *cost* of an audit to the IRS as the total cost of the auditor's time recorded on the particular audit, which we compute from auditor time[5] and wage data. In principle, audit costs also include other components, such as overhead or attorney's fees for litigated cases, but these are not possible for us to measure with our data. Note that we are focusing only on the budgetary costs of audits to the IRS, not the broader societal costs imposed on taxpayers.

5.3 The Audit Problem

To explore vertical fairness in audit allocations, we start with the tools most readily available to improve the fairness of algorithmic tools: the now-canonical fairness definitions applied in the literature [105, 244]. In this section, we first formalize the audit selection problem. Second, we discuss vertical equity in the context of the audit allocation problem, and consider how common fairness definitions may improve vertical as well as horizontal equity in this context. Third, we discuss implementation of these metrics and model evaluation.

5.3.1 Formal Definitions and Preliminaries

In this work we define the basic audit problem as the following: given a budget and a set of taxpayers with associated features and audit costs, return a selection of taxpayers for audit that detects and recovers as much under-reported tax liability as possible within the given budget.[6]

For the majority of this chapter, we model the budget K as a fixed number of audited tax returns, which we represent as a percentage of the population. We use a budget of 0.644%, which is the average percentage of audit coverage between 2010-2014. Taxpayers are indexed by $i \in 1, ..., N$ and have features X_i. One of the features in X is $\mathcal{I}_i$, the taxpayer's income. The *income bucket* $b_i \in \mathcal{B} = 1...10$ of the taxpayer is the decile of $\mathcal{I}_i$. Taxpayers submit a report of tax liability $\tilde{\ell}_i$, which may be different than their true liability ℓ_i. We let $\delta_i = \ell_i - \tilde{\ell}_i$ denote the taxpayer's adjustment or misreport amount. We will also use $m_i = \mathbf{1}[\delta_i > \tau]$ for an indicator variable being above the misreport threshold τ. In our main experiments, we set $\tau = 200$, and write $\pi_i := \Pr[\delta_i \geq \tau | X_i]$. We denote the cost incurred to the IRS by auditing an individual i as c_i. We use a_i as an indicator for whether taxpayer i is audited, and α_i for

[4] While these bins and associated thresholds are relevant to our analysis and implemented algorithms, to our knowledge they are not currently used by IRS to categorize returns or to determine taxpayer eligibility for benefits.

[5] Notably, our available data for auditor time does not account for auditor time spent on audit appeals.

[6] In practice, the audit problem undertaken by the IRS must balance a variety of objectives, including revenue maximization, deterrence, minimization of taxpayer burden, and reduction of improper payments.

a probabilistic relaxation. Occasionally, we use $\hat{\cdot}$ to indicate prediction, e.g. $\hat{\delta}_i$ as predicted misreport amount.

The machine learning models we use throughout this chapter which we integrate into the audit selection process either predict *probability* of misreporting $\hat{\pi}_i$ (for classification models), or *expected amount* of misreporting $\hat{\delta}_i$ (for regression models). In order to create an audit allocation from these predictions, however, we must select only 0.644% of the population, which is in practice much less than the percentage of individuals predicted to not comply. Thus in order to create an audit allocation from machine learning model predictions, we rank model outputs by magnitude of prediction and take the top 0.644%. The audit problem can be formalized as: $\max_a \sum_i \delta_i \cdot a_i$such that $\frac{\sum_i a_i}{N} < K$.

If we consider K to denote a *dollar* budget as opposed to an audit rate budget, as we do in Section 5.8, the constraint will be changed to $\sum_i a_i c_i < K$. In practice, we use $\hat{\delta}_i$ or $\hat{\pi}_i$ to approximate δ_i.[7]

5.3.2 Algorithmic Fairness and Vertical Equity

We now discuss vertical equity in the IRS audit allocation context and its connection to several common algorithmic fairness metrics from the literature.

Vertical Equity. Vertical equity requires that different individuals be treated *appropriately differently.* In the taxation and audit context, we focus on vertical equity with respect to the appropriate treatment of taxpayers at different income levels. Appropriately different treatment depends on context-specific considerations and value judgments. To illustrate, given the fact that audits are costly for taxpayers (in terms of money as well as time, effort, and mental stress), policymakers may wish to avoid models that concentrate audits on low-income taxpayers out of concern for distributional social goals and in recognition of the declining marginal utility of taxpayers' income. Other potential baselines for setting policy in this space are aligning audit rates with true rates of non-compliance, or with an *Oracle*-based selection, i.e. an allocation which selects individuals in order of true misreport amount. In our setting, because under-reporting rates increase with income (Figure 5.1) and an oracle places a higher probability of selection as income increases, these factors would suggest that audit rates should increase in income as well. Motivated by such considerations, we explore formalizing the notion of vertical equity as *monotonicity* and evaluate the discrepancy between audit allocation and true rates of misreport as an important component of vertical equity. Our focus on monotonicity is intended to illustrate how one might incorporate vertical equity concerns into algorithmic fairness, but we note that a fuller analysis from an optimal tax framework is beyond our scope here.[8]

Montonicity Monotonicity (with respect to income) would require that the audit probability increase as income increases. Formally, given income buckets b and b', $b \geq b' \implies \Pr[a_i =$

[7] As stated, this is an integer program, but we solve the linear relaxation due to computing constraints and because observations represent many people.

[8] A full optimal policy analysis would have to consider such factors as heterogeneity in the audit burden or in the deterrence effect of audits by income. For example, audits of higher income taxpayers can be more involved, but audits of lower-income taxpayers may require obtaining harder to produce information and often involve freezing refunds for liquidity-constrained taxpayers while the audit proceeds. A fuller optimal policy analysis would also need to consider how audit policies interact with other tax variables (such as the income tax schedule and underpayment penalties) for achieving revenue and distributional goals. Each of these factors may impact vertical equity.

$1|b_i = b] \geq \Pr[a_i = 1|b_i = b']$. We consider directly constraining the audit allocation to be monotonic in Section 5.6.

Oracle Allocation An *oracle* is a theoretical omniscient model with access to the true amounts of misreporting in the data (i.e. the ground truth labels). Formally, the oracle represents the model $\hat{\delta}_i = \delta_i$, where δ_i is the amount of true misreport of individual i. The oracle creates an audit allocation by selecting individuals for audit in order of their true amount of misreport amount until exhausting the allocation budget. Thus, the audit allocation selected by the oracle is naturally aligned with true incidence of misreport. Although we do not explicitly enforce this behavior, we evaluate the vertical equity of model allocations by the extent to which they match the audit rate by income of the oracle model.

Demographic Parity. Demographic Parity (DP) requires, in our context, equal audit probability across income buckets. That is: $\Pr[a_i = 1|b_i = b] = \Pr[a_i = 1|b_i = b'], \forall\ b,\ b'$. Note that with a fixed budget and groups of equal size, asking for DP amounts to requiring the same audit rate for each group, which weakly satisfies monotonicity. Compared to the status quo described in Figure 5.1, this would result in lower audit rates for low-to-middle income taxpayers as well as very high income taxpayers, and higher audit rates for middle-to-upper income taxpayers. Important limitations to DP include that (1) as noted, equal audit rates do not imply equal audit burdens if taxpayers bear different costs, and (2) a perfectly accurate classifier would not satisfy DP unless the misreporting rates are exactly equal, which they are not.

Equal True Positive Rates [105]. Equal True Positive Rates (TPR) requires that the audit probability of *non-compliant* taxpayers not depend on income group, i.e., $\Pr[a_i = 1|m_i = 1, b_i = b] = \Pr[a_i = 1|m_i = 1, b_i = b'], \forall\ b,\ b'$. Equal TPR ensures that no group of non-compliant taxpayers can expect a higher or lower chance of audit based solely on their income, but this does not mean that compliant taxpayers of each income group face the same chance of an audit.

Equalized Odds. Equalized Odds (EO) asks that the audit probability of both compliant and non-compliant taxpayers should not depend on their income group, i.e.: $\Pr[a_i = 1|m_i = 0, b_i = b] = \Pr[a_i = 1|m_i = 0, b_i = b']$, and $\Pr[a_i = 1|m_i = 1, b_i = b] = \Pr[a_i = 1|m_i = 1, b_i = b']$. EO extends equal TPR fairness by requiring audits of compliant taxpayers at the same rate across groups in addition to auditing non-compliant taxpayers at the same rate across groups.

In Appendix A, we consider conditions under which equal TPR or EO will result in monotonicity of the audit rate with respect to income. Specifically, we consider a hypothetical allocation that audits all taxpayers with $\hat{\pi}_i > 0.5$, and show that under certain (differing) conditions, audit allocations that satisfy either either equal TPR or EO will result in monotonicity of the audit rate with respect to the misreport rate. Because the misreport rate increases with income (Figure 5.1), this suggests that enforcing one of the fairness constraints on a model generating audit allocations may also lead to monotonicity of audits with respect to income. We note that this result is suggestive, since models that satisfy a fairness constraint for the hypothetical allocation described above need not do so for the actual audit allocation induced after imposing a budget. Thus, we must ultimately test whether the targeted fairness constraints are satisfied on the audit allocation that results from a model once a budget is incorporated. Next, we describe algorithms to instantiate these conditions and evaluate the performance tradeoffs. We implement these algorithms and report results in Section

5.5.

5.3.3 Model Evaluation

In order to compare model allocations, we will consider several performance metrics. First, in order to approximate how well an audit allocation matches the ground truth rate of misreport, we consider how closely audit rates correspond to selection based on an oracle. Specifically, we calculate the *overlap* between a model's allocation and the oracle's, formally, the size of the intersection of the model and oracle's audit allocation over the total number of audits in an allocation: $\frac{\sum_i a_{i,O} a_{i,M}}{K \times N}$, where $a_{i,O}$ and $a_{i,M}$ represent audit indicators for the oracle and a model respectively, K is the audit budget as a percentage of the population, and N is the total number of taxpayers.[9] Note that the overlap will be between 0 and 1, with 1 representing an exact match of the oracle's allocation. We consider models that more closely match the oracle allocation with respect to income to have preferable vertical equity performance in our context.

Second, we consider *revenue* collected, which is simply the sum of adjustments over all audits. Recovering revenue is one of the key goals of the IRS and is itself relevant for distributive policy, since it funds services provided to citizens. We define revenue as follows: $\sum_{i \in N} \mathbf{a}_i \delta_i$, where N is the number of taxpayers in the dataset.

Third, we consider the *no-change rate*, which is the fraction of audits resulting in no (substantial) adjustment. No-change audits are undesirable from both IRS and taxpayer perspective, as both the auditor and taxpayer could have saved significant time, effort, and stress. We define the no-change rate as $\frac{\sum_i \mathbf{a}_i \cdot (1-m_i)}{\sum_i \mathbf{a}_i}$.

Fourth, we consider the *cost* of the audit to the IRS, which is important both in terms of the feasibility of an audit policy and its net revenue implications. We define cost as $\sum_{i \in N} \mathbf{a}_i c_i$, where c_i is our estimate of cost per return.We describe how we obtain cost estimates in Section 5.8. In Sections 5.4-5.7, we hold audit rates fixed and measure incurred cost. In Section 5.8, however, we consider constraints on the total dollar cost of policies, and show how they may explain the existing discrepancy between income and the audit rate.

5.3.4 Model Implementation

There exists a large body of research surrounding how to best implement and guarantee the common fairness metrics outlined above [8, 38, 65, 105, 134, 263]. From this rich literature, we choose to rely on a technique developed by Agarwal et al. [8], which intervenes in a model's training process to add a constraint during optimization which incentivizes the model to satisfy a given constraint in its predictions [8, 65]. Methods that enforce fairness constraints during training time are often described as “in-processing," as opposed to those which intervene at prediction time, which are called “post-processing." Agarwal et al.'s (in-processing) technique allows for demographic parity, true positive rate parity, equalized odds, and other constraints to be satisfied in expectation in a model's predictions on the training distribution. We include results from other methods of enforcing fairness constraints, including post-processing techniques, as a discussion of the differences between various methods in Appendix D.6.

[9] The total number of taxpayers, taking into account the sampling weights. This metric is equivalent to the top-k intersection of model outputs, where k is the audit allocation budget. This metric is often used to compare model-generated explanations [25, 61, 96].

Model Type	Label Type	Fairness Constraint	Revenue ($B)	No-Change Rate	Cost ($B)	Net Revenue ($B)	Oracle Overlap
Oracle	-	×	29.40	0.0%	0.33	29.07	1.00
LDA	Class	×	6.07	12.8%	0.21	5.86	0.09
Random Forest	Class	×	3.05	3.5%	0.08	2.97	0.00
Grad Boosted	Class	×	4.05	4.2%	0.08	3.97	0.00
Random Forest	Class	✓(DP)	2.75	8.0%	0.07	2.67	0.08
Random Forest	Class	✓(TPR)	0.69	12.4%	0.15	0.54	0.04
Random Forest	Class	✓(EO)	0.53	13.6%	0.15	0.38	0.04
Random Forest	Class	✓(Mono)	3.00	4.0%	0.10	2.90	0.01
Random Forest	Reg	×	10.22	23.3%	0.50	9.72	0.23
Grad Boost	Reg	×	10.20	20.0%	0.50	9.70	0.22

Table 5.1: Revenue, no-change rate, cost, net revenue, and oracle overlap for all models considered in this chapter. No-change rate represents the percentage of audits that were allocated to compliant taxpayers; cost reflects cost to the IRS as described in Section 5.8. These results reflect audit allocations that select the top 0.644% of taxpayers predicted most likely to misreport from each model. All metrics are reported on the test set, using the representative NRP sampling weights to scale up to the US taxpayer population.

5.4 Flexible classifiers and audit classification

We begin by examining the hypothesis that the disproportionately high audit rate observed for low income earners may stem from using simpler classification models in guiding audit allocations. We demonstrate that (i) the disparity displayed in audit rates does not appear to arise from the less complex models similar to those the IRS has historically used; and, (ii) applying more complex models—in this case, Random Forests and Gradient Boosting—actually *exacerbates* the burden on lower income taxpayers.

5.4.1 Experimental Setup

In this section, we consider the audit allocation determined by Linear Discriminant Analysis (LDA) (an approximation of the historical choice by the IRS), a Random Forest Classifier, and a Gradient Boosting Classifier. In principle, classifiers may perform well at reducing the no-change rate, furthering IRS's objective to avoid burdening compliant taxpayers. To be clear, the audit allocation is not simply the model's predictions, but rather the individuals most highly predicted for misreport up to the audit budget, as described in Section 5.3.1. We use NRP data from 2010-2014 to train all models in this chapter to predict the likelihood of misreporting. We randomly split this data into a train and validation (75%) and test (25%) sets. We search for optimal hyperparameters using *sklearn*'s GridSearchCV method with 5-fold cross validation.[10]

All results in this and following sections are calculated on the test set, which is reserved for reporting results. Results are reported by rescaling costs and revenues to reflect estimated average annual values for the full population (averaged between 2010-2014). For each classification model, we sort taxpayers in descending order of predicted *misreport probability* to produce a ranking. We then apply an audit rate budget of 0.644% of the population, reflecting

[10] As described in detail in Appendix D.2, we train all but LDA models with *sampling weights* provided in the NRP data, meant to ensure the data is representative of the taxpayer population. For LDA models, we sub-sample a dataset from the NRP data that respects the sample weights by randomly selecting (with replacement) rows from the weighted training data according to the weights. For example, suppose that each row x has a sample weight w, and the sum of all weights in the training set is W. Then each observation has a $\frac{w}{W}$ chance of getting selected as any given row in the sub-sampled data.

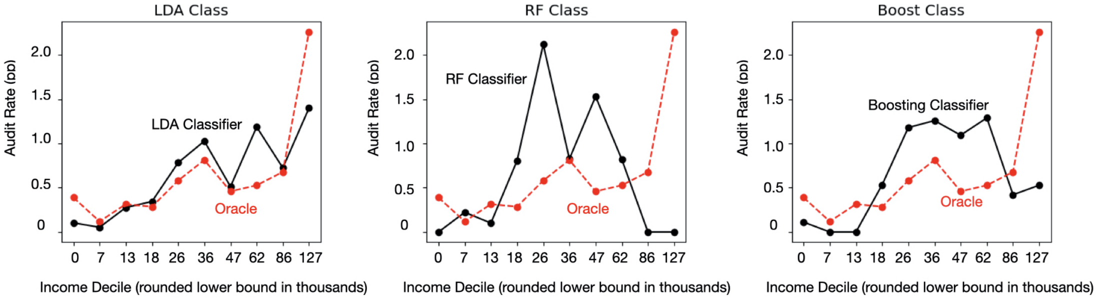


Figure 5.2: From Left to right: Audit Rate by Income LDA Classifier, Random Forest Classifier, and Gradient Boosted Classifier, presented in black. The oracle allocation on the same budget is presented in red on the same graph.

the average audit rate from 2010-2014, and select audits a_i by taking the top 0.644% of the population (i.e. 1125000 audits)in rank order. Further details are in Appendix D.2.

5.4.2 Results

Figure 5.2 displays the audit rate by income of allocations obtained via ranking the predictions of LDA, Random Forest Classification, and Gradient Boosted models by predicted probability of misreport and selecting the top 0.644% of the population. Revenue and no-change rate of these models are included in Table 5.1. We highlight implications below.

First, higher model flexibility can lead to high audit focus on lower and middle income populations. As Table 5.1 shows, the Random Forest Classifier is well-optimized for the classification task: it has an extremely low no-change rate—just 3.5%—whereas simpler models have no-change rates higher than 12.8%. However, the Random Forest Classifier focuses almost exclusively on the lower-middle and middle of the income spectrum, not targeting the highest earning 20% at all. Similarly, the Gradient Boosted classification model concentrates most of the audit selection to the middle of the income spectrum (4-8th decile), with a strong drop-off for the top 20% of the population. (Appendix D.4 shows that another simpler model (logistic regression) also results in rough monotonicity.)

Second, the simpler LDA model more closely matches the oracle. The LDA classifier has an audit selection curve that is roughly monotonic in income, with large increases in audit rate in the high income region. As LDA has been the IRS' historical modeling approach (although it differs in practice with our implementation), this suggests that the large spike in operational audit selection rate on the lower end of the income spectrum apparent in 2014 may not stem directly from the predictions algorithmic components of the decision system, but rather other policy and modeling choices.

Third, increased classification accuracy does *not* imply increased revenue. Table 5.1 shows that the Random Forest and Gradient Boosted models have significantly lower no-change rates than the LDA model (3.5% and 4.2% vs. 12.8%), yet also substantially *lower* revenue (≈$3B and $4B vs. ≈6B). This highlights that improved performance on one objective (e.g., accuracy) may come at the expense of other seemingly intertwined objectives (e.g., revenue).

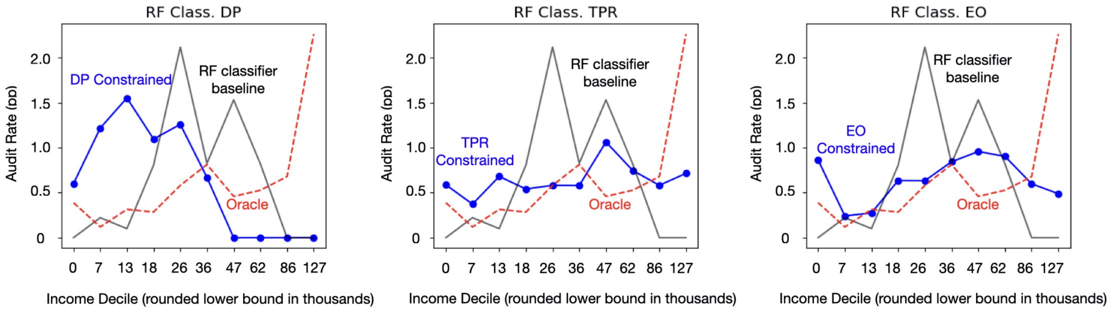


Figure 5.3: In-process fairness techniques imposed on a Random Forest model. From left to right: enforcing Demographic Parity (DP), Equal True Positive Rates (TPR), and Equalized Odds (EO). Black (blue) series represent the unconstrained (constrained) allocation.

5.5 Fairness Constrained Classification

We now explore the use of bias mitigation methods to promote vertical equity.

5.5.1 Experimental Details

We enforce algorithmic fairness definitions on the Random Forest model at different points in the audit selection process: *during* training, or in-processing, following Agarwal et al. [8], and *after* training but before prediction, or post-processing (deferred to Appendix D.6, following Hardt et al. [105]). Our setup for training the fairness-constrained models mirrors our setup for the fairness-unconstrained models, with the exception that we do not train the models with sampling weights, but rather subsample a dataset from the NRP weighted data as we do for LDA models as described in Section 5.4. This is because the in-processing methods are implemented using the FairLearn package [22], and the FairLearn package leverages *sklearn*'s sampling weight functionality in the course of their algorithm.

5.5.2 Results

Our high-level result is that enforcing fairness constraints during training results in steep trade-offs with limited fairness payoffs for the budgeted allocation problem. Figure 5.3 displays audit rate by income decile for Random Forest Classifier trained to respect each of the fairness definitions considered. We present revenue and no-change rate in Table 5.1.

Equal TPR and EO models do lead to overall lower focus on low and middle income groups. However, they continue to under-target the highest end of the income spectrum when compared with the oracle predictor. And perhaps surprisingly, despite this shift to focus slightly more on higher ends of the income spectrum, enforcing these constraints actually leads to a large decrease in revenue: from over $3B to as low as $600M in revenue. We additionally notice a decrease in the no-change rate towards levels closer to the baseline LDA predictor. Finally, they imperfectly enforce the targeted fairness constraints once the audit budget is imposed: this is immediately evident in the allocation from a model constrained to respect demographic parity, as the audit rate is not equal across groups.

Given these results, we argue that enforcing fairness constraints during training is not an effective technique to improve vertical equity in an audit allocation setting. We highlight some broader implications of vertical equity for algorithmic fairness in Section 5.9.

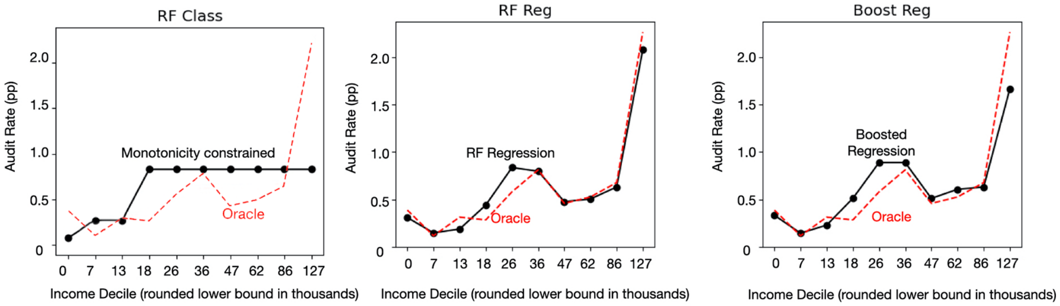


Figure 5.4: Left: Monotonicity constraints explicitly enforced on audit allocations of a Random Forest Classifier. The black line represents the allocation, the red line represents the oracle. Right: Audit Rate by Income in Random Forest Regressor, and Gradient Boosted Regressor, presented in black. The oracle allocation on the same budget is presented in red on the same graph.

5.6 Enforcing Monotonicity

In this section, we instead enforce monotonicity directly. We do this by solving the following linear program:

$$\max_{\alpha} \sum_{b \in \mathcal{B}} \sum_{i \in b} \alpha_i \hat{\pi}_i w_i \qquad \text{s.t. } \alpha_i \in [0,1] \forall i; \quad \sum_{b \in \mathcal{B}} \sum_{i \in b} w_i \alpha_i = 1; \sum_{i \in b_1} \alpha_i w_i \leq \sum_{i \in b_2} \alpha_i w_i \quad \cdots \quad \sum_{i \in b_9} \alpha_i w_i \leq \sum_{i \in b_{10}} \alpha_i w_i$$

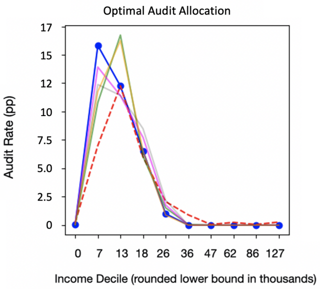
where all notation follows Section 5.3.1, w_i represents sampling weights, and the Random Forest Classifier generates $\hat{\pi}_i$.

The leftmost panel of Figure 5.4 shows the audit distribution of the solution to the linear program. Notably, all income buckets from the fourth decile and above are audited at the same rate. In other words, the constrained solution audits higher income deciles at the minimum in order to focus most energy on the fourth decile. The trade-off with performance is relatively modest relative to the unconstrained classifier, as seen in Table 5.1: revenue does decrease, but by only $50 million; the no-change rate increases by half a percentage point. These results indicate that, especially compared to enforcing traditional fairness constraints, enforcing monotonicity may be a relatively economical approach to encourage (one notion of) vertical equity. The next section shows, however, that this approach may be far from optimal.

5.7 From Classification to Regression

We now demonstrate that changing the model's prediction target from the *probability* of misreport to *expected misreport amount*—i.e. changing from a classification to regression algorithm— can reduce burden on lower-income taxpayers and make audit rates more closely mirror the oracle while also *increasing* revenue. This demonstrates that, in some circumstances, changing the model's prediction task to reflect behavioral desiderata–rather than enforcing a constraint on top of a model optimizing for an imperfectly aligned task—is a more effective technique to reach equity goals.

We train regression models with the same process described in Section 5.4 for classification models, but use the misreport amount as the label rather than to a binary indicator of misreport. The audit rate by income decile of Random Forest and Gradient Boosting



Models	NC Rate	Revenue	Cost	Net Revenue
Revenue Optimal Allocation				
LDA	0.33	18.8B	0.125B	18.7B
RF CLS	0.29	19.0B	0.125B	18.9B
RF Reg	0.35	21.1B	0.125B	21.0B
Oracle	0.000	41.0B	0.125B	40.9B
Naive Allocation				
LDA	0.22	2.3B	0.125B	2.2B
RF CLS	0.04	3.7B	0.125B	3.6B
RF Reg	0.30	2.2B	0.125B	2.1B
Oracle	0.000	12.0B	0.125B	11.9B

Figure 5.5: Left: Revenue-optimal allocation from all models considered in chapter so far, considering budget as a dollar amount. The x-axis represents income deciles, and the y-axis represents audit rate. We consider the budget to be 125 million, or the average budget over 2010-2014 using our approximation of cost described in Section 5.8. The revenue-optimal allocation requires that the individuals with the highest *ratio of revenue returned to IRS over cost to the IRS* are selected for audit up to the dollar budget, which results in a similar allocation from all models. Right: No-change rate, revenue, cost, and net revenue of allocations from different models considered in the chapter when modeling audit budget as a dollar amount, for both for net-revenue optimal and naive allocations.

regression models are displayed in black in Figure 5.4, along with the oracle in dashed red.

We highlight two chief results. First, shifting the prediction target from the probability of misreport (classification) to the expected amount of misreport (regression) shifts audit focus from lower income to higher income taxpayers, resulting an audit allocation that is not only nearly monotonic, but also closely matches the oracle allocation. As can be seen in Figure 5.4 and the right column of Table 5.1, the resulting allocation is in fact closer to the oracle than any other prior allocation. Thus, changing from a classification to a regression task can be seen as one method to directly optimize for (multiple notions of) vertical equity in the IRS context.

Second, while changing the prediction target from presence of significant misreport to amount of misreport does increase the no-change rate (up to 20-23%), it also results in a dramatic increase in revenue. Table 5.1 shows that assessed revenue under regression rises to \$10B, compared to the \$3.6B baseline of high-powered classification models.

Thus, within the set of higher complexity models, switching from classification to regression may provide an effective way to decrease the mismatch between audit allocations and ground truth levels of misreport, as well as decrease audit focus on lower and middle income individuals, while *increasing* under-reported tax liability detected by the IRS. We leave the discussion of how regression-based allocations interact with the IRS goal of broad-spectrum noncompliance deterrence—which may necessitate additional focus on lower-magnitude noncompliance—to future work.

5.8 Agency Resources and The Impact of a Narrow Return-on-Investment Approach

We now turn to examining the relationship between vertical equity and agency resources. As noted, how an audit proceeds depends upon the type of noncompliance suspected: for example, many audits on lower-to-middle income individuals concern a potentially incorrectly claimed tax *credit*, whereas audits on higher income individuals more often involve insufficient taxes being paid on income or other assets [122]. Audits concerning tax credits are largely done via correspondence, where the IRS sends a letter to the taxpayer requesting verification of qualification for the claimed credit [122]. Other types of misreporting often incur in-person IRS audits [122]. Correspondence audits are extremely resource-efficient for the IRS. On the other hand, in-person audits require more time and expertise, and tend to incur much higher costs. Further, a non-response from a correspondence audit is taken as an admission of non-compliance, resulting in revenue returned to the IRS [101], and keeping investigation costs low. One study on EITC correspondence audits found that up to 75% were determined to be noncompliant due to nonresponse, undeliverable mail, or insufficient response [101]. Thus, the ease of correspondence audits, coupled with the high nonresponse rate leading to frequent revenue returned to the IRS, may result in more reliably recovered income than in-person audits, in addition to their lower direct costs. Here, we use a simple model to explore whether a constrained monetary budget, coupled with differential cost of audits across the income spectrum, might affect audit allocation. We model the audit budget in terms of a *dollar* cost[11] as opposed to a constraint on the fraction of the population audited.

5.8.1 Experimental Details

In our consideration of the effects of agency resource limitations on audit allocation, we focus on the dollar cost of audits to the IRS and its budgetary constraints. We calculate a simplified version of cost that only takes into account the cost of the actual tax examination, based on data from previous real operational audits. We calculate cost as the product of the examiner's time spent on a given audit with their hourly pay. We average this product over income deciles and *activity code*, which roughly corresponds to groupings of individuals based upon what tax forms they have filled out, to estimate audit cost. We incorporate cost into our analysis by directly including the dollar budget as an audit selection constraint, thus creating a linear program to maximize total predictive value (i.e. probability or amount of misreport) with respect to the dollar budget. As we show in Appendix D.7, this formulation is equivalent to a fractional knapsack problem; thus, the optimal solution is to select individuals in order of their ratio of cost to return to the IRS, in other words, return-on-investment. We use a dollar budget of $125M, the average estimated total cost of audits from years 2010-2014. Further details are in Appendix D.8.

5.8.2 Results

We present three main results. First, due to the differing *audit costs* to the IRS by income, return-on-investment focused audit selection results in an allocation which overwhelmingly targets lower income taxpayers. In the left panel of Figure 5.5, we show the optimal audit selection policy under a dollar budget with rankings from each of the models considered in the chapter thus far. As described in Appendix D.7, the revenue-optimal audit allocation is to choose returns with the return on investment, i.e. the best ratio of predicted reward (adjustment in regression or change probability in classification) to audit cost. Based on our calculations of audit cost, audits in the highest income decile may cost up to 41 times

[11] We note that a fixed monetary budget may not perfectly capture the resource constraints faced in practice; for instance, the limited number of auditors of a given expertise level may bind more tightly than any short-term dollar budgets. Still, this simplification captures important heterogeneity in the degree to which audits push against agency resource constraints. In addition to shedding light on the status quo audit distribution, such analysis may be interesting to the field of applied ML, as relatively few papers consider budget-constrained allocation models.

the least costly audits. Given the disparities in audit costs over the income spectrum, the revenue-optimal audit selection method results in an allocation that almost exclusively targets lower income individuals.

Second, the return on investment of auditing lower income individuals may shed light on the status quo allocation's focus on low and middle income individuals. We note that the optimal allocation with a dollar budget looks similar to the 2014 operational audit selection policy (Figure 5.1). Given the decreasing IRS budget over time, prioritization of net revenue maximization may influence the vertical equity of status-quo allocations. However, we note that the extremely low cost of audits on the lower end of the income spectrum result at least partially from a policy choice made to proceed with different types of audits in asymmetric ways: i.e., via *correspondence audits* on the lower end of the spectrum, and in-person audits on the higher end. This decision, coupled with the choice to view a lack of response as noncompliance, results in less time, and fewer resources, spent on audits for individuals in the lower end of the income spectrum, thus resulting in the constrained revenue-optimal allocation focusing so highly on low-income individuals.

Third, we find that to improve vertical equity and increase revenue collected, regression models require a higher dollar budget. As demonstrated in Section 5.7 and Table 5.1, regression models produce the highest net revenue allocations amongst models constrained to only audit a given percentage of the population (0.644%). However, the cost to the IRS of these allocations are considerably higher than classification methods—and indeed, higher than our approximation of average IRS budget between 2010-2014, $125M. At this low dollar budget, regression models under-perform on revenue compared to classification models, demonstrated in the right panel of Figure 5.5: this is because regression models target individuals in the higher income realm, where the audit cost is greater, thus preventing such allocations from targeting enough individuals to generate high revenue returns. This suggests that increasing the dollar budget available for audits may present an opportunity for not only more net revenue, but also in a more equitable allocation of audits.

5.9 Discussion

Through this unique collaboration with the Treasury Department and IRS, we have studied the impact of machine learning on vertical equity. Our work suggests that: (1) more accurate *classifiers* may exacerbate rather than improve income fairness concerns; (2) off-the-shelf fairness solutions are not well-suited for attaining income fairness; (3) fundamental modeling changes, like switching from a binary target to a regression target, can improve income fairness; and (4) external constraints, like institutional budgets, may influence fairness regardless of what underlying predictive model is used. Specifically, a return-on-investment focused audit allocation may undermine vertical equity under current conditions. More broadly, this work underscores the importance of vertical equity, in addition to horizontal equity, in real-world application areas of machine learning. To our knowledge, the term does not appear in the algorithmic fairness literature,[12] and traditional fairness metrics can be seen as focusing on horizontal, rather than vertical, equity. Given the importance of achieving

[12] Outside the fairness community, but inside the general umbrella of technology and engineering, the term *has* been used; in particular, [257] use both terms in a study of equity in access to transportation, and point towards a possible link to algorithmic fairness. However, their interpretation of vertical and horizontal equity are substantially different from ours; for instance, they suggest that group fairness should be linked to *vertical* equity.

vertical equity for policy, this work points towards further development of algorithmic fairness techniques as a promising path for future research.

Our results also reveal a subtle dimension of fairness when resources are allocated under a budget constraint. When there is greater uncertainty for high-income individuals, classification risk scores can shift audit allocations to lower-income individuals simply because misreports are easier to predict. Exploring the role of heterogeneity in uncertainty and its fairness implications might explain a wide range of other policies that have disparate impact (e.g., enforcement against blue collar vs. white collar crime). In the tax context, this insight also underscores the need for information collection mechanisms (e.g., third party reporting by offshore financial institutions) to reduce such uncertainty in the high income space, which has been the subject of significant policy debate [74, 189].

We conclude by noting several limitations and opportunities for further work. First, we do not have access to the exact models employed by the IRS or the complete procedures, so we cannot make definitive inferences about past or current practice. Second, we only observe (an imperfect proxy of) the IRS cost of an audit, not taxpayer costs; the true societal cost of an audit may thus be materially different than what is used in Section 5.8. Third, our approach has not distinguished between underreporting from misreported income versus over-claimed refundable credits; some policymakers may view these forms of noncompliance differently. Finally, while the notion of monotonicity is motivated in part by the near-monotonicity of adjustments and the oracle results, it is not grounded in a full welfare analysis. Such an approach might take into account audit costs to taxpayers, deterrence effects, and other policy levers, such as tax rates or penalty amounts. Accounting for these dimensions may not necessarily yield strict monotonicity as a form of vertical equity, and we view this theoretical development as an important path to refining vertical fairness.

Despite these limitations, this work represents an important step given the policy significance and complexity of this setting. The scale of the problem is substantial — amongst U.S. taxpayers alone, improvements in this area can affect more than 100M individuals annually. Moreover, "government by algorithm" continues to grow [79], and understanding how to incorporate fundamental fairness and redistribution concerns in taxation may serve as a model for other governance-related settings. Finally, insights derived in this setting — such as the differing effects of costs when considered as a constraint rather than in the objective — may carry over to other unrelated settings. Our finding that a narrow return-on-investment approach may degrade rather than improve vertical equity may be critical in a range of policy contexts [196]. Thus, both the technical concepts and policy problem are important and vital avenues for future research.

Chapter 6

Vertical Equity and Related Fairness Concepts

In this chapter, we delve in to slightly greater detail on the concept of vertical equity, and how it relates to other conceptualizations of fairness; namely, individual fairness and proportionality.

6.1 Vertical Equity and Horizontal Equity

Vertical equity is not so much a definition of fairness, but one framework for understanding equity in a given context. While we make no attempt to formalize vertical equity, as it is too general of a framework, vertical equity generally concerns appropriately accounting for relevant differences across individuals or groups, in other words, the "unequal, but equitable, treatment of unequals" [178]. In order to turn vertical equity into a useful set principles to guide equitable distribution of a burden, good, or service, assumptions must be made about the given context: what are relevant differences? What is an "appropriate" treatment of these differences?

Historically, vertical equity is strongly tied to the context of taxation, and is often understood as a foundational principle for building an equitable tax system[77]. Vertical equity is often cited as the ethical framework behind the progressive nature of the US income tax system: individuals with higher income pay a higher percentage tax. A simplified version of a common set of principles used to justify this tax allocation with vertical equity is that (a) the main relevant difference to account for across individuals when determining income tax is income[1] (b) the marginal utility of income is monotonically decreasing in income; and (c) the burden of taxation should be equally distributed across the population[2] and/or social welfare should be maximized[182]. Progressive taxation is often argued to be a consequence of such assumptions,

[1] The meaning of income is complicated, as tax scholars are cognizant of the many axes over which tax filers with the same income may not be in the same financial situation: for example, single versus joint filers, or individuals with and without children; etc. [18] We do not attempt to delve in to these complexities in this discussion of VE, but assume the relevant income referenced here takes these differences into account.

[2] This is often referred to as the ability-to-pay principle [18].

as well as other formulations of welfare maximization [182] or as a result of axiomatic social marginal welfare weights [217].

Vertical equity is often compared to *horizontal equity*, a framework which prioritizes the equal treatment of equals[77]. Again, horizontal equity requires several assumptions before it can be understood in a given context: what comprises equal treatment, and how can we determine when individuals or groups are equal? In tax policy, horizontal and vertical equity are seen as related, but separate, goals [170]. Under certain assumptions in the taxation context, horizontal equity is often viewed as ensuring that individuals with the same ability to pay experience the same amount of tax burden—however, individuals with the same ability to pay may have different incomes, but with different life circumstances, e.g. a married couple filing jointly with a child, versus a single filer [18]. In the tax audit context, horizontal equity can be instantiated as, for example, ensuring that irrelevant factors, such as race, does not impact audit probability.

We argue that most common definitions of fairness can be understood under the umbrella of horizontal equity. In particular, group fairness measures quantify whether some relevant prediction quantity (overall rate of predicting a particular label, true positive rate, etc.) differs by protected feature. The protected feature is generally assumed to be irrelevant to the task (except insofar as it may be correlated with relevant features); violating group fairness measures would suggest, then, that similar individuals are not being treated similarly. Individual fairness, as we discuss below, seems to instantiate the concept of horizontal equity on the level of individuals: similar people should be treated similarly according to a given metric.

Once notions of vertical equity and horizontal equity are contextualized in a given application, then they can be compared, and can overlap. In Chapter 5, we apply vertical equity to the *tax audit* context. Based on our investigation of noncompliance data, our assumptions here were that (a) again, the main relevant difference to account for across individuals when determining income tax is income; (b) the audit rate should be monotonic in income; and (c) the allocation of audits should closely match that of an *oracle model*, with perfect knowledge of noncompliance, which selects in order of greatest non-compliance. We consider several implementations of horizontal equity, including equalized odds and equalized true positive rates across income groups.

We show that under certain (differing) conditions, equalized odds and equalized true positive rates (instantiations of horizontal equity) will lead to monotonicity in audit rate across income groups (one component of our implementation of vertical equity). We present the theorem statement and proof in Appendix D.8. However, as later demonstrate, even if the instantiations of these frameworks in a given context overlap, achieving vertical equity via techniques aimed at establishing horizontal equity are not always optimal.

If both vertical and horizontal equity are instantiated with the same set of assumptions (i.e., relevant quantities to the decision and notion of similarity or difference), it is possible that vertical equity may imply horizontal equity, or vice versa [170]. However, part of the utility of vertical and horizontal equity as frameworks is that in practice, they *invite* different instantiations of these missing pieces, as their central concerns vary. An audit allocation can be monotonic and maximize social welfare from the perspective of ability-to-pay, but still over-allocate audits to one demographic group. While the perfect ideal of a similarity metric or notion of appropriate different treatment and relevant attributes may roll of these concerns into one, in practice, it may be impossible to find such a set of perfect definitions—so to approach an optimal-as-possible allocation of burden, goods, or services, it can be helpful to use both of these ethical frameworks to create a set of desiderata for allocation system function. In what follows, we compare the concept of vertical equity with common conceptualizations of fairness.

6.2 Vertical Equity and Individual Fairness

Individual fairness is concerned with *treating similar people similarly* [72], and thus, is much more strongly tied to *horizontal equity* than vertical equity. At a high level, both vertical and horizontal equity are more general notions than individual fairness, as horizontal and vertical equity are not specific to acting on the level of individuals or groups, whereas individual fairness, as evident from its name, only focuses on the treatment of individual people.

Individual fairness [72] is defined as a Lipschitz constraint on a model, given similarity metrics over model inputs and outputs. More formally,

Definition 6.1. *Individual Fairness [72]: Given a distribution of individuals V, a similarity metrics between individuals $d : V \times V \rightarrow \mathbb{R}$, a set of possible outcomes A, and a model which maps individuals to distributions of outcomes $M : V \rightarrow \delta(A)$, and a distance metric D over distributions of outcomes, a model M satisfies individual fairness if:*

$$D(Mx, My) \leq d(x, y)$$

As mentioned above, we do not attempt to give a formal definition of vertical or horizontal equity, as it is largely used more as a framework for understanding equity, rather than a definition of fairness in and of themselves. However, individual fairness can be seen as a formalization of the concept of horizontal equity on the level of individuals.

Still, even in the abstract, it is clear that *vertical* equity and individual fairness are not interchangeable. As a quick example of how these two concepts differ, consider an allocation where everyone gets an equal share—for example, everyone has the same tax rate. This would satisfy individual fairness, as any two individuals deemed the same by the given metric (e.g. have the same income) would be treated equally—more formally, $D(Mx, My) = 0 \leq d(x, y) \forall x, y$. However, vertical equity would *not* be satisfied if there exists any meaningful difference between individuals, as these individuals would receive the same treatment.

6.3 Vertical Equity and Proportionality

The concept of proportionality is treating individuals “in relation to their due”, as opposed to completely equally [98]. Proportional equality only recommends equal allocations when the individuals at hand are indistinguishable with respect to the allocation task—for example, equal work in a group project deserves equal credit. Thus, proportionality has to to with the equitable but unequal treatment of unequals, and is related to vertical equity.

These notions have a slightly different valence in practice, however. Proportionality is more often tied with an *individualized* concept of the concept of the unequal, but equitable, treatment of unequals; perhaps even agnostic to a greater social context [102]. Proportionality is often described as reaping what one sows, or getting out what one puts in: for example, in a group project, credit being shared according to effort spent or amount contributed [102]. Historically, vertical equity has been tied to finding an *societally* optimal allocation of a burden or service/good, typically of tax burden [182, 217]. This societal optimality is often framed as maximizing societal welfare or minimizing societal harm, which may lead to different allocations than common conceptions of proportionality.

Part III

Policy and Legal Repercussions of Flexibility in Modeling Decisions

Chapter 7

Model Multiplicity: Opportunities, Concerns, and Solutions

In this chapter, we consider the legal repercussions of the modeling flexibility afforded by the machine learning pipeline. In practice, for a given prediction task–e.g. predicting probability of loan default—there will not exist *one* singularly most accurate model, but a set of models with equivalent accuracy, made with different choices along the modeling pipeline. While these models may look similar from an aggregate accuracy standpoint, as we have shown throughout this thesis, there are often other behavioral differences between equally accurate models–for example, they may differ in their individual predictions, as shown in Chapter 2, differing explanations, as shown in Chapter 4, or even different fairness properties, as shown by prior work [214].

This fact affords model makers the flexibility to prioritize other values, such as fairness, in their model's behavior, at no cost to accuracy. In fact, we argue that the existence of this set of equally accurate models—which we call *model multiplicity*—leads to legal pressure to choose models which are least discriminatory among the set of equally accurate models. Symmetrically, however, this vast choice among potential models leads to a question of how to justify the decisions of the eventual model selected. We suggest that, in order to take advantage of the flexibility that model multiplicity provides while preventing its downsides, modeling choices should be carefully documented and justified throughout the model creation process.[1]

7.1 Introduction

How do model developers select which model to deploy for a given prediction task? Even after a model developer translates a decision into a prediction task (e.g. casting the task of determining who is "creditworthy" as predicting whether applicants are likely to default

[1] We note that, for the sake of ease of our formal definitions and exploration of model multiplicity, we assume that the prediction task for a given model is set, we suggest that the ideas of the repercussions and implications of model multiplicity extend to decisions over how to translate a policy or other application problem into an optimization task [197].

on a loan), there are myriad decisions made about how to make a model, all of which may influence its ultimate behavior: what model type should be used (from simple linear models to more complex random forests and neural networks); what factors should be considered as inputs to the model; how many times should the model iterate through the training data to learn the patterns therein? How should model developers select between all the possible models that could have been created for a prediction task?

The standard answer to this question is to choose the model that maximizes accuracy. Using maximum accuracy as a decision criterion for model selection may suggest that there is *one* model with the best accuracy, a common assumption in the technical literature. However, recent work has reminded us that there are usually multiple models with equivalent accuracy but significantly different properties. For example, Rodolfa et al. [214] have demonstrated that maximally accurate models can produce varying degrees of demographic disparities; D'Amour et al. [53] have shown that models with the same accuracy can be more or less robust; Chen et al. [41] have shown that it is possible to create interpretable models with the same accuracy as neural networks.

We call this phenomenon *model multiplicity*: when models with equivalent accuracy for a certain prediction task differ in terms of their internals—which determine a model's decision process—and their predictions. The existence of model multiplicity presents exciting opportunities because it offers model developers the flexibility to prioritize, and optimize for, desirable properties at no cost to accuracy, contrary to some conventional wisdom [20, 42, 68, 174, 239, 248, 265]. The existence of equally accurate models that differ along other axes, including fairness, interpretability, and robustness, allows for model selection to be guided by these other desiderata alongside accuracy. For example, as much recent work in algorithmic fairness has demonstrated, it is often possible to improve the fairness of models with no cost to accuracy [48, 68, 214, 253]. Model multiplicity can also improve individual experiences with automated decision making by allowing model developers to create models that make recourse easier (e.g., by limiting the use of features to only those over which individuals have control). While the freedom that model multiplicity affords is broad, in this chapter we largely focus on its implications with respect to the fairness of a model and the ability for people subject to the model to seek recourse. We also show that model multiplicity has legal implications—which we study in the context of lending—because it places pressure on model developers to search for and adopt the least discriminatory model among those that are equally accurate.

However, along with these benefits comes a potentially surprising revelation: given that there are multiple models for a prediction task with equivalent accuracy, selecting models on the basis of accuracy alone—the default procedure in many deployment scenarios—does not lead to a selection of one unique model best suited for the task. Model selection on the basis of accuracy alone is an underspecified [53] selection process. Unless other considerations are explicitly incorporated into the model development process, model developers selecting models on the basis of accuracy are unlikely to happen upon the model, among all those which are equally accurate, that best addresses those considerations (e.g., minimizes disparate impact).

Further, model multiplicity undermines the justification that we can offer individuals for being subject to any adverse decision process or outcome. Consider the situation where an individual is denied a loan, yet there exists an equally accurate model which would have recommended acceptance. Why must they be subject to the model that rejected them and not an equally accurate, and thus equally viable, one that does not? The fact that such high-stakes decisions may come down to arbitrary choices on the part of model developers may be quite unsettling—and may even conflict with the expectations of the laws that govern such decision making. Thus, while model multiplicity allows for greater choice in the model selection process, it also imposes an additional burden on model developers to put that freedom of choice to good use and to justify how they reach their decisions.

In this chapter, we attempt to answer the following question: *how do we take advantage of model multiplicity while addressing its concerning implications?* To do so, we propose a process by which model developers can specify, justify, and document a wider set of behaviors which qualify a model for use in a specific context to guide the model selection process. Concretely, we present three main contributions: (1) a principled understanding of the relationships between multiplicity, accuracy, and variance, providing intuition for why multiplicity may actually *increase* with accuracy, backed by theoretical results deferred to Appendix E.1; (2) connections between the technical aspects of model multiplicity and their legal implications; and (3) a set of policy recommendations for how to take advantage of model multiplicity while addressing the concerns it raises. Ultimately, we hope that the explicit recognition of model multiplicity, along with legal requirements preventing discrimination, will lead policy makers and model developers to hold models to a higher standard on axes beyond accuracy—and restore the justifiability of model decisions.

The rest of this chapter proceeds as follows: in Section 7.2, we provide an overview of model multiplicity and situate it in the existing literature. Section 7.3 explores the relationship between multiplicity and accuracy, connecting multiplicity to standard ideas from machine learning theory. Sections 7.4 and 7.5 articulate the potential benefits and harms respectively of model multiplicity, drawing connections to the law. In Section 7.6, we provide recommendations for a model development process that explicitly accounts for multiplicity.

7.2 Defining Multiplicity

Model multiplicity occurs when models with equivalent accuracy for a certain prediction task differ in terms of their internals or their predictions. In this section, we define model multiplicity in more detail, beginning by describing the setting in which we consider model multiplicity, our definition of model accuracy, key terms for the chapter, and, finally, the definitions of the components of model multiplicity: *procedural* and *predictive multiplicity* .

7.2.1 Preliminary Definitions

Setting In this chapter, we focus on classification models, although the main insights of this work apply to the regression setting as well. A classification model predicts to which class, or category, an input x belongs, from some pre-set collection of categories. For example, predicting whether an individual will default on their loan is a classification task. Classification models have *decision surfaces* which delineate between different classes in the model's input space (see Figure 7.1). We will focus on the case where there are only two classes (also known as binary classification).

Accuracy Broadly, accuracy is a measure of how well a model's predictions match the underlying labels in the data. Importantly, model developers cannot know how accurate the

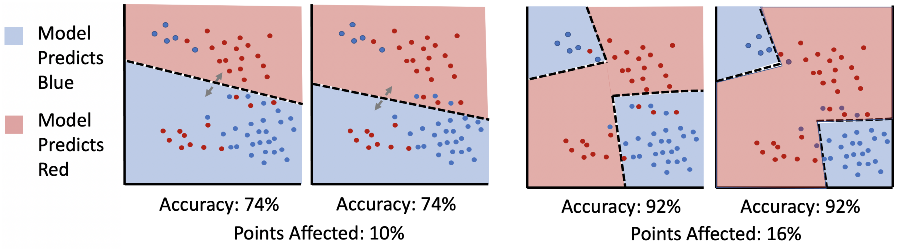


Figure 7.1: A stylized graphic displaying how models with higher accuracy can actually lead to *more* model multiplicity: On the left, we show a simple linear model on some data, with accuracy of approximately 75%. On the right, we show a more complex model which fits the data better and reaches 92% accuracy. In order to achieve a better fit to the data, the more complex model has a more complex decision surface. In having a more complex decision surface, there are more opportunities for shifts in decision surface to take place in reaction to changes in the training process, and thus there are more points in the distribution that are susceptible to a change in prediction.

model is on all possible model inputs (e.g., over all possible loan applicants); accuracy must be estimated on available data. In practice, there are a variety of measures of accuracy; for simplicity, we will take accuracy to mean the fraction of predictions for which the model is correct. When we refer to several models exhibiting equivalent accuracy, they may not have exactly the same accuracy, but accuracy that is functionally indistinguishable (e.g., an accuracy of 97.8989 and 97.8990).[2]

7.2.2 Procedural and Predictive Multiplicity

Model multiplicity describes how models for a given prediction task can differ even when they exhibit equal accuracy. We draw attention to two ways in which models can differ despite being equally accurate: in their internals, or *procedural multiplicity*, and in their predictions, or *predictive multiplicity.*

Procedural Multiplicity Procedural multiplicity refers to the phenomenon where several models for a given prediction task have equivalent accuracy, yet differ in their model internals. More technically, procedural multiplicity occurs when models which have the same accuracy exhibit some difference in their decision surface, as this changes the way in which a model's inputs are combined to reach a conclusion. In other words, procedural multiplicity describes the situation where models of equal accuracy differ in the *process* by which they reach a given prediction. One example of a difference in the model's internals is the use of various input *features* into a model's decision for a given prediction: for example, one model may use gender as a feature to make loan granting decisions; another may not. Another example of a difference in model internals is a difference in *model class.* For example, a random forest model and a linear model may have equivalent accuracy for a certain task, but likely vary in the way they reach each prediction. One way procedural multiplicity can become apparent to model subjects is when equally accurate models produce qualitatively different explanations for the same decision. In one example from Anders et al., two credit scoring models make the exact same predictions on every point, but one model justifies its decision on the basis of gender, while the other relies on income and tax payments [9].

[2] We note that what levels of accuracy are functionally indistinguishable may depend on the context in which the model is used.

Predictive Multiplicity Predictive multiplicity refers to the phenomenon where models with equivalent accuracy for a certain task differ in their predictions (i.e., two models predict different classes for the same input). Predictive multiplicity, like accuracy, is measured on the labeled data available to a model developer: given a prediction task, models that exhibit predictive multiplicity have equal accuracy but predict different classes for some data points in the training or test set.[3] Thus, model developers cannot measure the full extent of disagreement between any two models, but can only estimate it based on available data.

Relationship Between Procedural and Predictive Multiplicity Note that differences in model predictions on a certain input require differences in the decision surface, implying predictive multiplicity is a special case of procedural multiplicity. The converse does not hold: two models with the same prediction on a given point may still exhibit variation in the process by which that outcome was reached [9, 25]. However, we draw attention to predictive multiplicity on its own due to its unique normative and legal implications. Throughout this chapter, when we refer to procedural multiplicity, we refer to the aspects of procedural multiplicity that occur even in the absence of (observed) predictive multiplicity: that is, models with equal accuracy with different decision processes that do not necessarily manifest in different predictions on the available data. Of course, any change to a model's decision surface, and thus any two models exhibiting procedural multiplicity, will differ on *some* potential input point; but if no such input is present in the data, then this difference will not result in observable predictive multiplicity.

7.2.3 Sources of Multiplicity

When creating a model for a given learning problem, every decision point a model developer faces along the model building pipeline serves as a fork, where each potential choice may lead to multiplicitous models. In the context of this chapter, we define a learning problem to be the prediction of a pre-defined target. While there may be further multiplicity-like problems stemming from the various ways that a nebulous real-world goal may be translated into predicting a specific target [197], we view these as out of scope for this chapter. However, all modeling decisions made once the prediction target is set are within-scope and possible sources of model multiplicity.

Decisions that can result in multiplicity include choosing what features should be included as input to the model [63], which points are included in the training set [24], which model class should be adopted [41], what random numbers the model's parameters are initialized with [24, 172], among many others [168]. Through these choices, the model developer creates one model, but each other choice they could have taken may have lead to a model that would have performed with similar accuracy. In theory, the sources of multiplicity are infinite, as there are infinite possible modeling choices. In practice, however, the range of choices is restricted by practical (including budgetary) constraints.

7.2.4 Aggregate and Individual Effects

Model multiplicity can result in differences between models at the *aggregate* level or at the *individual* level. By aggregate effects, we refer to differences in global model properties

[3] There is disagreement in the literature on this definition: for example, Marx et al. [168] define predictive multiplicity only on a model's training set.

between multiplicitous models (e.g., satisfaction of group-level fairness criteria (such as equal selection rates across different demographic groups)). By individual effects, we refer to the way in which differences between models of equal accuracy impact individuals' experience with the model, including differences in individual predictions or explanations of those predictions. Aggregate and individual effects are not disjoint categories of model behavior, as some forms of model multiplicity may impact both aggregate-level and individual-level outcomes. Often, however, individual effects do not manifest at the aggregate level. For example, differences in individual predictions may not impact the overall treatment of any demographic group. We therefore find that making these two perspectives explicit helps to better understand the overall impacts of multiplicity.

7.2.5 Arbitrariness Versus Randomness

In this work, we draw a distinction between arbitrariness and randomness in selection processes. By an arbitrary selection process, we mean a completely unconsidered decision—one that is made without thought or perhaps even without knowledge that a choice was being made. By a random selection process, we mean a decision which is *purposefully* left to chance. We draw this distinction to stress that a random selection process is predicated on a conscious choice to employ this selection method: as Perry and Zarsky [204] write, "the decision to opt for chance must be reasoned."[4]

7.2.6 Related Work

Model multiplicity has been recognized in the machine learning literature, though not always under the same name, starting with Breiamn's characterization of the "Rashomon Effect" [33]. For example, Dong and Rudin [63] and Fisher et al. [87] demonstrate procedural multiplicity in feature importance, showing that models relying on different sets of features can reach the same accuracy; Black et al. [25] and Mehrer et al. [172] have shown that deep models with similar accuracies relying on the same features may still combine those features in different ways to reach a given output. Recent studies also provide evidence for predictive multiplicity: Marx et al. [168], who introduced the term, focus on its effects at the individual level (equally accurate models can make different predictions for individuals), while others have demonstrated its effects at the aggregate level (equally accurate models can have different properties, including fairness and robustness) [53, 215]. A recent line of work has sought to quantify and mitigate model multiplicity in a variety of settings [24, 25, 26, 49, 200, 212, 221]. Our work builds upon and synthesizes this technical foundation to understand the relationship between model multiplicity, complexity, and error, as discussed in Section 7.3, and to relate the wide range of effects of model multiplicity to the law.

Some legal scholars have also begun to consider the possibility of model multiplicity and its implications, though this discussion is largely focused around predictive, and not procedural, multiplicity. Kim [137] has argued that predictive multiplicity means that certain interventions aimed at reducing disparate impact "do not require special legal justification" as the lack of one "correct" model means that there is "no clear baseline" against which any departures might be challenged. Kim points out that it simply does not make sense to say that someone has been unfairly denied a job that they would have otherwise secured if not for the attempt to reduce

[4] As Perry and Zarsky [204] describe in their work, there are many situations where random (not arbitrary) selection is justifiable: for example, allocating a scarce, indivisible resource among many with equally strong claims—such as allocating public housing among equally needy applicants.

disparate impact because there is nothing that entitles anyone to having a particular model chosen over an equally accurate alternative. On this account, multiplicity provides developers with the freedom to choose the model among those with equal accuracy that exhibits the least disparate impact without raising concerns with disparate treatment. However, this work does not address the concerns that model multiplicity may raise. Creel and Hellman [51] briefly note the unsettling implications of predictive multiplicity with respect to arbitrariness in algorithmic decision making, but ultimately argue that arbitrariness is only a problem in algorithmic decision making when there is an algorithmic monoculture that locks an individual out from certain opportunities across the board (e.g., when all lenders use the same algorithm and thus all reach similarly adverse decisions for a particular individual). Contra Creel and Hellman [51], many legal scholars have been calling for legal protections, inspired by due process principles and practices, to address the potential arbitrariness of algorithmic decision making more generally, even in the absence of an algorithmic monoculture [44, 45, 50]. In contrast to prior work, we address both the benefits and the concerns of procedural and predictive multiplicity, and we provide concrete recommendations for how to take advantage of the benefits of model multiplicity in practice, without falling prey to the concerns that it might provoke.

7.3 Accuracy and Model Multiplicity

By default, accuracy is the primary measure by which machine learning systems are evaluated. This focus is pervasive throughout machine learning scholarship and practice [23], perhaps best evidenced by the Common Task Framework [66], through which independent researchers compare predictive performance on common datasets. But accuracy plays a larger role in model development than evaluation alone: accuracy is typically the main or sole criterion used for model *selection*. When deciding which of many possible models to deploy, a practitioner will often choose the most accurate one.

The idea that model selection can be reduced to accuracy-maximization rests on a pair of premises: that accuracy is the primary measure of how "good" a model is, and that, for a given task, models that maximize accuracy do not differ meaningfully from one another. In other words, if accuracy-maximization leads to a unique or near-unique optimal model, then no other criteria need be used in model selection. Even if we accept that accuracy should be the primary evaluation criterion (setting aside, for now, properties like fairness, robustness, and interpretability that might be perceived as crucial to model performance in practice), evidence suggests that accuracy-maximizing models are not unique [24, 25, 26, 53, 168, 200, 215]. And yet, the intuition that there exists a unique "correct" model, and that accuracy-maximization should ultimately discover it, remains pervasive [137, 153, 213]. In what follows, we trace the roots of this intuition and offer a theoretical basis for why, as machine learning becomes more sophisticated, we should expect accuracy-maximization to yield *more* multiplicity, rather than less. As a result, accuracy is an incomplete basis for model selection. We focus here on predictive multiplicity, though it may be possible to derive analogous results for procedural multiplicity as well.

Does accuracy-maximization reduce predictive multiplicity? Our intuition that accuracy-maximization should lead to little or no predictive multiplicity comes from the idea that there exists a single "best" or "correct" predictor (known as the Bayes optimal predictor [222]), and increasingly sophisticated models will converge to this optimal predictor.

In general, Bayes optimal models are unique, and it may be tempting to apply this intuition more broadly: we might believe that even when our models aren't Bayes-optimal, the maximally accurate model for a given dataset is near-unique. We can make this idea rigorous: Theorem E.2, included in the appendix, demonstrates as the error of a model approaches that of the Bayes optimal predictor, the model must approach the Bayes optimal predictor.[5] In other words, as models get more accurate, they must converge to one another in the limit. Results like these can lead to a slippage in intuition: Bayes optimal predictors are unique, so the best predictor we can build should also be unique. And yet, empirical evidence seems to suggest the opposite: developing more accurate models can often lead to *more* multiplicity (see Figure 7.1 for an example) [24, 172]. While this might appear to contradict Theorem E.2, in reality, models are sufficiently far from Bayes optimal, leaving plenty of room for multiplicity. To derive a more nuanced view, we turn to standard bias-variance decompositions of error.

Bias, variance, and multiplicity. Conceptually, errors in machine learning systems come from three sources: bias, variance, and irreducible noise [62, 94]. This decomposition helps us understand fundamental trade-offs in machine learning: more expressive and sophisticated machine learning techniques (such as deep learning) have less bias because the average model can more accurately approximate the Bayes optimal predictor than less expressive techniques (such as linear regression); but this increased expressivity comes at the cost of high variance, since any particular model is much more sensitive to random choices in the model development pipeline. Crucially, as Theorem E.3 shows, multiplicity is tightly related to variance. To the extent that increased accuracy is achieved through increased model complexity (and therefore variance), we should therefore expect to see *more* predictive multiplicity, as noted in Corollary E.3.1. Thus, accuracy is not an antidote to multiplicity, and model selection cannot simply be reduced to accuracy-maximization. Instead, we must explicitly consider and deal with multiplicity, beginning with an understanding of the benefits and challenges it brings.

7.4 Opportunities

By shattering the intuition that there is *one* most accurate—and therefore correct—model, multiplicity can introduce much more freedom into the model selection process. On the aggregate level, this means that model developers can express preferences over values beyond accuracy *at no cost to accuracy*, including with respect to properties like fairness, robustness, and interpretability, among others. This same flexibility manifests on an individual level: to illustrate this point, we focus on the ability to improve the recourse available to the individuals subject to a model's adverse decisions. This section will consider the benefits at both levels.

7.4.1 Aggregate Benefits: Flexibility

By demonstrating that there are many different ways of making equally accurate predictions, multiplicity gives model developers the flexibility to prioritize other values in their model selection process without having to abandon their commitment to maximizing accuracy. While this benefit is broad, we focus in particular on its implications for fairness. In fact, as we'll discuss in this section, the flexibility afforded by multiplicity is particularly relevant to

[5] This result holds as long as data points are more predictable than 50-50 coin flips.

the law because it creates legal pressure for model developers to reduce avoidable disparate impact in their deployed models. We demonstrate this flexibility—and its connections to the law—through both procedural and predictive multiplicity.

Procedural Multiplicity Model developers can leverage procedural multiplicity to ensure that a model has desirable model internals without sacrificing accuracy. As shown in prior work, model developers might exploit procedural multiplicity to select a model class that is more robust or interpretable than other model classes of equal accuracy [53, 216]. This is far from an exhaustive list, as procedural multiplicity creates the possibility for *any* quality of a model's decision process to be prioritized at potentially no cost to accuracy. However, in the context of fairness, the possibility that replacing or removing certain features from a model may not affect its accuracy is particularly relevant. If there are certain features that are perceived as a normatively objectionable basis for decision making, procedural multiplicity suggests—and research has demonstrated empirically [27, 63]—that model developers can remove these from their models while still potentially achieving the same level of accuracy in their predictions. For example, features may be normatively objectionable because they are legally protected characteristics such as race or sex or because they are proxies for such characteristics, such as zip code. Discrimination law imposes exactly these kinds of constraints on model developers in certain regulated domains via a prohibition on so-called "disparate treatment." For example, the Equal Credit Opportunity Act (ECOA) prohibits the consideration of race, sex, age, and a number of other legally protected characteristics in lending decisions [2, 80]. Thus, lenders using machine learning to develop credit scoring models are understood to be legally prohibited from including these features in their models. While this prohibition is designed to prevent lenders from relying on features that have served as the basis for discriminatory decision making in the past, it is also designed to encourage lenders to find other features that serve their goals at least as well. Procedural multiplicity demonstrates that it may be technically possible to do so, putting to bed the idea that there is only ever one set of features that would allow model developers to achieve some level of accuracy in their decision making.

Predictive Multiplicity While procedural multiplicity gives model developers the flexibility to incorporate their normative preferences into the model's decision-making process, predictive multiplicity allows model developers to impose their preferences on the model's predictions—potentially without impacting accuracy. In the context of fairness, predictive multiplicity creates the possibility to minimize differences in prediction-based metrics across groups, notably differential validity (i.e., differences in the accuracy of the predictions) and disparate impact (i.e., differences in the predictions themselves). Rodolfa et al. [215] show that this is possible in practice across a wide range of real-world applications, including such high-stakes domains as criminal justice, housing, and education. Similarly, algorithmic hiring companies such as HireVue require that their models return similar distributions of predictions across demographic groups, and claim that this has little impact on predictive accuracy [147].

This aspect of predictive multiplicity speaks directly to the disparate impact doctrine in discrimination law, which imposes liability on model developers for avoidable disparities in the rate at which members of legally protected groups obtain the desired outcome from a decision-making process. As the official commentary on ECOA states, the law "prohibit[s] a creditor practice that is discriminatory in effect because it has a disproportionately negative impact on a prohibited basis, even though the creditor has no intent to discriminate and the practice appears neutral on its face, unless the creditor practice meets a legitimate business need that cannot reasonably be achieved as well by means that are less disparate in their impact" [2].[6] To appreciate what this means in the context of a lender employing machine learning, imagine that the "creditor practice" in question is the use of a machine learning model developed to predict default and that the lender's primary "business need" is predicting default as accurately as possible so as to make appropriate lending decisions. The official commentary suggests that even when machine learning has been adopted for this purpose, involves no legally proscribed features, and demonstrates a high degree of accuracy, lenders still face liability if they fail to adopt whatever alternative "means" might exist for achieving their same goal but with smaller disparities in outcomes across legally protected groups. Predictive multiplicity suggests that there may exist such alternative "means" because there may be a different model of equivalent accuracy that generates less disparate impact. The disparate impact doctrine can thus be interpreted to say that predictive multiplicity creates legal risk for those who fail to adopt the least discriminatory model among those that are equally accurate [209, 210].

7.4.2 Individual Benefits: Improved Possibilities for Recourse

Multiplicity can also provide the flexibility necessary to improve individuals' experience of model procedures and their outcomes. To illustrate this point in the context of fairness, we explain how procedural and predictive multiplicity can improve an individual's capacity to achieve recourse—that is, to obtain a more desirable outcome after receiving an adverse decision.

Procedural Multiplicity Recent scholarship has suggested that one of the important functions of providing explanations of model decisions is to help people subject to an adverse decision understand how they might obtain a more favorable prediction in the future [247]. In the United States, the Fair Credit Reporting Act (FCRA) and ECOA both require that lenders explain their decisions to consumers who were unsuccessful in their applications for credit [80, 188]. Both laws compel lenders to provide so-called "adverse action notices" that state the "principle reasons" for adverse decisions, on the belief that doing so may help consumers more effectively navigate the process of obtaining credit in the future [16, 220]. Scholars have suggested that lenders might comply with these requirements by offering counterfactual explanations that point out, for example, that an applicant would have been successful if their annual income had been $10,000 higher [241]. In light of such an explanation, the consumer might look for ways to increase their income and then reapply for a loan. However, as prior work has pointed out, such explanations may not facilitate recourse if the highlighted factors are immutable and thus cannot be acted upon by the consumer [241]. Explanations only facilitate recourse if they suggest changes to features that consumers can actually execute in practice. This insight has motivated a good deal of recent research focused on developing methods to produce explanations that suggest viable and efficient paths to future success [130]. Procedural multiplicity suggests that there is another—and more direct—way to achieve these same goals. Rather than searching for different possible explanations of the decisions of a fixed model that would be easiest for consumers to act upon (the current focus in the literature on recourse), model developers could exploit procedural multiplicity to find models that exhibit the same degree of accuracy but differ in the degree to which they rely on features known to be difficult or impossible for people to change. Thus,

[6] While disparate impact is not written directly into ECOA, the formal guidance suggests that it is understood to apply under the law.

procedural multiplicity gives model developers a way to take recourse into account in the model development process, not just in deciding which techniques to rely on when explaining a model's decisions.

Predictive Multiplicity As algorithms have been adopted in a growing range of high-stakes decisions, scholars have begun to worry about the possible harms of an *algorithmic monoculture* [51, 138]. For example, if several lenders all converge on one credit scoring model (and thus on the same predictions of default for each applicant), consumers who were rejected by one lender may find that they have no better luck when they submit an application to other lenders. This, too, is a problem of recourse, but at the level of an entire domain of decision making, rather than at the level of a model. Predictive multiplicity may serve as a natural bulwark against this worrisome possibility: even if lenders all maximize prediction accuracy, they may still end up with models that produce different individual-level predictions. The perhaps surprising benefit of predictive multiplicity is that, even when models are selected on the basis of accuracy alone, there will be inherent heterogeneity in the models selected by different firms [51].

7.5 Concerns

While procedural and predictive multiplicity gives us the flexibility to prioritize values beyond accuracy, this very same flexibility can be cause for serious concern. The fact that we can choose among many possible models with equivalent accuracy can lead to problems of underspecification and to arbitrariness in decision making. Selecting models on the basis of accuracy alone can obfuscate large differences between multiplicitous models that we might actually care about, but have failed to explicitly integrate into the set of considerations that go into the model development process. Perhaps even more importantly, model multiplicity also means that accuracy alone is an insufficient justification for why one model was chosen over another equally viable (i.e., accurate) alternative. In this section, we consider the concerning implications of model multiplicity and how the law bears on some of these concerns.

7.5.1 Aggregate Concerns: Underspecification

As we've shown, model multiplicity gives model developers the option to prioritize values beyond accuracy, since models with equal accuracy can have quite different aggregate- and individual-level effects. This also means, however, that failing to consider what other behaviors may be desired, and continuing to choose models on the basis of accuracy alone, leaves model behavior on axes other than accuracy up to an arbitrary choice: without explicitly specifying what behaviors a model should exhibit—such as fairness, robustness, and interpretability—and optimizing for them, it is unlikely that a model will naturally exhibit such behaviors. D'Amour et al. [53] call this the problem of underspecification.[7] Underspecification reveals that we need to make our desired model properties explicit if we want our models to exhibit them.

[7] We note that there is a subtle difference between *underspecification*, where a model developer fails to fully articulate and incorporate their full set of behavioral desiderata into the model building process (as D'Amour et al. [53] show in the case of model robustness) and *mis-specification*, where a model developer chooses the wrong target to optimize for (as Obermeyer et al. [187] demonstrate in a healthcare system's choice to use healthcare costs as a proxy for healthcare needs).

Procedural Multiplicity Procedural multiplicity can give rise to three rather serious problems. First, as mentioned, selecting a model on the basis of accuracy does not guarantee that it will exhibit other desirable properties. Second, because it may be possible for models with different internals to still generate the same set of predictions, changes made to the internals of a model may not have the anticipated effect on predictions. Third, procedural multiplicity can be leveraged to remove anything from the decision-making process that would raise legal or normative concerns (e.g., legally protected or otherwise controversial features), while preserving a troubling, but avoidable, outcome (e.g., disparate impact). We focus on the second two concerns.

First, procedural multiplicity means that removing features proscribed by discrimination law may do nothing to reduce disparities in predictions, which may have been the explicit intent of such an intervention. As discussed earlier, discrimination law imposes strict prohibitions on the use of certain characteristics in decision making across a range of high-stakes domains, including lending. These prohibitions on disparate treatment were put in place to protect people who possess these characteristics from systematically worse treatment than others (hence the term “protected characteristics”). Procedural multiplicity undermines these protections because it opens up the possibility that people with these characteristics might be subject to the same disfavorable predictions without directly considering these characteristics [57]. While disparate impact doctrine has developed, in part, in recognition of the potentially limited efficacy on prohibitions on disparate treatment [15]—placing demands on decision makers to be able to justify disparities in model predictions, even if they haven’t considered any protected characteristics—calls for procedural interventions remain commonplace. For example, Black et al. [27] observe this phenomenon in debates about the design and use of risk assessment tools in the criminal justice system, where procedural interventions recommended by experts and advocates, such as removing nonviolent arrests from the criminal history considered by the tools, seem to be suggested with the expectation that they will reduce racial disparities in tools’ predictions. Procedural multiplicity means that there is no guarantee that these changes will have the desired effects on model predictions.

Second, given that there might be many ways to develop a model that generates the same predictions, developers could search for models that seem to be more palatable from a procedural perspective (e.g., because they don’t involve legally proscribed or otherwise controversial features) but display the same worrisome predictive behavior. Objecting to these predictions might be more difficult when the process that generates them seems benign or perhaps even desirable. This is not just a hypothetical concern; recent work has shown that it is possible to create two models with *exactly the same predictions* that rely on *completely different features to make up their decision* [9, 25]. This suggests that not only might procedural interventions fail to have their intended effects on predictions, but that procedural multiplicity can be exploited adversarially to develop a compelling justification for whatever disparities in predictions that model developers might like to preserve. While this possibility, often referred to as *proxy discrimination* [57], is well-studied in the literature, we note that it is a result of procedural multiplicity.

Taken together, these observations about procedural multiplicity highlight the need for model developers—or those seeking to influence or regulate their choices—to fully specify the kinds of predictions that they would like models to generate. Unless these are optimized for explicitly, there is no reason why maximizing accuracy or making procedural interventions will lead to the desired model behavior.

Predictive Multiplicity The reality of predictive multiplicity highlights that model selection on the basis of accuracy does not guarantee the desired prediction-based behaviors beyond accuracy. Specifically, in the context of fairness, predictive multiplicity tells us that there may be several equally accurate models that each vary in the degree to which accuracy, selection rates, or other fairness metrics differ across groups. Unless this is made an explicit consideration in the model development process, the chosen model can be an arbitrarily bad pick with respect to fairness metrics among those that are all equally accurate.

7.5.2 Individual Level Concerns: Loss of Justifiability

Model multiplicity also creates serious challenges for justifying the ultimate choice of model, given that different choices can result in more or less favorable situations and predictions for any given individual. Globally, this raises a fundamental question: what justification is there for subjecting a particular person to an adverse model procedure or model prediction if that person would have received more favorable treatment under a different, but equally accurate model? This section will consider the crisis of justifiability brought about by both procedural and predictive multiplicity and again discuss how the law bears on this challenge.

Procedural Multiplicity As discussed, procedural multiplicity admits the possibility of creating models with very different internals, even if they all exhibit the same degree of accuracy and all result in the same predictions. This increased flexibility, however, leads to a difficulty in justifying why a particular way of reaching the prediction is necessary. We again focus on the example of recourse: we previously suggested that predictive multiplicity is desirable when it allows developers to favor models with internals that would make recourse easier (e.g., selecting models with features that people would find less challenging to change). Yet, for any given individual, there might exist an alternative model with identical predictions that would have given the individual an easier path to recourse. Consider a scenario in which a lender offers an explanation for an adverse decision that an applicant for credit would find challenging to act on. Even if the applicant accepts that this is a valid explanation for their adverse prediction and the easiest of all possible explanations for the applicant to act on, the applicant might nevertheless ask: why did the lender choose the model that makes recourse more difficult for me instead of the model that would have made recourse easier for me, given that both would have resulted in the same predictions? The applicant might ask more generally: why must I be subject to this model rather than the other? Procedural multiplicity makes it challenging to answer these questions because accuracy alone cannot justify the ultimate choice of model.

Predictive Multiplicity Predictive multiplicity can be just as unsettling when it comes to the justifiability of decisions because individuals might receive favorable predictions under some models and unfavorable predictions under others, even when all of these models are equally accurate. To illustrate this point, consider a situation in which there are two models that exhibit the same accuracy, but only one of which would instruct a lender to grant an applicant's request for credit. If the lender happens to choose the one that denies the applicant's request, how would the lender justify its adverse decision, given that the lender could have just as easily chosen the other model? This line of questioning is unsettling because it reveals that choosing a model based on accuracy alone is akin to choosing arbitrarily between more or less favorable predictions for certain individuals.

The disquieting prospect that consumers' access to credit might rest on decisions made without adequate care was one of the main concerns that motivated the passage of FCRA and

ECOA, both of which target arbitrariness in lending decisions. The legislative record suggests the FCRA was designed to "protect consumers from inaccurate or arbitrary information in a consumer report which is being used as a factor in determining an individual's eligibility for credit, insurance, or employment" [1]. It seeks to do this by requiring that lenders adopt reasonable procedures to ensure the "accuracy, relevancy, and proper utilization" of the information in credit reports. In regulating the information that goes into high-stakes decision making, FCRA seems to be designed to guard against capricious, sloppy, and otherwise faulty decision making. As discussed earlier, ECOA requires lenders facing a disparate impact claim to demonstrate that "the creditor practice meets a legitimate business need"; in practice, this is often accomplished by demonstrating that their credit scoring models reasonably accurately predict default. In other words, absent some justification for assessing applicants for credit in a manner that generates a disparate impact, lenders will be found liable for discrimination. Finally, both FRCA and ECOA require that lenders provide adverse action notices, on the belief that having to justify their decisions will cause lenders to be less arbitrary in their decision making [220]. Note that lenders are only required to justify their particular way of making decisions when they face a disparate impact charge. Absent any identified disparate impact, FCRA and ECOA only require that lenders provide an explanation for any particular decision, not a justification for the manner in which they make decisions. Yet it is possible to interpret this more modest requirement as an *indirect* way of trying to ensure that there are good justifications for why lenders make decisions the way that they do. For example, if the proffered reason for an adverse decision is something that seems to lack face validity as a predictor of default, then consumers might question whether the basis for decision making is well justified (namely because it seems unlikely that predictions of default on that basis would be accurate) [220]. These laws are obviously both premised on the idea that there should be good reasons for the manner in which lenders go about making their highly-consequential decisions.[8] The problem with predictive multiplicity is that it makes avoiding arbitrariness difficult even when lenders seek maximally accurate predictions.

Accuracy has traditionally provided a justification for model selection because it was assumed that there must be one unique model of maximally achievable accuracy. If selecting on the basis of accuracy leaves model developers with only one choice, then, according to this thinking, the ultimate choice must be justified. Multiplicity reveals this assumption to be false. While we might welcome the fact that selecting models on the basis of accuracy does not limit developers' choices to just one option, we should also recognize the threat that it poses to the justifications that we can now offer for the ultimate choice of model. Just as selecting on the basis of accuracy does not entitle anyone to a specific prediction [137], selecting on the basis of accuracy need not condemn anyone to a specific prediction. Whatever the chosen model, there always exists an alternative model of equal accuracy that would reverse an individual's prediction.[9] And any given individual might ask: why was one model chosen over the other? Model multiplicity means that we have lost a fundamental basis for justification that needs to be replaced.

This is well reflected in the worries expressed by Citron and Pasquale [45] when they point to a "a study of 500,000 files [in which] 29% of consumers had credit scores that

[8] While Creel and Hellman [51] suggest that arbitrariness in decision making is only a problem when there is no alternative decision maker to whom a person can turn after receiving an adverse decision, these legal requirements seem to be designed to guard against arbitrariness in the decision making of private actors whether or not there are alternatives in the marketplace.

[9] In theory, such a model always exists; whether one could reasonably be found in practice depends on both the model developer's choices and the individual in question.

differed by at least 50 points between the three credit bureaus." They argue that "[b]arring some undisclosed, divergent aims of the bureaus, these variations suggest a substantial proportion of arbitrary assessments" [45]. If we assume that the three credit bureaus all have access to similar information, that they are all seeking to predict default, and that they each have the means to achieve similar accuracy in their predictions, then much of the resulting divergence in scores for particular individuals is likely the result of predictive multiplicity. Rather than accepting this as an unavoidable or even desirable effect of the heterogeneity naturally engendered by predictive multiplicity, Citron and Pasquale argue that the divergence is evidence of arbitrariness, on the likely belief that if the bureaus had good reasons for choosing their credit scoring models, the models would not return different predictions. Accuracy is no longer a sufficiently good reason because selecting models on that basis cannot supply one correct answer; there now remains an unaddressed degree of arbitrariness. In a perhaps surprising reversal, what we described earlier as a welcome guard against algorithmic monoculture is here presented as a threat to justifiability: why must any individual be subject to the chosen model when an equally accurate alternative exits that would have given the individual a more desirable prediction?

In order to recover the justifiability of model decisions, accuracy can no longer be used as the reason why a particular model was chosen in high-stakes applications. There must be additional criteria used to determine whether a model performs sufficiently well for high-stakes deployment, and why one model—and therefore its decision procedure and predictions—should have been chosen over an equally accurate alternative.

7.6 Solutions

The problems arising from model multiplicity underscore the need for a more careful model selection process that explicitly takes multiplicity into account. A core component of the risks imposed by model multiplicity is that there is no *thought* given to the selection of the model among apparently equally viable choices: the selection is arbitrary, as it occurs without admission or even knowledge that a choice is being made. In order to take advantage of model multiplicity while making justifiable model decisions, we must create a non-arbitrary method of choosing between high-accuracy models that specifies the behaviors we wish to see in the model, and documents the reasoning behind the choices made. Towards this goal, we can make explicit and justify a set of criteria for acceptable model properties or model behavior beyond accuracy alone, document these criteria and justifications, and then only consider models that meet these criteria. We call this set of criteria the *meta-rule*. As there may still be multiplicitous models that all satisfy the meta-rule, we suggest ways to further choose between models with differing individual predictions to prevent arbitrariness that satisfy a given meta-rule via various prediction aggregation techniques. Importantly, all of these choices—the meta-rule and the aggregation technique—must be justified and documented. This, ultimately, serves as a justification for why an individual is subject to a certain model decision.

7.6.1 Meta-Rules

Using a meta-rule provides a reasoned way to choose amongst multiplicitious models: the explicit consideration of what model behaviors make up the meta-rule may provide model developers greater clarity on how to optimize for these behaviors during the model building process, preventing issues of underspecification discussed in Section 7.5.1. For example, for a loan prediction model, a meta-rule may be: the model must have over 95% accuracy, rely on features only available in the individual's recent banking activity, and have near-equal true positive rates across demographic groups. Moreover, the documentation of these decisions and the reasoning behind them can serve as a justification for the model.

The restriction to only consider models that satisfy all criteria of a meta-rule reduces the set of multiplicitious models which all reach similar accuracy on a given prediction task to a smaller set—which, crucially, all satisfy certain specifications for what it means to be an acceptable model in a given context. Importantly, a meta-rule should specify the *actual behaviors desired*: if minimal racial disparity is preferable subject to maximal accuracy, this should be enforced through an explicit outcomes-based constraint, rather than a procedural constraint that stakeholders may expect to reach such an outcome.

A meta-rule should be deliberated over and documented, with justifications for each qualification on the model. Put together, the explanations of the desiderata within a meta-rule constitute the justification behind a decision from a model that satisfies such desiderata. This is because the meta-rule compels model developers to *explicitly* consider the differences that may exist between multiplicitous models and decide on the criteria that are relevant to the application that would disqualify a model (even with high accuracy), instead of choosing arbitrarily.

In practice, the ability to explore the space of equally accurate models in order to find one which satisfies a meta-rule may be constrained by the model developer's ability to experiment with different design choices, which in turn may be influenced by restrictions on the amount of time and money that they can spend on the exploration process. Following Selbst and Barocas [220], the meta-rule should also document the practical constraints that developers face (e.g. available funding, available talent, available data, etc.) in their model selection process, as this ultimately influences the breadth with which they may search for multiplicitous models. Doing so would help to justify the choice of model among a potentially infinite set of alternatives, while also providing the necessary information for others (e.g., the person subject to the decision, an auditor, a regulator, etc.) to assess whether the efforts to find more desirable alternatives were reasonably exhaustive under these constraints.

Further, the very question of how to search among equally accurate models is only beginning to be addressed by the research literature. While some dimensions of exploration may be costly, such as collecting more data to explore alternate features to include in a model, the most common method of exploration—hyperparameter variation [168, 215]—is already standard machine learning practice. Whereas model developers currently explore a range of possible models through hyperparameter tuning and select one that maximizes accuracy, a meta-rule would require that the model developer maintain a set of models that satisfy the meta-rule.

In theory, and often even in practice [214], it is unlikely there is only *one* model which satisfies a meta-rule. As we discuss in the next section, we can account for residual differences in predictions between models which satisfy the meta-rule with model aggregation techniques.

7.6.2 Aggregation Techniques

Given a set of models that all satisfy the meta-rule, how might a decision maker choose among models? In fact, there are several ways to produce a single model from a set of equally "good" models. Here, we focus on three such techniques, and, in particular, we demonstrate that each may be appropriate for use in different contexts. Let $\mathcal{M}$ be a distribution over models that satisfy decision makers' meta-rule.[10] The techniques that follow require that the model developer can construct a *random sample* from $\mathcal{M}$, as opposed to enumerating all of the models in $\mathcal{M}$. The three aggregation techniques we consider are **mode aggregation**, **randomized predictions**, and **random model selection**. Importantly, these techniques help to restore justifiability because they each involve deliberating over how to choose between multiplicitious models.

- **Mode aggregation** [25]: The mode predictor $\overline{m}$ aggregates models from the model distribution $\mathcal{M}$ by outputting the majority vote over the models $m \in \mathcal{M}$ for each example x. Formally, in the case of binary classification, this is

  $$\overline{m}(x) \triangleq \begin{cases} 1 & \Pr_{m \sim \mathcal{M}}[m(x) = 1] \geq \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}.$$

  Note that the mode predictor $\overline{m}$ is the one that minimizes the expected disagreement between itself and a randomly chosen model $m \sim \mathcal{M}$.

- **Randomized Predictions**: Under randomized prediction, the decision maker uses the classifier m^{rand} that, for each example x, randomly samples a model $m \sim \mathcal{M}$ and outputs $m(x)$. Formally, this is

  $$\Pr[m^{\text{rand}}(x) = y] \triangleq \Pr_{m \sim \mathcal{M}}[m(x) = y].$$

- **Random Model Selection**: Under random model selection, the decision maker randomly samples some $m \sim \mathcal{M}$ and applies that m to every decision subject. Note that random model selection differs from an arbitrary selection in that the randomness (and the act of choosing) is made explicit [204].

Each of these techniques, alongside documentation of the reasons why a given method was chosen, provides a justifiable way to resolve multiplicity in different contexts. When decisions are made by a centralized authority, the decision maker's objective may be to resolve multiplicity by providing a consistent predictor (i.e., contains no explicit randomness) that minimizes multiplicity across the model distribution $\mathcal{M}$. The government, for example, has a special legal burden to ensure consistency in decision making [34]. In such cases, the mode predictor best achieves these goals: it is the model that minimizes multiplicity compared to the model distribution $\mathcal{M}$.[11] Recent work has shown that, beyond stabilizing model predictions, mode aggregation also results in more stable model explanations, and thus suggests that models which return the mode over a random sample of similar models have more stable internals than individual models [25].

[10] In practice, $\mathcal{M}$ may be constrained by the decision maker's ability to experiment with different design choices (e.g., type of model, random seed, etc.).

[11] Black et al. [25] provides an evaluation of this approach, including theoretical guarantees on the consistency of mode-aggregated decisions.

On the other hand, consider decisions that are low-stakes and frequent, such as choosing which advertisement to show to a user. Suppose 70% of models in the distribution $\mathcal{M}$ predict that a user x prefers credit card ads, and 30% predict that x prefers ads for cars. While the mode predictor would resolve this multiplicity by always showing x an ad for credit cards, under randomized prediction, the model will show the user credit card ads 70% of the time and car ads the other 30% of the time.[12] Of course, there are plenty of applications where such randomized predictions are undesirable; but in applications where decisions are low-stakes and repeated, this randomized sampling might give a person outcomes that better reflect the uncertainty contained in the model distribution.

Finally, there are cases where society would prefer that the model developer simply samples a random model $m \sim \mathcal{M}$ and always uses m. For example, consider an application like hiring or lending where multiple private actors make independent decisions. We may not want explicit randomness through sampling in these decisions, but if each supposedly independent actor uses the same mode predictor, then decision making effectively becomes a monoculture, which can have negative impacts both for individuals' recourse and social welfare [51, 138]. To prevent this, we might prefer that each model developer independently choose its own random model $m \sim \mathcal{M}$. And while random model selection may seem like the de facto resolution of predictive multiplicity in practice, private model developers may end up converging on the same models for a variety of reasons, including third-party vendors selling the same tools to multiple clients [210] or centralized evaluation (e.g., credit scores).

Crucially, all three of these methods mitigate arbitrariness since choosing among them requires considering and deliberating between the different options. By requiring model developers to document the model building process—and their ultimate decision on how to address remaining multiplicity—we can reach a justification for why a model's internals and predictions are the way that they are [220].

7.7 Conclusion

Our work considers the implications of *model multiplicity*, the phenomenon of multiple models with equal accuracy for a given prediction task exhibiting different individual predictions or aggregate properties. We show that model multiplicity leads to increased flexibility—and perhaps even legal pressure—to prioritize fairness, robustness, and interpretability, among other values, in the model building process. However, this increased flexibility also leads to the risk of avoidable discrimination and to a lack of justification for model decisions when the model is chosen on the basis of accuracy alone. While this work does not serve as a complete exploration of the impact that predictive multiplicity may have on law and policy, we hope that by bringing attention to model multiplicity that we can add to the momentum to take advantage of the opportunities that it creates and head off the resistance that it could provoke.

[12] Randomized prediction is often used in the fairness literature to ensure that individuals or groups have similar probabilities of receiving an given outcome from a classifier [8, 72]. Our use of randomized prediction ensures that an individual has a chance of getting any outcome available to them under some $m \sim \mathcal{M}$.

Chapter 8

Conclusion

In this thesis, we present examples of how considering the AI pipeline in the ideation of notions of fairness, as well as in the creation of techniques to mitigate unfair behavior, can expand our understanding of what algorithmic fairness means, and of how to mitigate unfairness. Towards the former, we have shown how considering the AI pipeline expands fairness conceptualization by encouraging us to examine whether a model's decision *procedure* is unfair. After displaying how learning rule instability can lead to certain types of unfair behavior, we have shown how to mitigate inconsistency in predictions and model explanations.

Towards the latter point, we have demonstrated how the AI pipeline expands our toolbox of bias mitigation techniques, by showing how changing choices along the AI pipeline not necessarily related to fair behavior at first glance—such as changing a model's prediction target from classification to regression—can greatly improve context-specific fairness behavior. This work also showcases the utility of pipeline-based fairness interventions in deeply contextualized, real-world machine learning bias mitigation: given that pipeline-based interventions are not inherently tied to any one notion of fairness, pipeline interventions can be constructed for a wide variety of desired behaviors.

Finally, we have pointed to legal and policy implications of this flexibility and instability, which we have combined under the umbrella of *model multiplicity.* In particular, we argue that model multiplicity puts legal pressure on companies to search among the set of equally viable models for a given task, to find the most equitable model, and we suggest a documentation framework for providing justification for decisions made along the model creation pipeline, to remedy any concerns of arbitrary decision-making.

As a whole, this work aims to show the immense flexibility that the machine learning creation pipeline gives practitioners with respect to how to reach the goals they set out to acheive with machine learning models— but also, how the choices made along the AI pipeline have to be carefully considered in terms of their impact on desired model behaviors, as even seemingly small choices can have a large impact. By considering all the choices made along the machine learning pipeline—from feature selection, to model type, to the objective functions—as places where changes can be made to improve fairness behavior, we can greatly expand our arsenal of unfairness-fighting tools, and our understanding of fairness behavior.

This work shows that pipeline-based perspective of on fairness in AI systems can add to the arsenal that we use to understand and improve fairness behavior. However, at present it is unclear how to transfer knowledge from this approach across applications or even across implementations. In order for such methods to be used with similar efficacy as mainstream fairness interventions—i.e. in order to acheive some degree of generalizability—much research must be done to map the choices made along the pipeline to fairness behaviors.

For example, on the purely technical side, how do decisions at each stage of the pipeline—data selection; feature creation; definition of the prediction target; model, learning rule, and hyperparameter selection—influence machine learning behavior? What additional unfairnesses can these decisions produce? Are there any generalizable patterns across deployment contexts? Can any of these intervention points reliably be used as fairness solutions? Even more broadly, how do parts of the pipeline *beyond* purely technical functioning, such as meeting points between machine and human control and agency constraints, influence fairness performance? We leave these lines of inquiry as exciting areas for future work.

Bibliography

[1] *116 Cong. Reg. 36572*, 1970.

[2] *Comment for 1002.6 - Rules Concerning Evaluation of Applications.* https://www.consumerfinance.gov/rules-policy/regulations/1002/interp-6/, 2011.

[3] *Equal Credit Opportunity Act (Regulation B).* https://www.fdic.gov/regulations/laws/rules/6500-200.html, 2011.

[4] Peter Addo, Dominique Guegan, and Bertrand Hassani. *Credit Risk Analysis Using Machine and Deep Learning Models.* Risks, Apr 2018.

[5] Julius Adebayo, Michael Muelly, Ilaria Liccardi, and Been Kim. *Debugging Tests for Model Explanations.* In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, Advances in Neural Information Processing Systems, volume 33, pages 700–712. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/075b051ec3d22dac7b33f788da631fd4-Paper.pdf.

[6] National Taxpayer Advocate. *Annual Report to Congress*, 2019. URL https://www.taxpayeradvocate.irs.gov/reports/2019-annual-report-to-congress/full-report/.

[7] National Taxpayer Advocate. *The IRS is Significantly Underfunded to Serve Taxpayers and Collect Tax*, 2020. URL https://www.taxpayeradvocate.irs.gov/wp-content/uploads/2020/08/Most-Serious-Problems-IRS-Significantly-Underfunded.pdf.

[8] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudik, John Langford, and Hanna Wallach. *A Reductions Approach to Fair Classification.* In International Conference on Machine Learning, pages 60–69. PMLR, 2018.

[9] Christopher Anders, Plamen Pasliev, Ann-Kathrin Dombrowski, Klaus-Robert Müller, and Pan Kessel. *Fairwashing explanations with off-manifold detergent.* In International Conference on Machine Learning, pages 314–323. PMLR, 2020.

[10] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. *Machine bias: There's software used across the country to predict future criminals. And it's biased against blacks.* ProPublica, 2016.

[11] Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. *A closer look at memorization in deep networks.* In Proceedings of the 34th International Conference on Machine Learning-Volume 70, 2017.

[12] Dmitrii Babaev et al. *ET-RNN: Applying Deep Learning to Credit Loan Applications.*

In KDD, 2019.

[13] Mihalj Bakator and Dragica Radosav. *Deep Learning and Medical Diagnosis: A Review of Literature.* Multimodal Technologies and Interaction, 2(3):47, Aug 2018.

[14] Ramnath Balasubramanian et al. *Insurance 2030: The impact of AI on the future of insurance.* McKinsey & Company, 2018.

[15] Solon Barocas and Andrew D Selbst. *Big data's disparate impact.* California Law Review, 104:671–732, 2016.

[16] Solon Barocas, Andrew D Selbst, and Manish Raghavan. *The hidden assumptions behind counterfactual explanations and principal reasons.* In Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, pages 80–89, 2020.

[17] Rachel K. E. Bellamy, Kuntal Dey, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, Seema Nagar, Karthikeyan Natesan Ramamurthy, John T. Richards, Diptikalyan Saha, Prasanna Sattigeri, Moninder Singh, Kush R. Varshney, and Yunfeng Zhang. *AI Fairness 360: An Extensible Toolkit for Detecting, Understanding, and Mitigating Unwanted Algorithmic Bias.* CoRR, abs/1810.01943, 2018. URL `http://arxiv.bs/1810.0194`.

[18] Marcus C Berliant, Robert P Strauss, et al. *The horizontal and vertical equity characteristics of the federal individual income tax, 1966-1977.* Horizontal Equity, Uncertainty and Economic Well-being, pages 179–211, 1985.

[19] Marianne Bertrand and Sendhil Mullainathan. *Are Emily and Greg more employable than Lakisha and Jamal? A field experiment on labor market discrimination.* American economic review, 94(4):991–1013, 2004.

[20] Dimitris Bertsimas, Arthur Delarue, Patrick Jaillet, and Sebastien Martin. *The price of interpretability.* arXiv preprint arXiv:1907.03419, 2019.

[21] Reuben Binns. *Fairness in machine learning: Lessons from political philosophy.* In Conference on Fairness, Accountability and Transparency, pages 149–159. PMLR, 2018.

[22] Sarah Bird, Miro Dudík, Richard Edgar, Brandon Horn, Roman Lutz, Vanessa Milan, Mehrnoosh Sameki, Hanna Wallach, and Kathleen Walker. *Fairlearn: A toolkit for assessing and improving fairness in AI.* Technical Report MSR-TR-2020-32, Microsoft, May 2020. URL `https://www.microsoft.com/en-us/research/publication/fairlearn-a-toolkit-for-assessing-and-improving-fairness-in-ai/`.

[23] Abeba Birhane, Pratyusha Kalluri, Dallas Card, William Agnew, Ravit Dotan, and Michelle Bao. *The values encoded in machine learning research.* arXiv preprint arXiv:2106.15590, 2021.

[24] Emily Black and Matt Fredrikson. *Leave-One-out Unfairness.* In Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, FAccT '21, pages 285–295, 2021.

[25] Emily Black, Klas Leino, and Matt Fredrikson. *Selective Ensembles for Consistent Predictions.* arXiv preprint, 2021.

[26] Emily Black, Zifan Wang, Matt Fredrikson, and Anupam Datta. *Consistent Counterfactuals for Deep Models.* arXiv preprint arXiv:2110.03109, 2021.

[27] Emily Black, Solon Barocas, Alexandra Chouldechova, Logan Koepke, Kristian Lum,

Michael Madaio, and Sarah Riley. *Reducing Racial Disparity Through Procedural Interventions: Conceptions and Outcomes.* 2022.

[28] Emily Black, Hadi Elzayn, Alexandra Chouldechova, Jacob Goldin, and Daniel Ho. *Algorithmic Fairness and Vertical Equity: Income Fairness with IRS Tax Audit Models.* In ACM FAccT 2022, 2022.

[29] Emily Black, Manish Raghavan, and Solon Barocas. *Model Multiplicity: Opportunities, Concerns, and Solutions.* In ACM FAccT 2022, 2022.

[30] Bloomberg. *Equifax AI Innovation Opens Doors to Millions Seeking Credit.* 2020.

[31] J Frédéric Bonnans and Alexander Shapiro. *Perturbation analysis of optimization problems.* Springer Science & Business Media, 2013.

[32] Olivier Bousquet and André Elisseeff. *Stability and generalization.* Journal of machine learning research, 2(Mar):499–526, 2002.

[33] Leo Breiman. *Statistical modeling: The two cultures (with comments and a rejoinder by the author).* Statistical science, 16(3):199–231, 2001.

[34] Lisa Schultz Bressman. *Beyond accountability: Arbitrariness and legitimacy in the administrative state.* NYUL Rev., 78:461, 2003.

[35] Anna Brown, Alexandra Chouldechova, Emily Putnam-Hornstein, Andrew Tobin, and Rhema Vaithianathan. *Toward algorithmic accountability in public services: A qualitative study of affected community perspectives on algorithmic decision-making in child welfare services.* In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, pages 1–12, 2019.

[36] Kieran Browne and Ben Swift. *Semantics and explanation: why counterfactual explanations produce adversarial examples in deep neural networks.* arXiv preprint arXiv:2012.10076, 2020.

[37] L Elisa Celis, Damian Straszak, and Nisheeth K Vishnoi. *Ranking with fairness constraints.* arXiv preprint arXiv:1704.06840, 2017.

[38] L Elisa Celis, Lingxiao Huang, Vijay Keswani, and Nisheeth K Vishnoi. *Classification with fairness constraints: A meta-algorithm with provable guarantees.* In Proceedings of the conference on fairness, accountability, and transparency, pages 319–328, 2019.

[39] Tax Policy Center. *Sources of revenue for the federal government*, 2019.

[40] Dongsheng Che, Qi Liu, Khaled Rasheed, and Xiuping Tao. *Decision tree and ensemble learning algorithms with their applications in bioinformatics.* Software tools and algorithms for biological systems, pages 191–199, 2011.

[41] Chaofan Chen, Kangcheng Lin, Cynthia Rudin, Yaron Shaposhnik, Sijia Wang, and Tong Wang. *An interpretable model with globally consistent explanations for credit risk.* arXiv preprint arXiv:1811.12615, 2018.

[42] Irene Chen, Fredrik D Johansson, and David Sontag. *Why is my classifier discriminatory?* arXiv preprint arXiv:1805.12002, 2018.

[43] Pin-Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. *Ead: elastic-net attacks to deep neural networks via adversarial examples.* In Proceedings of the AAAI Conference on Artificial Intelligence, volume 32, 2018.

[44] Danielle Keats Citron. *Technological due process.* Wash. L Rev., 85:1249, 2007.

[45] Danielle Keats Citron and Frank Pasquale. *The scored society: Due process for automated predictions.* Wash. L. Rev., 89:1, 2014.

[46] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. *Certified Adversarial Robustness via Randomized Smoothing.* In Proceedings of the 36th International Conference on Machine Learning (ICML), 2019.

[47] International Warfarin Pharmacogenetics Consortium. *Estimation of the warfarin dose with clinical and pharmacogenetic data.* New England Journal of Medicine, 360(8): 753–764, 2009.

[48] A Feder Cooper and Ellen Abrams. *Emergent Unfairness in Algorithmic Fairness-Accuracy Trade-Off Research.* In Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society, pages 46–54, 2021.

[49] Amanda Coston, Ashesh Rambachan, and Alexandra Chouldechova. *Characterizing fairness over the set of good models under selective labels.* arXiv preprint arXiv:2101.00352, 2021.

[50] Kate Crawford and Jason Schultz. *Big data and due process: Toward a framework to redress predictive privacy harms.* BCL Rev., 55:93, 2014.

[51] Kathleen Creel and Deborah Hellman. *The Algorithmic Leviathan: Arbitrariness, Fairness, and Opportunity in Algorithmic Decision Making Systems.* Virginia Public Law and Legal Theory Research Paper, (2021-13), 2021.

[52] Francesco Croce, Maksym Andriushchenko, and Matthias Hein. *Provable Robustness of ReLU networks via Maximization of Linear Regions.* AISTATS 2019, 2019.

[53] Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D Hoffman, et al. *Underspecification presents challenges for credibility in modern machine learning.* arXiv preprint arXiv:2011.03395, 2020.

[54] Susanne Dandl, Christoph Molnar, Martin Binder, and Bernd Bischl. *Multi-objective counterfactual explanations.* In International Conference on Parallel Problem Solving from Nature, pages 448–469. Springer, 2020.

[55] Jeffrey Dastin. *Amazon scraps secret AI recruiting tool that showed bias against women.* Reuters, 2018.

[56] Anupam Datta, Shayak Sen, and Yair Zick. *Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems.* In IEEE Symposium on Security and Privacy, pages 598–617, 2016.

[57] Anupam Datta, Matt Fredrikson, Gihyuk Ko, Piotr Mardziel, and Shayak Sen. *Proxy Discrimination in Data-Driven Systems.* arXiv preprint arXiv:1707.08120, 2017.

[58] Anupam Datta, Matt Fredrikson, Gihyuk Ko, Piotr Mardziel, and Shayak Sen. *Use Privacy in Data-Driven Systems: Theory and Experiments with Machine Learnt Programs.* In ACM SIGSAC Conference on Computer and Communications Security, 2017.

[59] Jeffrey De Fauw, Joseph R Ledsam, Bernardino Romera-Paredes, Stanislav Nikolov, Nenad Tomasev, Sam Blackwell, Harry Askham, Xavier Glorot, Brendan O'Donoghue, Daniel Visentin, et al. *Clinically applicable deep learning for diagnosis and referral in retinal disease.* Nature medicine, 24(9):1342–1350, 2018.

[60] Amit Dhurandhar, Pin-Yu Chen, Ronny Luss, Chun-Chen Tu, Paishun Ting, Karthikeyan Shanmugam, and Payel Das. *Explanations based on the missing: Towards contrastive explanations with pertinent negatives.* arXiv preprint arXiv:1802.07623, 2018.

[61] Ann-Kathrin Dombrowski, Maximilian Alber, Christopher J Anders, Marcel Ackermann, Klaus-Robert Müller, and Pan Kessel. *Explanations can be manipulated and geometry is to blame.* arXiv preprint arXiv:1906.07983, 2019.

[62] Pedro Domingos. *A unified bias-variance decomposition.* In Proceedings of 17th International Conference on Machine Learning, pages 231–238, 2000.

[63] Jiayun Dong and Cynthia Rudin. *Variable importance clouds: A way to explore variable importance for the set of good models.* arXiv preprint arXiv:1901.03209, 2019.

[64] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. *Boosting adversarial attacks with momentum.* In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 9185–9193, 2018.

[65] Michele Donini, Luca Oneto, Shai Ben-David, John Shawe-Taylor, and Massimiliano Pontil. *Empirical risk minimization under fairness constraints.* arXiv preprint arXiv:1802.08626, 2018.

[66] David Donoho. *50 years of data science.* Journal of Computational and Graphical Statistics, 26(4):745–766, 2017.

[67] Dheeru Dua and Efi Karra Taniskidou. *UCI Machine Learning Repository.* https:/ive.ics.uci.edu/ml, 2017. URL `http://archive.ics.uc/m`.

[68] Sanghamitra Dutta, Dennis Wei, Hazar Yueksel, Pin-Yu Chen, Sijia Liu, and Kush Varshney. *Is there a trade-off between fairness and accuracy? a perspective using mismatched hypothesis testing.* In International Conference on Machine Learning, pages 2803–2813. PMLR, 2020.

[69] Nikita Dvornik, Cordelia Schmid, and Julien Mairal. *Diversity with cooperation: Ensemble methods for few-shot classification.* In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 3723–3731, 2019.

[70] Cynthia Dwork. *Differential Privacy*, 2006.

[71] Cynthia Dwork and Christina Ilvento. *Fairness Under Composition.* CoRR, abs/1806.06122, 2018.

[72] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. *Fairness through awareness.* In Innovations in Theoretical Computer Science, pages 214–226, 2012.

[73] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Leon Roth. *Preserving Statistical Validity in Adaptive Data Analysis.* In Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing (STOC), 2015.

[74] Herbert Edelhertz. *The nature, impact, and prosecution of white-collar crime*, volume 2. National Institute of Law Enforcement and Criminal Justice, 1970.

[75] Ran El-Yaniv et al. *On the Foundations of Noise-free Selective Classification.* Journal of Machine Learning Research, 11(5), 2010.

[76] André Elisseeff, Massimiliano Pontil, et al. *Leave-one-out error and stability of learning*

algorithms with applications. NATO science series sub series iii computer and systems sciences, 2003.

[77] David Elkins. *Horizontal equity as a principle of tax theory.* Yale L. & Pol'y Rev., 24: 43, 2006.

[78] Hadi Elzayn, Evelyn Smith, Jacob Goldin, and Daniel Ho. *Horizontal Equity in IRS Audits.* In Forthcoming, 2022.

[79] David Freeman Engstrom, Daniel E Ho, Catherine M Sharkey, and Mariano-Florentino Cuéllar. *Government by algorithm: Artificial intelligence in federal administrative agencies.* NYU School of Law, Public Law Research Paper, (20-54), 2020.

[80] Equal Credit Opportunities Act, Public Law 93-495. *Codified at 15 U.S.C. § 1691, et seq.*, 1974.

[81] Geert Litjens et. al. *A survey on deep learning in medical image analysis.* Medical Image Analysis, 2017.

[82] Christian Etmann, Sebastian Lunz, Peter Maass, and Carola-Bibiane Schönlieb. *On the Connection Between Adversarial Robustness and Saliency Map Interpretability.* In ICML, 2019.

[83] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. *Certifying and removing disparate impact.* In ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015.

[84] Vitaly Feldman. *Does Learning Require Memorization? A Short Tale about a Long Tail.* CoRR, abs/1906.05271, 2019.

[85] FICO. *FICO xML Challenge.* https://community.fico.com/s/explainable-machine-learning-challenge, 2018.

[86] FICO. *Dataset Usage License, FICO xML Challenge.* https://community.fico.com/s/explainable-machine-learning-challenge?tabset-3158a=a4c37, 2018.

[87] Aaron Fisher, Cynthia Rudin, and Francesca Dominici. *All Models are Wrong, but Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously.* J. Mach. Learn. Res., 20(177):1–81, 2019.

[88] Timo Freiesleben. *Counterfactual Explanations & Adversarial Examples–Common Grounds, Essential Differences, and Potential Transfers.* arXiv preprint arXiv:2009.05487, 2020.

[89] Yoav Freund and Robert E Schapire. *A decision-theoretic generalization of on-line learning and an application to boosting.* Journal of computer and system sciences, 55 (1):119–139, 1997.

[90] Aymeric Fromherz, Klas Leino, Matt Fredrikson, Bryan Parno, and Corina Păsăreanu. *Fast Geometric Projections for Local Robustness Certification.* In International Conference on Learning Representations (ICLR), 2021.

[91] Giorgio Fumera, Fabio Roli, and Alessandra Serrau. *Dynamics of variance reduction in bagging and other techniques based on randomisation.* In International Workshop on Multiple Classifier Systems, 2005.

[92] Clare Garvie, Alvaro Bedoya, and Jonathan Frankle. *The Perpetual Lineup*, 2016.

[93] GDPR. *European Parliament and Council of European Union (2016) Regulation (EU) 2016/679.* `https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32016R0679&from=EN`, 2016.

[94] Stuart Geman, Elie Bienenstock, and René Doursat. *Neural networks and the bias/variance dilemma.* Neural computation, 4(1):1–58, 1992.

[95] Sahin Cem Geyik, Stuart Ambler, and Krishnaram Kenthapadi. *Fairness-aware ranking in search & recommendation systems with application to linkedin talent search.* In Proceedings of the 25th acm sigkdd international conference on knowledge discovery & data mining, pages 2221–2231, 2019.

[96] Amirata Ghorbani, Abubakar Abid, and James Zou. *Interpretation of neural networks is fragile.* In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, pages 3681–3688, 2019.

[97] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. *Explaining and Harnessing Adversarial Examples.* arXiv 1412.6572, 12 2014.

[98] Stefan Gosepath. *Equality.* In Edward N. Zalta, editor, The Stanford Encyclopedia of Philosophy. Metaphysics Research Lab, Stanford University, Summer 2021 edition, 2021.

[99] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Dino Pedreschi, Franco Turini, and Fosca Giannotti. *Local rule-based explanations of black box decision systems.* arXiv preprint arXiv:1805.10820, 2018.

[100] Chirag Gupta, Arun K Kuchibhotla, and Aaditya K Ramdas. *Nested conformal prediction and quantile out-of-bag ensemble methods.* arXiv preprint arXiv:1910.10562, 2019.

[101] John Guyton, Kara Leibel, Dayanand S Manoli, Ankur Patel, Mark Payne, and Brenda Schafer. *The effects of EITC correspondence audits on low-income earners.* Technical report, National Bureau of Economic Research, 2018.

[102] Jonathan Haidt. *The righteous mind: Why good people are divided by politics and religion.* Vintage, 2012.

[103] Boris Hanin and David Rolnick. *Deep ReLU Networks Have Surprisingly Few Activation Patterns.* In Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, 2019.

[104] Lars Kai Hansen and Peter Salamon. *Neural network ensembles.* IEEE transactions on pattern analysis and machine intelligence, 12(10):993–1001, 1990.

[105] Moritz Hardt, Eric Price, and Nati Srebro. *Equality of opportunity in supervised learning.* Advances in neural information processing systems, 29, 2016.

[106] Moritz Hardt, Eric Price, and Nati Srebro. *Equality of opportunity in supervised learning.* In Advances in Neural Information Processing Systems, volume 29 of NIPS'16, pages 3315–3323, 2016.

[107] Moritz Hardt, Benjamin Recht, and Yoram Singer. *Train Faster, Generalize Better: Stability of Stochastic Gradient Descent.* In Proceedings of the 33rd International Conference on International Conference on Machine Learning, ICML'16, 2016.

[108] Md Kamrul Hasan, Md Ashraful Alam, Dola Das, Eklas Hossain, and Mahmudul

Hasan. *Diabetes prediction using ensembling of different machine learning classifiers.* IEEE Access, 8:76516–76531, 2020.

[109] K. He, X. Zhang, S. Ren, and J. Sun. *Deep Residual Learning for Image Recognition.* In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[110] Hoda Heidari, Michele Loi, Krishna P Gummadi, and Andreas Krause. *A moral framework for understanding fair ml through economic models of equality of opportunity.* In Proceedings of the conference on fairness, accountability, and transparency, pages 181–190, 2019.

[111] Juyeon Heo, Sunghwan Joo, and Taesup Moon. *Fooling Neural Network Interpretations via Adversarial Model Manipulation.* Advances in Neural Information Processing Systems, 32:2925–2936, 2019.

[112] Kashmir Hill. *Wrongfully accused by an algorithm.* The New York Times, June, 24, 2020.

[113] Daniel Ho. *AI for Government.* URL `https://www.youtube.com/watch?v=SnGUWHgLP-Q&ab_channel=ICMEStudio`.

[114] Kenneth Holstein, Jennifer Wortman Vaughan, Hal Daumé III, Miro Dudik, and Hanna Wallach. *Improving fairness in machine learning systems: What do industry practitioners need?* In Proceedings of the 2019 CHI conference on human factors in computing systems, pages 1–16, 2019.

[115] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. *Labeled faces in the wild: A database forstudying face recognition in unconstrained environments.* 2008.

[116] Kenneth Hung, William Fithian, et al. *Rank verification for exponential families.* Annals of Statistics, 47(2):758–782, 2019.

[117] William J Hunter and Michael A Nelson. *An IRS production function.* National Tax Journal, 49(1):105–115, 1996.

[118] Ben Hutchinson and Margaret Mitchell. *50 years of test (un) fairness: Lessons for machine learning.* In Proceedings of the Conference on Fairness, Accountability, and Transparency, pages 49–58, 2019.

[119] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. *Adversarial Examples Are Not Bugs, They Are Features.* In Advances in Neural Information Processing Systems 32. 2019.

[120] International Warfarin Pharmacogenetic Consortium. *Estimation of the Warfarin Dose with Clinical and Pharmacogenetic Data.* New England Journal of Medicine, 360(8): 753–764, 2009.

[121] Internal Revenue Service (IRS). *National Research Program Overview*, . URL `https://www.irs.gov/irm/part4/irm_04-022-001`.

[122] Internal Revenue Service (IRS). *IRS Update on Audits*, . URL `https://www.irs.gov/newsroom/irs-update-on-audits`.

[123] Internal Revenue Service (IRS). *Compliance Presence*, . URL `https://www.irs.gov/statistics/compliance-presence`.

[124] Internal Revenue Service (IRS). *NRP Examining Process*, . URL `https://www.irs.gov/irm/part4/irm_04-022-004r`.

[125] Internal Revenue Service (IRS). *IRS Newsroom*, . URL `https://www.irs.gov/newsroom/the-tax-gap`.

[126] Matt Jordan, Justin Lewis, and A. Dimakis. *Provable Certificates for Adversarial Examples: Fitting a Ball in the Union of Polytopes.* In NeurIPS, 2019.

[127] Matthew Joseph, Michael Kearns, Jamie H Morgenstern, and Aaron Roth. *Fairness in learning: Classic and contextual bandits.* In Advances in Neural Information Processing Systems, 2016.

[128] Shalmali Joshi, Oluwasanmi Koyejo, Warut Vijitbenjaronk, Been Kim, and Joydeep Ghosh. *Towards realistic individual recourse and actionable explanations in black-box decision making systems.* arXiv preprint arXiv:1907.09615, 2019.

[129] Margot E Kaminski. *The right to explanation, explained.* Berkeley Tech. LJ, 34:189, 2019.

[130] Amir-Hossein Karimi, Gilles Barthe, Bernhard Schölkopf, and Isabel Valera. *A survey of algorithmic recourse: definitions, formulations, solutions, and prospects.* arXiv preprint arXiv:2010.04050, 2020.

[131] Amir-Hossein Karimi, Julius von Kügelgen, Bernhard Schölkopf, and Isabel Valera. *Algorithmic recourse under imperfect causal knowledge: a probabilistic approach.* arXiv preprint arXiv:2006.06831, 2020.

[132] Jakob Nikolas Kather, Cleo-Aron Weis, Francesco Bianconi, Susanne M Melchers, Lothar R Schad, Timo Gaiser, Alexander Marx, and Frank Gerrit Z"ollner. *Multi-class texture analysis in colorectal cancer histology.* Scientific reports, 6:27988, 2016.

[133] Mark T Keane and Barry Smyth. *Good counterfactuals and where to find them: A case-based technique for generating counterfactuals for explainable ai (xai).* In International Conference on Case-Based Reasoning, pages 163–178. Springer, 2020.

[134] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. *Preventing Fairness Gerrymandering: Auditing and Learning for Subgroup Fairness.* In International Conference on Machine Learning, pages 2564–2572. PMLR, 2018.

[135] Paul Kiel. *It's Getting Worse: The IRS Now Audits Poor Americans at About the Same Rate as the Top 1%.*

[136] Paul Kiel and Jesse Eisinger. *Who's more likely to be audited: A person making $20,000 or $400, 000*, 2018.

[137] Pauline T. Kim. *Race-aware algorithms: Fairness, nondiscrimination and affirmative action.* California Law Review, 110, 2022.

[138] Jon Kleinberg and Manish Raghavan. *Algorithmic monoculture and social welfare.* Proceedings of the National Academy of Sciences, 118(22), 2021.

[139] Ron Kohavi, David H Wolpert, et al. *Bias plus variance decomposition for zero-one loss functions.* In ICML, volume 96, pages 275–83, 1996.

[140] John Kolen and Jordan Pollack. *Back Propagation is Sensitive to Initial Conditions.* In Neural Information Processing Systems (NeurIPS), 1991.

[141] Anders Krogh and Jesper Vedelsby. *Neural Network Ensembles, Cross Validation, and Active Learning.* Neural Information Processing Systems (NeurIPS), 1995.

[142] Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. *Counterfactual*

Fairness. In Advances in Neural Information Processing Systems, 2017.

[143] Ilja Kuzborskij and Christoph H. Lampert. *Data-Dependent Stability of Stochastic Gradient Descent.* In ICML, 2018.

[144] Himabindu Lakkaraju, Jon Kleinberg, Jure Leskovec, Jens Ludwig, and Sendhil Mullainathan. *The selective labels problem: Evaluating algorithmic predictions in the presence of unobservables.* In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 275–284, 2017.

[145] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. *Simple and scalable predictive uncertainty estimation using deep ensembles.* arXiv preprint arXiv:1612.01474, 2016.

[146] Hemank Lamba, Kit T Rodolfa, and Rayid Ghani. *An Empirical Comparison of Bias Reduction Methods on Real-World Problems in High-Stakes Policy Settings.* ACM SIGKDD Explorations Newsletter, 23(1):69–85, 2021.

[147] Loren Larsen. *Resumes, Robots, and Racism: The Truth about AI in Hiring.* HireVue, Mar 2019.

[148] Michael T Lash, Qihang Lin, Nick Street, Jennifer G Robinson, and Jeffrey Ohlmann. *Generalized inverse classification.* In Proceedings of the 2017 SIAM International Conference on Data Mining, pages 162–170. SIAM, 2017.

[149] Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. *Comparison-based inverse classification for interpretability in machine learning.* IPMU, 2018.

[150] Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, and Marcin Detyniecki. *Issues with post-hoc counterfactual explanations: a discussion.* arXiv preprint arXiv:1906.04774, 2019.

[151] Colin Lecher. *What Happens When an Algorithm Cuts Your Health Care.* The Verge, 2018.

[152] Yann LeCun, LD Jackel, Léon Bottou, Corinna Cortes, John S Denker, Harris Drucker, Isabelle Guyon, Urs A Muller, Eduard Sackinger, Patrice Simard, et al. *Learning algorithms for classification: A comparison on handwritten digit recognition.* Neural networks: the statistical mechanics perspective, 261:276, 1995.

[153] David Lehr and Paul Ohm. *Playing with the data: what legal scholars should learn about machine learning.* UCDL Rev., 51:653, 2017.

[154] Klas Leino and Matt Fredrikson. *Stolen Memories: Leveraging Model Memorization for Calibrated White-Box Membership Inference.* 2020.

[155] Klas Leino, Shayak Sen, Anupam Datta, Matt Fredrikson, and Linyi Li. *Influence-Directed Explanations for Deep Convolutional Networks.* In IEEE International Test Conference (ITC), 2018.

[156] Klas Leino, Matt Fredrikson, Emily Black, Shayak Sen, and Anupam Datta. *Feature-Wise Bias Amplification.* In International Conference on Learning Representations, 2019. URL `https://openreview.net/?id=S1ecm2C9K`.

[157] William Lincoln and Josef Skrzypek. *Synergy of Clustering Multiple Back Propagation Networks.* In Neural Information Processing Systems (NeurIPS), 1990.

[158] Henrik Linusson, Ulf Johansson, and Henrik Boström. *Efficient conformal predictor*

ensembles. Neurocomputing, 397:266–278, 2020.

[159] Zachary C Lipton. *The mythos of model interpretability.* Queue, 16(3):31–57, 2018.

[160] Hongbin Liu, Jinyuan Jia, and Neil Zhenqiang Gong. *On the Intrinsic Differential Privacy of Bagging.* arXiv preprint arXiv:2008.09845, 2020.

[161] Siqi Liu, Sidong Liu, Weidong Cai, Sonia Pujol, Ron Kikinis, and Dagan Feng. *Early diagnosis of Alzheimer's disease with deep learning.* In 2014 IEEE 11th international symposium on biomedical imaging (ISBI), pages 1015–1018. IEEE, 2014.

[162] Tuve Löfström, Ulf Johansson, and Henrik Boström. *Effective utilization of data in inductive conformal prediction using ensembles of neural networks.* In The 2013 International Joint Conference on Neural Networks (IJCNN), pages 1–8. IEEE, 2013.

[163] Elizabeth Lopatto. *Clearview AI CEO says 'over 2,400 police agencies' are using its facial recognition software*, 2020.

[164] Richard Maclin, Jude W Shavlik, et al. *Combining the predictions of multiple classifiers: Using competitive learning to initialize neural networks.* In International Joint Conference on Artificial Intelligence (IJCAI), 1995.

[165] Jill MacNabb. *Study of Tax Court Cases In Which the IRS Conceded the Taxpayer was Entitled to Earned Income Tax Credit (EITC).* URL `https://www.taxpayeradvocate.irs.gov/wp-content/uploads/2020/08/Research-Studies-Study-of-Tax-Court-Cases-in-Which-the-IRS-Conceded-/the-Taxpayer-was-Entitled-to-Earned-Income-Tax-Credit-EITC.pdf`.

[166] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. *Towards Deep Learning Models Resistant to Adversarial Attacks.* In International Conference on Learning Representations, 2018.

[167] Divyat Mahajan, Chenhao Tan, and Amit Sharma. *Preserving causal constraints in counterfactual explanations for machine learning classifiers.* arXiv preprint arXiv:1912.03277, 2019.

[168] Charles T. Marx, Flávio du Pin Calmon, and Berk Ustun. *Predictive Multiplicity in Classification.* CoRR, abs/1909.06677, 2019. URL `http://arxiv.org/abs/1909.06677`.

[169] Rory Mc Grath, Luca Costabello, Chan Le Van, Paul Sweeney, Farbod Kamiab, Zhao Shen, and Freddy Lecue. *Interpretable Credit Application Predictions With Counterfactual Explanations.* In NIPS 2018-Workshop on Challenges and Opportunities for AI in Financial Services: the Impact of Fairness, Explainability, Accuracy, and Privacy, 2018.

[170] Paul R McDaniel and James R Repetti. *Horizontal and vertical equity: the Musgrave/Kaplow exchange.* Fla. Tax Rev., 1:607, 1992.

[171] Jamse R. Jr. McTigue. *In a Limited Study, Preparers Made Significant Errors.* URL `https://www.gao.gov/assets/gao-14-467t.pdf`.

[172] Johannes Mehrer, Courtney J Spoerer, Nikolaus Kriegeskorte, and Tim C Kietzmann. *Individual differences among deep neural network models.* Nature communications, 11 (1):1–12, 2020.

[173] Ronny Meir, G Tesauro, DS Touretzky, and TK Leen. *Bias, variance and the combination of least squares estimators.* Advances in neural information processing systems,

pages 295–302, 1995.

[174] Aditya Krishna Menon and Robert C Williamson. *The cost of fairness in binary classification.* In Conference on Fairness, Accountability and Transparency, pages 107–118. PMLR, 2018.

[175] Gordon et al. Merchant. *Model lifecycle transformation: How banks are unlocking efficiencies: Accenture*, Dec 2020.

[176] Ilya Mironov. *Rényi Differential Privacy.* In Proceedings of 30th IEEE Computer Security Foundations Symposium (CSF), 2017.

[177] Shira Mitchell, Eric Potash, Solon Barocas, Alexander D'Amour, and Kristian Lum. *Algorithmic fairness: Choices, assumptions, and definitions.* Annual Review of Statistics and Its Application, 8:141–163, 2021.

[178] Gavin Mooney and Stephen Jan. *Vertical equity: weighting outcomes? or establishing procedures?* Health Policy, 39(1):79–87, 1997.

[179] Ramaravind K Mothilal, Amit Sharma, and Chenhao Tan. *Explaining machine learning classifiers through diverse counterfactual explanations.* In Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, pages 607–617, 2020.

[180] Sendhil Mullainathan and Ziad Obermeyer. *On the Inequity of Predicting A While Hoping for B.* In AEA Papers and Proceedings, volume 111, pages 37–42, 2021.

[181] Madhumita Murgia. *Who's using your face? The ugly truth about facial recognition/.* Financial Times, 2019.

[182] Richard A. Musgrave and Joel Slemrod. *Progressive taxation, equity, and tax design*, page 341–356. Cambridge University Press, 1994. doi: 10.1017/CBO9780511571824.019.

[183] Ury Naftaly, Nathan Intrator, and David Horn. *Optimal ensemble averaging of neural networks.* Network: Computation in Neural Systems, 8(3):283–296, 1997.

[184] Maria-Irina Nicolae, Mathieu Sinn, Minh Ngoc Tran, Beat Buesser, Ambrish Rawat, Martin Wistuba, Valentina Zantedeschi, Nathalie Baracaldo, Bryant Chen, Heiko Ludwig, Ian Molloy, and Ben Edwards. *Adversarial Robustness Toolbox v1.2.0.* CoRR, 1807.01069, 2018. URL `https://arxiv.org/pdf/1807.01069`.

[185] Adam Noack, Isaac Ahern, Dejing Dou, and Boyang Li. *Does Interpretability of Neural Networks Imply Adversarial Robustness?* CoRR, abs/1912.03430, 2019.

[186] Ziad Obermeyer, Brian Powers, Christine Vogeli, and Sendhil Mullainathan. *Dissecting racial bias in an algorithm used to manage the health of populations.* Science, 366 (6464):447–453, 2019.

[187] Ziad Obermeyer, Brian Powers, Christine Vogeli, and Sendhil Mullainathan. *Dissecting racial bias in an algorithm used to manage the health of populations.* Science, 366 (6464):447–453, 2019.

[188] Bureau of Consumer Financial Protection. *Fair Credit Reporting Act.* `https://www.ftc.gov/enforcement/statutes/fair-credit-reporting-act`, 2020.

[189] U.S. Department of the Treasury. *How Financial Reporting Helps American Workers and Ensures that Top Earners Pay Their Fair Share*, Oct 2021. URL `https://home.treasury.gov/news/featured-stories/how-financial-reporting-helps-american-workers-and-ensures-that/-top-earners-pay-their-fair-share`.

[190] U.S. Government Accountability Office. *Artificial Intelligence: An Accountability Framework for Federal Agencies and Other Entities.* June 2021.

[191] David Opitz and Richard Maclin. *Popular ensemble methods: An empirical study.* Journal of artificial intelligence research, 11:169–198, 1999.

[192] David Opitz and Jude Shavlik. *Generating Accurate and Diverse Members of a Neural-Network Ensemble.* In Advances in Neural Information Processing Systems, 1996.

[193] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua V Dillon, Balaji Lakshminarayanan, and Jasper Snoek. *Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift.* arXiv preprint arXiv:1906.02530, 2019.

[194] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami. *The Limitations of Deep Learning in Adversarial Settings.* In 2016 IEEE European Symposium on Security and Privacy (EuroS P), 2016.

[195] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Ulfar Erlingsson. *Scalable Private Learning with PATE.* In International Conference on Learning Representations, 2018.

[196] Nicholas R Parrillo. *Against the Profit Motive: The Salary Revolution in American Government, 1780-1940.* Yale University Press, 2013.

[197] Samir Passi and Solon Barocas. *Problem formulation and fairness.* In Proceedings of the Conference on Fairness, Accountability, and Transparency, pages 39–48, 2019.

[198] Remigijus Paulavičius and Julius Žilinskas. *Analysis of different norms and corresponding Lipschitz constants for global optimization.* Technological and Economic Development of Economy, 12:301–306, 01 2006. doi: 10.1080/13928619.2006.9637758.

[199] Martin Pawelczyk, Klaus Broelemann, and Gjergji Kasneci. *Learning model-agnostic counterfactual explanations for tabular data.* In Proceedings of The Web Conference 2020, pages 3126–3132, 2020.

[200] Martin Pawelczyk, Klaus Broelemann, and Gjergji. Kasneci. *On Counterfactual Explanations under Predictive Multiplicity.* In Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI), Proceedings of Machine Learning Research, 2020.

[201] Judea Pearl. *Causality: Models, Reasoning and Inference.* Cambridge University Press, USA, 2nd edition, 2009. ISBN 052189560X.

[202] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. *Scikit-learn: Machine Learning in Python.* Journal of Machine Learning Research, 12:2825–2830, 2011.

[203] Michael P Perrone and Leon N Cooper. *When networks disagree: Ensemble methods for hybrid neural networks.* Technical report, BROWN UNIV PROVIDENCE RI INST FOR BRAIN AND NEURAL SYSTEMS, 1992.

[204] Ronen Perry and Tal Z Zarsky. *May the Odds Be Ever in Your Favor: Lotteries in Law.* Ala. L. Rev., 66:1035, 2014.

[205] Robi Polikar. *Ensemble learning.* In Ensemble machine learning, pages 1–34. Springer,

2012.

[206] Rafael Poyiadzi, Kacper Sokol, Raul Santos-Rodriguez, Tijl De Bie, and Peter Flach. *FACE: feasible and actionable counterfactual explanations.* In Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, pages 344–350, 2020.

[207] U.S. President. *Exec. Order No. 13985 86 Fed. Reg. 7009, Advancing Racial Equity and Support for Underserved Communities Through the Federal Government.*

[208] PwC. *Managing the risks of machine learning and artificial intelligence models in the financial services industry*, 2020.

[209] Manish Raghavan and Solon Barocas. *Challenges for mitigating bias in algorithmic hiring.* Brookings. Retrieved February, 25:2020, 2019.

[210] Manish Raghavan, Solon Barocas, Jon Kleinberg, and Karen Levy. *Mitigating bias in algorithmic hiring: Evaluating claims and practices.* In Proceedings of the 2020 conference on fairness, accountability, and transparency, pages 469–481, 2020.

[211] Kaivalya Rawal, Ece Kamar, and Himabindu Lakkaraju. *Can I Still Trust You?: Understanding the Impact of Distribution Shifts on Algorithmic Recourses*, 2021.

[212] Xavier Renard, Thibault Laugel, and Marcin Detyniecki. *Understanding Prediction Discrepancies in Machine Learning Classifiers.* arXiv preprint arXiv:2104.05467, 2021.

[213] Michael L Rich. *Machine learning, automated suspicion algorithms, and the fourth amendment.* University of Pennsylvania Law Review, pages 871–929, 2016.

[214] Kit T Rodolfa, Erika Salomon, Lauren Haynes, Iván Higuera Mendieta, Jamie Larson, and Rayid Ghani. *Case study: predictive fairness to reduce misdemeanor recidivism through social service interventions.* In Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, pages 142–153, 2020.

[215] Kit T Rodolfa, Hemank Lamba, and Rayid Ghani. *Empirical observation of negligible fairness–accuracy trade-offs in machine learning for public policy.* Nature Machine Intelligence, 3(10):896–904, 2021.

[216] Cynthia Rudin. *Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead.* Nature Machine Intelligence, 1(5): 206–215, 2019.

[217] Emmanuel Saez and Stefanie Stantcheva. *Generalized social marginal welfare weights for optimal tax theory.* American Economic Review, 106(1):24–45, 2016.

[218] Omer Sagi and Lior Rokach. *Ensemble learning: A survey.* Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8(4):e1249, 2018.

[219] J Schuppe. *How facial recognition became a routine policing tool in America*, 2019.

[220] Andrew D Selbst and Solon Barocas. *The intuitive appeal of explainable machines.* Fordham L. Rev., 87:1085, 2018.

[221] Lesia Semenova, Cynthia Rudin, and Ronald Parr. *A study in Rashomon curves and volumes: A new perspective on generalization and model simplicity in machine learning.* arXiv preprint arXiv:1908.01755, 2019.

[222] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms.* Cambridge university press, 2014.

[223] Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. *Learn-*

ability, stability and uniform convergence. Journal of Machine Learning Research, 11, 2010.

[224] Shubham Sharma, Jette Henderson, and Joydeep Ghosh. *Certifai: Counterfactual explanations for robustness, transparency, interpretability, and fairness of artificial intelligence models.* arXiv preprint arXiv:1905.07857, 2019.

[225] Aditya Shinde et al. *Comparative Study of Regression Models and Deep Learning Models for Insurance Cost Prediction.* In ISDA, 2018.

[226] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. *Deep inside convolutional networks: Visualising image classification models and saliency maps.* arXiv preprint arXiv:1312.6034, 2013.

[227] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. *Deep inside convolutional networks: Visualising image classification models and saliency maps.* In International Conference on Learning Representations (ICLR), 2014.

[228] Ashudeep Singh and Thorsten Joachims. *Fairness of exposure in rankings.* In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 2219–2228, 2018.

[229] Justin Sirignano, Apaar Sadhwani, and Kay Giesecke. *Deep learning for mortgage risk.* CoRR, abs/1607.02470, 2016.

[230] Kacper Sokol and Peter A Flach. *Counterfactual explanations of machine learning predictions: opportunities and challenges for AI safety.* In SafeAI at AAAI, 2019.

[231] Congzheng Song, Thomas Ristenpart, and Vitaly Shmatikov. *Machine Learning Models That Remember Too Much.* In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 2017.

[232] Liwei Song, Reza Shokri, and Prateek Mittal. *Membership inference attacks against adversarially robust deep learning models.* In 2019 IEEE Security and Privacy Workshops (SPW). IEEE, 2019.

[233] Wenqing Sun, Bin Zheng, and Wei Qian. *Computer aided lung cancer diagnosis with deep learning algorithms.* In Medical imaging 2016: computer-aided diagnosis, volume 9785, page 97850Z. International Society for Optics and Photonics, 2016.

[234] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. *Axiomatic attribution for deep networks.* In International Conference on Machine Learning, pages 3319–3328. PMLR, 2017.

[235] Supreme Court of the United States. *Griggs v. Duke Power Co.* 401 U.S. 424, 1971.

[236] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. *Intriguing properties of neural networks*, 2013.

[237] tensorflow-determinism Python package. Available at: https://pypi.org/project/tensorflow-determinism/. Retrieved on 6/5/2020.

[238] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. *Robustness May Be at Odds with Accuracy.* In International Conference on Learning Representations, 2019.

[239] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. *Robustness may be at odds with accuracy.* In International Conference on Learning Representations, 2019.

[240] Kagan Tumer and Joydeep Ghosh. *Error correlation and error reduction in ensemble classifiers.* Connection science, 8(3-4):385–404, 1996.

[241] Berk Ustun, Alexander Spangher, and Yang Liu. *Actionable Recourse in Linear Classification.* In Conference on Fairness, Accountability, and Transparency, pages 10–19, 2019.

[242] Giorgio Valentini, Marco Muselli, and Francesca Ruffino. *Cancer recognition with bagged ensembles of support vector machines.* Neurocomputing, 56:461–466, 2004.

[243] Arnaud Van Looveren and Janis Klaise. *Interpretable counterfactual explanations guided by prototypes.* arXiv preprint arXiv:1907.02584, 2019.

[244] Sahil Verma and Julia Rubin. *Fairness definitions explained.* In 2018 ieee/acm international workshop on software fairness (fairware), pages 1–7. IEEE, 2018.

[245] Sahil Verma, John Dickerson, and Keegan Hines. *Counterfactual Explanations for Machine Learning: A Review.* arXiv preprint arXiv:2010.10596, 2020.

[246] James Vincent. *NYPD used facial recognition to track down Black Lives Matter activist.* The Verge, August, 2020.

[247] Sandra Wachter, Brent Mittelstadt, and Chris Russell. *Counterfactual explanations without opening the black box: Automated decisions and the GDPR.* Harvard Journal of Law & Technology, 31(2):841–887, 2018.

[248] Tong Wang. *Gaining free or low-cost interpretability with interpretable partial substitute.* In International Conference on Machine Learning, pages 6505–6514. PMLR, 2019.

[249] Yibo Wang and Wei Xu. *Leveraging deep learning with LDA-based text analytics to detect automobile insurance fraud.* Decision Support Systems, 105:87–95, 2018.

[250] Yu-Xiang Wang, Jing Lei, and Stephen E. Fienberg. *Learning with Differential Privacy: Stability, Learnability and the Sufficiency and Necessity of ERM Principle.* Journal of Machine Learning Research, 17(183):1–40, 2016. URL `http://jmlr.orgrs/v17/15-313.htl`.

[251] Zifan Wang, Haofan Wang, Shakul Ramkumar, Matt Fredrikson, Piotr Mardziel, and Anupam Datta. *Smoothed Geometry for Robust Attribution.* Neurips, 2020.

[252] Zifan Wang, Matt Fredrikson, and Anupam Datta. *Boundary Attributions Provide Normal (Vector) Explanations.* ArXiv, abs/2103.11257, 2021.

[253] Michael Wick, swetasudha panda, and Jean-Baptiste Tristan. *Unlocking Fairness: a Trade-off Revisited.* In Advances in Neural Information Processing Systems, volume 32, 2019.

[254] Eric Wong and Zico Kolter. *Provable Defenses against Adversarial Examples via the Convex Outer Adversarial Polytope.* In Proceedings of the 35th International Conference on Machine Learning (ICML), 2018.

[255] Han Xiao, Kashif Rasul, and Roland Vollgraf. *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms*, 2017.

[256] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L. Yuille. *Improving Transferability of Adversarial Examples With Input Diversity.* In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.

[257] An Yan and Bill Howe. *Fairness in practice: a survey on equity in urban mobility.* A Quarterly bulletin of the Computer Society of the IEEE Technical Committee on Data Engineering, 42(3), 2020.

[258] Linyi Yang, Eoin M Kenny, Tin Lok James Ng, Yi Yang, Barry Smyth, and Ruihai Dong. *Generating Plausible Counterfactual Explanations for Deep Transformers in Financial Text Classification.* arXiv preprint arXiv:2010.12512, 2020.

[259] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha. *Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting.* In 2018 IEEE 31st Computer Security Foundations Symposium (CSF), 2018.

[260] Samuel Yeom and Matt Fredrikson. *Individual Fairness Revisited: Transferring Techniques from Adversarial Robustness.* In IJCAI, 2020.

[261] Samuel Yeom, Irene Giacomelli, Alan Menaged, Matt Fredrikson, and Somesh Jha. *Overfitting, robustness, and malicious algorithms: A study of potential causes of privacy risk in machine learning.* J. Comput. Secur., 28(1), 2020.

[262] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. *Fairness Constraints: Mechanisms for Fair Classification.* In Artificial Intelligence and Statistics, pages 962–970, 2017.

[263] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P Gummadi. *Fairness constraints: A flexible approach for fair classification.* The Journal of Machine Learning Research, 20(1):2737–2778, 2019.

[264] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. *Understanding deep learning requires rethinking generalization.* CoRR, abs/1611.03530, 2016.

[265] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. *Theoretically principled trade-off between robustness and accuracy.* In International Conference on Machine Learning, pages 7472–7482. PMLR, 2019.

[266] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. *Theoretically Principled Trade-off between Robustness and Accuracy.* In International Conference on Machine Learning (ICML), pages 7472–7482. PMLR, 2019.

[267] Zhi-Hua Zhou, Jianxin Wu, and Wei Tang. *Ensembling neural networks: many could be better than all.* Artificial intelligence, 137(1-2):239–263, 2002.

Appendix A

Leave-one-Out Unfairness Appendix

A.1 Proofs

We present the full proofs from Section 2.

Proposition A.1. *Let h be a learning rule optimizing 0-1 loss and $\epsilon(m)$ be a montonically-decreasing function such that $\mathrm{LUF}(h, S, x) \leq \epsilon(n)$ for all $S \sim \mathcal{D}^m$ and x. Then h is on-average leave-one-out stable with rate at most $\epsilon(m)$.*

Proof. We prove the case for binary classification. The result generalizes to multiclass problems in a straighforward fashion. Note that because LUF is bounded for all S, we can disregard the expectation over S in the definition of LOO-stability, and assume that the randomness in the expectations comes from the learning rule h exclusively. By linearity of expectation, we have that,

$$\begin{aligned}\mathbb{E}[|\ell(h_S, z_i) - \ell(h_{S^{(\setminus i)}}, z_i)|] &= \Pr[h_S(z_i) \neq h_{S^{(\setminus i)}}(z_i)] \\ &= |\Pr[h_S(z_i) = 1] - \Pr[h_{S^{(\setminus i)}}(z_i) = 1]|\end{aligned}$$

Now, assuming,

$$\forall x, S, \max_i |\Pr[h_S(x) = 1] - \Pr[h_{S^{(\setminus i)}}(x) = 1]| \leq \epsilon(m)$$

we have:

$$\begin{aligned}&\frac{1}{m}\sum_{i=1}^{m} \mathop{\mathbb{E}}_{S\sim\mathcal{D}^n}[|\ell(h_S, z_i) - \ell(h_{S^{(\setminus i)}}, z_i)|] \\ \leq &\max_S \frac{1}{m}\sum_{i=1}^{m} \mathbb{E}[|\ell(h_S, z_i) - \ell(h_{S^{(\setminus i)}}, z_i)|]\end{aligned}$$

The result follows by noting that each term in the above sum is bounded by $\epsilon(m)$.

□

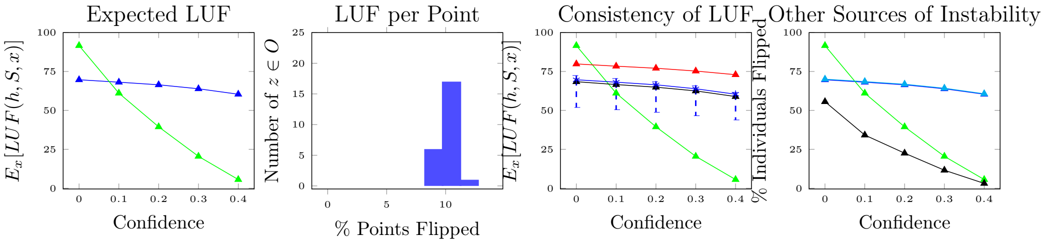


Figure A.1: Additional experiments on the LFW dataset, with identical setup to the results presented in the paper, but with a static group of face pairs encountered during training. From left to right, we have: the experiments presented in Section 5 concerning the extent of LUF in deep models (first two graphs), experiments from Section 5 showing the consistency of leave-one-out unfair behavior across different model architectures and seeds, and experiments showing the effect of instability from other sources discussion in Section 6.

Proposition A.2. *Let h be an (ϵ, δ)-differentially private learning rule, and $x \sim \mathcal{D}$ be a point. Then* $\mathrm{LUF}(h, x) \leq e^{\epsilon} - 1 + \delta$.

Proof. We prove the case where h produces binary classifiers. The extension to multi-class learning is straightforward. The result follows from a general property of differentially-private algorithms [73, Lemma 6] which is that when h_S ranges in $[0, 1]$,

$$|\mathbb{E}[h_S(x)] - \mathbb{E}[h_{S^{(\setminus i)}}(x)]| \leq e^{\epsilon} - 1 + \delta$$

Noting that $\mathbb{E}[h_S(x)] = \Pr[h_S(x) = 1]$, the result follows.

□

A.2 Additional LFW Results

We present additional experiments on the LFW dataset, with identical setup to the results presented in the paper, with the exception of the training set face pair generation process. In this setup, the model is trained on a static set of face pairs common across all models, as opposed to a being trained with a generator creating random face pairs that may differ on each training run. This static setup along with the rest of the precautions taken in all our experiments ensures that all possible sources of instability are controlled, aside from leave-one-out unfairness. One other difference in this set of experiments is that we sample 25 points to remove from the dataset, as opposed to 50 as in the results presented in the main paper. The results are qualitatively similar to the results presented in the paper, and still show far greater expected LUF than any other models presented, with LUF of approximately 69%. However, due to memory constraints, a comparatively small set of pairs of faces from LFW can be contained statically in memory, and the accuracy of the model suffers: the accuracy of h_S in this setup is 76%, with a generalization error of 22%.

A.3 Datasets

The German Credit data set consists of individuals financial data, with a binary response indicating their creditworthiness. There are 1000 points, and 20 attributes. We one-hot encode the data to get 61 features, and standardize the data to zero mean and unit variance using SKLearn Standard scaler. We partitioned the data intro a training set of 700, a leave-one-out-set of 100, and a test set of 200.

The Adult dataset consists of a subset of publicly-available US Census data, binary response indicating annual income of > 50k. There are 14 attributes, which we one-hot encode to get 96 features. We normalize the numerical features to have values between 0 and 1. After removing instances with missing values, there are $30,162$ examples which we split into a training set of 14891, a leave one out set of 100, and a test set of 1501 examples.

The Seizure dataset comprises time-series EEG recordings for 500 individuals, with a binary response indicating the occurrence of a seizure. This is represented as 11500 rows with 178 features each. We split this into 7,950 train points and 3,550 test points. We standardize the numeric features to zero mean and unit variance.

Fashion MNIST contains images of clothing items, with a multilabel response of 10 classes. There are 60000 training examples and 10000 test examples. We pre-process the data by normalizing the numerical values in the image array to be between 0 and 1.

The Labeled Faces in the Wild dataset (LFW) consists of 13,000 cropped images of 1,680 individuals' faces, with a multiclass label of 1,680 classes, corresponding to which individual is in what image. We pre-process the images by normalizing the numerical values in the image array to be between 0 and 1. Since the model that we use on the data is a face-matching model, we create a training set of pairs of images from the processed LFW. First, we split the original LFW dataset in a training set and test set, of sizes 6,873 and 2,291. For the results presented in the main paper, we use a data generator to create 6,873 pairs of images from the training set on each epoch. These pairs of images have a 50% match rate (that is, 50% of the pairs are of the same individual, and 50% are not). For the results for LFW presented in the supplementary material, we generate a static training set of 6,873 face pairs that stay consistent epoch to epoch. Note that this results in many fewer unique face pairs seen by the face-matching algorithm. For the test set in both the main paper and the supplementary material, we generate a static 2,291 pairs of images from the test set, again with 50% match rate.

A.4 Calculating LUF in All Experiments

We provide a description of how we calculated (an approximation of) LUF in our experiments. Given predictions of the entire dataset for both h_S and $h_{S(\setminus i)}$ models: For binary classification models, for each $h_{S(\setminus i)}$, and for h_S, we calculate whether the output is class 1 or 0. We then take the difference in binary predictions from the baseline model and $h_{S(\setminus i)}$, for each of the 100 $h_{S(\setminus i)}$. We choose the maximum difference over all $h_{S(\setminus i)}$ for each point (i.e., searching to see if the removal of *any* point removed results in a change in prediction for an individual in the distribution.) This approximates the leave-one-out unfairness for each x in the dataset, in the setting of a deterministic learning rule, as described in the main paper. Note that the approximation arises from the fact that we sample 100 points at random with which to create $h_{S\setminus i}$, rather than creating a different model for each point in the dataset. We then divide the number of individuals experiencing LUF by the size of the dataset to calculate the expected LUF over the dataset.

For multiclass problems, we follow a similar procedure, except that we calculate the probabilities that $h_{S(\setminus i)}$ and h_S output a given class for each class, compute the differences between these probabilities, take the maximum over k classes, and then proceed as in the binary case.

Finally, for calculating the effects of random seed and architecture on unfair arbitrariness as displayed in the discussion, we follow the exact same procedure as above, but where $h_{S(\setminus i)}$ is a model trained on a different seed, or in the architecture results, on a different architecture.

A.5 Experimental Setup Further Details

For German Credit and Seizure datasets, we trained all models in the paper with three hidden layers, of size 100, 32, and 16, over 100 epochs. The inner activations are ReLu, and the final activation is Sigmoid. The model is trained with binary crossentropy. We used the Adam optimizer with the default parameters used by Keras. The linear models for both datasets are trained over 100 epochs with a batch size of 32. For the German Credit models in Section 3, and the random smoothing experiments in Section 4, we use a batch size of 32. For the adversarially trained models, we use a batch size of 4. For the models used to compare the variance of LUF over different architectures, we train a one hidden layer of size 100 with the same hyperparameters as described for the main model, and a 3-hidden layer model with layer sizes 64, 16, and 8, again with the other hyperparamters kept constant.

For the Adult dataset, our main model was one hidden layer of size 200, over 50 epochs with a batch size of 128. The activations, loss, and optimizer were the same as those for German Credit. The linear models for Adult were also trained with a batch size of 128 over 50 epochs. For the adversarial experiments, we use a batch size of 32. For the models used to compare the variance of LUF over different architectures, we train a one hidden layer of size 100 with the same hyperparameters as described for the main model, and a 3-hidden layer model with layer sizes 128, 32, and 16, again with the other hyperparamters kept constant.

For the Seizure dataset, we trained the main models in the paper with three hidden layers, of size 128, 32, and 16, over 100 epochs. All models, including the linear models, were trained with a batch size of 128. The activations, loss, and optimizer were the same as those for German Credit. For the models used to compare the variance of LUF over different architectures, we train models with the same architecture as the German Credit datasets, with the rest of the parameters kept the same as in the main experiments.

For the FMNIST data set, for all models in the paper, we used a LeNet[152] architecture modified for the size of the data, trained with dropout: this consists of 2 convolutional layers with 20 and 50 channels respectively, each followed by a max pooling layer, and finally a dense layers with 200 neurons. We train with SGD, batch size 128, and 50 epochs. For the linear data, we used a batch size of 128 and 50 epochs as well. For the adversarially trained models, we use a batch size of 32. The models over FMNIST with varying architecture included a the same model described above, but trained without dropout, and a shallower model with the middle layer (along with the corresponding pooling and convolution layers) removed.

For LFW, we train a face-matching model, that takes a pair of images and outputs a binary label whether or not the pair of images are of the same individual. The LFW face-matching model consists of a concatenation layer composing the two input images, a 4-layer convolutional stack, followed by a dense layer, and a sigmoid output. It is trained with the Adam optimizer with the default learning rate, batch size of 128, over 50 epochs. To compare the effect of LUF across architectures, we train use a ResNet50 model, pre-trained on ImageNet weights from Keras, modified to take two images as input and have a Sigmoid output. We also compare the effects of LUF on a model with the same architecture as the original one described, but with doubled filter sizes for the convolutions. All other models are trained with the same hyperparameters as the original model.

A.6 Experimental Setup For Decision Boundary Images

To generate the pictures in Figure 1, we train a model 3 Relu layers, each with 1000 neurons, trained on 100 uniform-random points with Bernoulli labels.

Appendix B

Selective Ensembles Appendix

Proofs

B.0.1 Proof of Theorem 1

Theorem B.1. *Let $H : \mathbf{X} \to \{-1, 1\} = \text{sign}(h)$ be a binary classifier and $g : \mathbf{X} \to \mathbb{R}$ be an unrelated function that is bounded from above and below, continuous, and piecewise differentiable. Then there exists another binary classifier $\hat{H} = \text{sign}(\hat{h})$ such that for any $\epsilon > 0$,*

$$\forall x \in \mathbf{X}\ . \qquad \textit{1. } \hat{H}(x) = H(x) \qquad \textit{2. } \inf_{x':H(x')\neq H(x)} \Big\{ ||x - x'|| \Big\} > {}^{\epsilon}/_{2} \implies \nabla \hat{h}(x) = \nabla g(x)$$

Proof. We divide $\mathbf{X}$ into regions $\{I_1 .... I_k\}$ determined by the decision boundaries of H. That is, each I_i represents an area where H predicts a certain class (-1 or 1) up until the boundaries to predict a different class.

Recall we are given a function $g : R^n \to R$ which is bounded from above and below. We create a set of functions $\hat{g_{I_i}}(x) : x \in I_i \to R$ such that

$$\hat{g}_{I_i}(x) = \begin{cases} g(x) - \inf_x g(x) \text{ if } H(I_i) = 1 \\ g(x) - (\sup_x g(x) + c) \text{ if } H(I_i) = -1 \end{cases}$$

Where c is some constant greater than zero. Additionally, let $d(x)$ be the ℓ_2 distance from x to the nearest decision boundary of h. Then, we define $\hat{h}$ to be:

$$\hat{h} = \begin{cases} \hat{g}_{I_i}(x) \text{ for x } \in I_i \text{ if } d(x) > \frac{\epsilon}{2} \\ \hat{g}_{I_i}(x) \times \frac{2d(x)}{\epsilon} \text{ for x } \in I_i \text{ if } d(x) \leq \frac{\epsilon}{2} \end{cases}$$

And, as described above, we define $\hat{H} = \text{sign}(h)$. First, we show that $\hat{H}(x) = H(x) \forall x \in R^n$. Consider the case when $H(x) = 1$, and $d(x) > \frac{\epsilon}{2}$. By construction, $\hat{H}(x) = \text{sign}(h(x)) = \text{sign}(\hat{g}_{I_i}(x)) = \text{sign}(g(x) - \inf_x g(x))$, where $x \in$ region I_i where $H(I_i) = 1$. By definition of the infimum, $g(x) - \inf_x g(x) >= 0$, and thus $\text{sign}(g(x) - \inf_x g(x)) = 1$, and $\hat{H}(x) = 1 = H(x)$. Note that in the case where $d(x) \leq \frac{\epsilon}{2}$, we can follow the same argument as the value of $\hat{h(x)}$ only differs by a positive constant. A similar argument follows for the case where $H(x) = -1$; thus, $\hat{H}(x) = H(x) \forall x \in R^n$.

Secondly, we show that $\bigtriangledown \hat{h}(x) = \bigtriangledown g(x) \forall x \text{where} d(x) > \frac{\epsilon}{2}$. Consider the case where $H(x) = 1$. By construction, $\hat{h}(x) = \hat{g}_{I_i}(x) = g(x) - \inf_x g(x)$. Note that this means $\hat{h}(x) = g(x)$ plus a constant, so the gradient of the two functions at x is the same. A symmetric argument holds for the case where $H(x) = -1$.

It remains to prove that $\hat{h}$ is continuous and piece-wise differentiable, in order to be a realizable as a Relu-network. See that $\hat{h}$ is piece-wise differentiable as g is, as required, which means that $\hat{g}_i$ are continuous well, and also $\hat{g}_i(x) \times \frac{d(x)}{\epsilon} \forall i$, which comprise $\hat{h}$. To see that $\hat{h}$ is continuous, consider the case where $d(x) = \epsilon$ for some x: then $\hat{g}_i(x) \times \frac{d(x)}{\epsilon} = \hat{g}_i(x) \times \frac{\epsilon}{\epsilon} = \hat{g}_i(x)$. Additionally, consider the case where $d(x) = 0$, i.e. x is on a decision boundary of $h(x)$, between two regions I_i, I_j. Then $\hat{h}(x) = \hat{g}_i(x) \times \frac{d(x)}{\epsilon} = \hat{g}_i(x) \times 0 = 0 = \hat{g}_j(x) \times 0 = \hat{g}_j(x)$. This shows that $\hat{h}$ has continuous behavior between transitioning from case to case of its output. Further, when $d(x) \neq 0, \epsilon$, then $\hat{h}$ is continuous since g is continuous as required, and thus $\hat{g}_i$ are continuous $\forall i$, as well as $\hat{g}_i \times \frac{2d(x)}{\epsilon}$, as the multiplied expression is simply a constant. □

B.0.2 Proof of Theorem 2

Theorem B.2. *An ensemble model $h_{S,r,n}$ will return the majority prediction $g_{h,S,R}(x)$ on a point x, or abstain, with probability 1-α. In other words, with probability at most α, they will return a value that is* not *$g_{h,S,R}(x)$.*

Proof. $En(h, S, R, n)$ is an ensemble of n models. By the definition of the Predict algorithm, $En(h, S, R, n)$ gathers a vector of class counts the prediction for x from each model in the ensemble. Let the class with the highest count be c_A, with counts n_A, and the class with the second highest count be called c_B, with counts n_B. In order for the ensemble to predict, the ensemble models runs a two-sided hypothesis test to ensure that $Pr[n_A \sim \text{Binomial}(n_A + n_B, 0.5)] < \alpha$, i.e. that A is the true majority prediction over R. See that

$$P[g_{h,S,R}(x) \neq c_A \wedge En(h, S, R, n)(x) = c_A]$$

$$= P[g_{h,S,R}(x) \neq c_A] P[En(h, S, R, n) \text{does not abstain} | g_{h,S,R}(x) \neq c_A]$$

$$\leq P[En(h, S, R, n) \text{ does not abstain} | g_{h,S,R}(x) \neq c_A]$$

By Hung and Fithian[], we have that

$$\leq P[En(h, S, R, n) \text{ does not abstain} | g_{h,S,R}(x) \neq c_A] = \alpha$$

Thus,

$$P[g_{h,S,R}(x) \neq c_A \wedge En(h, S, R, n)(x) = c_A] \leq \alpha$$

□

Datasets

The German Credit and Taiwanese data sets consist of individuals financial data, with a binary response indicating their creditworthiness. For the German Credit dataset, there are 1000 points, and 20 attributes. We one-hot encode the data to get 61 features, and standardize the data to zero mean and unit variance using SKLearn Standard scaler. We partitioned the data intro a training set of 700 and a test set of 200. The Taiwanese credit dataset has 30,000 instances with 24 attributes. We one-hot encode the data to get 32 features and normalize the data to be between zero and one. We partitioned the data intro a training set of 22500 and a test set of 7500.

The Adult dataset consists of a subset of publicly-available US Census data, binary response indicating annual income of > 50k. There are 14 attributes, which we one-hot encode to get 96 features. We normalize the numerical features to have values between 0 and 1. After removing instances with missing values, there are $30,162$ examples which we split into a training set of 14891, a leave one out set of 100, and a test set of 1501 examples.

The Seizure dataset comprises time-series EEG recordings for 500 individuals, with a binary response indicating the occurrence of a seizure. This is represented as 11500 rows with 178 features each. We split this into 7,950 train points and 3,550 test points. We standardize the numeric features to zero mean and unit variance.

Fashion MNIST contains images of clothing items, with a multilabel response of 10 classes. There are 60000 training examples and 10000 test examples. We pre-process the data by normalizing the numerical values in the image array to be between 0 and 1.

The Warfain dataset is collected by the International Warfarin Pharmacogenetics Consortium [120] about patients who were prescribed warfarin. We removed rows with missing values, 4819 patients remained in the dataset. The inputs to the model are demographic (age, height, weight, race), medical (use of amiodarone, use of enzyme inducer), and genetic (VKORC1, CYP2C9) attributes. Age, height, and weight are real-valued and were scaled to zero mean and unit variance. The medical attributes take binary values, and the remaining attributes were one-hot encoded. The output is the weekly dose of warfarin in milligrams, which we encode as "low", "medium", or "high", following [120]

Model Architecture and Hyper-Parameters

The German Credit and Seizure models have three hidden layers, of size 128, 64, and 16. Models on the Adult dataset have one hidden layer of 200 neurons. The FMNIST model is a modified LeNet architecture [152]. This model is trained with dropout. The LFW face-matching model consists of a concatenation layer composing the two input images, a 4-layer convolutional stack, followed by a dense layer, and a Sigmoid output. German Credit, Adult, and Seizure models are trained for 100 epochs; FMNIST and LFW models are trained for 50. German Credit models are trained with a batch size of 32, FMNIST 64, and Adult, Seizure, and LFW used batch sizes of 128. German Credit, Adult, Seizure a

		portion of test data with $p_{\text{flip}} > 0$						
Randomness	n	Ger. Credit	Adult	Seizure	Warfarin	Tai. Credit	FMNIST	Colon
RS	1	0.585	0.089	0.061	0.0	0.080	0.115	0.066
RS	(5, 10, 15, 20)	0.0	0.0	0.0	0.0	0.0	0.0	0.0
LOO	1	0.250	0.068	0.060	0.028	0.030	0.056	0.068
LOO	(5, 10, 15, 20)	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table B.1: The percentage of points with disagreement between at least one pair of models ($p_{\text{flip}} > 0$) trained with different random seeds (RS) or leave-one-out differences in training data, for singleton models ($n = 1$) and selective ensembles ($n > 1$). Results for selective ensembles all selective ensembles are shown together, as they all have no disagreement. Note that these results are for $\alpha = 0.01$. But this different α also leads to zero disagreement between predicted points.

				mean accuracy (abstain as error) / std. dev				
$\mathcal{S}$	n	Ger. Credit	Adult	Seizure	Wafarin	Tai. Credit	FMNIST	Colon
RS	5	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
RS	10	$.461 \pm .016$	$.807 \pm 1e - 3$	$.945 \pm 2e - 3$	$.646 \pm 3e - 3$	$.788 \pm 2e - 3$	$.870 \pm 5e - 3$	$.902 \pm 2e - 3$
RS	15	$.589 \pm .015$	$.822 \pm 8e - 4$	$.961 \pm 1e - 3$	$.661 \pm 3e - 3$	$.802 \pm 9e - 4$	$.890 \pm 2e - 3$	$.915 \pm 1e - 3$
RS	20	$.593 \pm .011$	$.822 \pm 7e - 4$	$.961 \pm 8e - 4$	$.662 \pm 1e - 3$	$.803 \pm 9e - 4$	$.991 \pm 1e - 3$	$.926 \pm 1e - 3$
LOO	5	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
LOO	10	$.618 \pm .017$	$.818 \pm 1e - 3$	$.947 \pm 4e - 3$	$.674 \pm 2e - 3$	$.807 \pm 1e - 3$	$.904 \pm 6e - 4$	$.901 \pm 2e - 3$
LOO	15	$.656 \pm .017$	$.828 \pm 1e - 3$	$.963 \pm 1e - 3$	$.678 \pm 9e - 4$	$.812 \pm 9e - 4$	$.908 \pm 1e - 3$	$.912 \pm 2e - 3$
LOO	20	$.661 \pm .018$	$.829 \pm 7e - 4$	$.964 \pm 1e - 3$	$.678 \pm 7e - 4$	$.812 \pm 8e - 4$	$.909 \pm 6e - 4$	$.912 \pm 2e - 3$

				mean abstention rate / std dev				
$\mathcal{S}$	n	Ger. Credit	Adult	Seizure	Warfarin	Tai. Credit	FMNIST	Colon
RS	5	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
RS	10	$.449 \pm .021$	$.068 \pm 2e - 3$	$.045 \pm 2e - 3$	$.078 \pm 5e - 3$	$.063 \pm 2e - 3$	$.087 \pm 8e - 3$	$.050 \pm 3e - 3$
RS	15	$.278 \pm .017$	$.041 \pm 1e - 3$	$.025 \pm 1e - 3$	$.049 \pm 3e - 3$	$.037 \pm 1e - 3$	$.055 \pm 2e - 3$	$.030 \pm 2e - 3$
RS	20	$.270 \pm .015$	$.040 \pm 1 - e3$	$.024 \pm 1e - 3$	$.047 \pm 2e - 3$	$.036 \pm 1e - 3$	$.054 \pm 9e - 4$	$.038 \pm 1e - 3$
LOO	5	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
LOO	10	$.215 \pm .030$	$.049 \pm 2e - 3$	$.045 \pm 5e - 3$	$.027 \pm 2e - 3$	$.025 \pm 1e - 3$	$.029 \pm 1e - 3$	$.054 \pm 2e - 3$
LOO	15	$.144 \pm 0.040$	$.030 \pm 2e - 3$	$.026 \pm 1e - 3$	$.017 \pm 2e - 3$	$.017 \pm 2e - 3$	$.021 \pm 3e - 3$	$.035 \pm 2e - 3$
LOO	20	$.135 \pm .040$	$.029 \pm 1e - 3$	$.025 \pm 1e - 3$	$.017 \pm 1e - 3$	$.017 \pm 2e - 3$	$.019 \pm 1e - 3$	$.035 \pm 3e - 3$

Table B.2: Accuracy (above) and abstention rate (below) of selective ensembles with $n \in \{5, 10, 15, 20\}$ constituents. Results are averaged over 24 models, standard deviation is presented. Note that these results are for **alpha=0.01**.

Metrics

B.0.3 SSIM

Experimental Results for $\alpha = 0.01$

Selective Ensembling Full Results

Explanation Consistency Full Results

B.0.4 Attributions

mean accuracy (abstain as error) / std. dev

$\mathcal{S}$	n	Ger. Credit	Adult	Seizure	Wafarin	Tai. Credit	FMNIST	Colon
RS	5	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
RS	10	$.461 \pm .016$	$.807 \pm 1e-3$	$.945 \pm 2e-3$	$.646 \pm 3e-3$	$.788 \pm 2e-3$	$.870 \pm 5e-3$	$.902 \pm 2e-3$
RS	15	$.589 \pm .015$	$.822 \pm 8e-4$	$.961 \pm 1e-3$	$.661 \pm 3e-3$	$.802 \pm 9e-4$	$.890 \pm 2e-3$	$.915 \pm 1e-3$
RS	20	$.593 \pm .011$	$.822 \pm 7e-4$	$.961 \pm 8e-4$	$.662 \pm 1e-3$	$.803 \pm 9e-4$	$.991 \pm 1e-3$	$.926 \pm 1e-3$
LOO	5	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
LOO	10	$.618 \pm .017$	$.818 \pm 1e-3$	$.947 \pm 4e-3$	$.674 \pm 2e-3$	$.807 \pm 1e-3$	$.904 \pm 6e-4$	$.901 \pm 2e-3$
LOO	15	$.656 \pm .017$	$.828 \pm 1e-3$	$.963 \pm 1e-3$	$.678 \pm 9e-4$	$.812 \pm 9e-4$	$.908 \pm 1e-3$	$.912 \pm 2e-3$
LOO	20	$.661 \pm .018$	$.829 \pm 7e-4$	$.964 \pm 1e-3$	$.678 \pm 7e-4$	$.812 \pm 8e-4$	$.909 \pm 6e-4$	$.912 \pm 2e-3$

mean abstention rate / std dev

$\mathcal{S}$	n	Ger. Credit	Adult	Seizure	Warfarin	Tai. Credit	FMNIST	Colon
RS	5	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
RS	10	$.449 \pm .021$	$.068 \pm 2e-3$	$.045 \pm 2e-3$	$.078 \pm 5e-3$	$.063 \pm 2e-3$	$.087 \pm 8e-3$	$.050 \pm 3e-3$
RS	15	$.278 \pm .017$	$.041 \pm 1e-3$	$.025 \pm 1e-3$	$.049 \pm 3e-3$	$.037 \pm 1e-3$	$.055 \pm 2e-3$	$.030 \pm 2e-3$
RS	20	$.270 \pm .015$	$.040 \pm 1-e3$	$.024 \pm 1e-3$	$.047 \pm 2e-3$	$.036 \pm 1e-3$	$.054 \pm 9e-4$	$.038 \pm 1e-3$
LOO	5	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
LOO	10	$.215 \pm .030$	$.049 \pm 2e-3$	$.045 \pm 5e-3$	$.027 \pm 2e-3$	$.025 \pm 1e-3$	$.029 \pm 1e-3$	$.054 \pm 2e-3$
LOO	15	$.144 \pm 0.040$	$.030 \pm 2e-3$	$.026 \pm 1e-3$	$.017 \pm 2e-3$	$.017 \pm 2e-3$	$.021 \pm 3e-3$	$.035 \pm 2e-3$
LOO	20	$.135 \pm .040$	$.029 \pm 1e-3$	$.025 \pm 1e-3$	$.017 \pm 1e-3$	$.017 \pm 2e-3$	$.019 \pm 1e-3$	$.035 \pm 3e-3$

Table B.3: Accuracy and abstention rate of selective ensembles with $n \in \{5, 10, 15, 20\}$ constituents for demographic groups in each dataset. We have ensembles sampled across different training sets with a difference of one data-point (LOO) and ensembles sampled over different random seeds (RF). At the top of each row, we have the baseline, single-model accuracy across groups for all datasets, averaged over 500 models. For the Adult and Taiwanese Credit datasets, we have groups of individuals identified in the data as male or female as groups; in German Credit, we have individuals above or below median age in the dataset; in the Warfarin dosing dataset, we have individuals identified as Black, White, and Asian in the dataset as groups.

mean accuracy (abstain as error) / std. dev

$\mathcal{S}$	n	Ger. Credit	Adult	Seizure	Warfarin	Tai. Credit	FMNIST	Colon
RS	5	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
RS	10	$.576 \pm .013$	$.820 \pm 8e-4$	$.960 \pm 1e-3$	$.660 \pm 2e-3$	$.800 \pm 1e-3$	$.888 \pm 2e-3$	$.914 \pm 1e-3$
RS	15	$.636 \pm .017$	$.827 \pm 5e-4$	$.965 \pm 1e-3$	$.668 \pm 2e-3$	$.807 \pm 9e-4$	$.897 \pm 2e-3$	$.919 \pm 1e-3$
RS	20	$.664 \pm .014$	$.830 \pm 5e-4$	$.967 \pm 9e-4$	$.670 \pm 3e-3$	$.810 \pm 8e-4$	$.902 \pm 1e-3$	$.921 \pm 1e-3$
LOO	5	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
LOO	10	$.653 \pm .017$	$.827 \pm 1e-3$	$.962 \pm 2e-3$	$.677 \pm 1e-3$	$.812 \pm 1e-3$	$.909 \pm 4e-4$	$.912 \pm 1e-3$
LOO	15	$.678 \pm .014$	$.832 \pm 7e-4$	$.968 \pm 9e-4$	$.679 \pm 9e-4$	$.814 \pm 9e-4$	$.910 \pm 1e-3$	$.916 \pm 2e-3$
LOO	20	$.689 \pm .014$	$.834 \pm 7e-4$	$.970 \pm 1e-3$	$.680 \pm 7e-4$	$.815 \pm 8e-4$	$.911 \pm 4e-4$	$.918 \pm 8e-4$

mean abstention rate / std dev

$\mathcal{S}$	n	Ger. Credit	Adult	Seizure	Warfarin	Tai. Credit	FMNIST	Colon
RS	5	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
RS	10	$.291 \pm .014$	$.043 \pm 1e-3$	$.02 \pm 1e-3$	$.050 \pm 3e-3$	$.039 \pm 2e-3$	$.059 \pm 2e-3$	$.032 \pm 3e-3$
RS	15	$.205 \pm .020$	$.032 \pm 1e-3$	$.018 \pm 1e-3$	$.037 \pm 3e-3$	$.028 \pm 1e-3$	$.042 \pm 2e-3$	$.023 \pm 2e-3$
RS	20	$.165 \pm .015$	$.024 \pm 7-e4$	$.014 \pm 7e-4$	$.031 \pm 4e-3$	$.023 \pm 8e-4$	$.036 \pm 1e-3$	$.019 \pm 2e-3$
LOO	5	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
LOO	10	$.151 \pm .041$	$.032 \pm 2e-3$	$.027 \pm 2e-3$	$.018 \pm 2e-3$	$.017 \pm 2e-3$	$.020 \pm 5e-4$	$.036 \pm 3e-3$
LOO	15	$.105 \pm 0.034$	$.022 \pm 1e-3$	$.019 \pm 1e-3$	$.013 \pm 2e-3$	$.013 \pm 2e-3$	$.016 \pm 2e-3$	$.027 \pm 2e-3$
LOO	20	$.079 \pm .029$	$.018 \pm 1e-3$	$.015 \pm 1e-3$	$.011 \pm 2e-3$	$.010 \pm 1e-3$	$.012 \pm 8e-4$	$.023 \pm 2e-3$

Table B.4: Accuracy (above) and abstention rate (below) of selective ensembles with $n \in \{5, 10, 15, 20\}$ constituents. Results are averaged over 24 models, standard deviation is presented. Note that these results are for **alpha=0.05**, which are presented in the main paper.

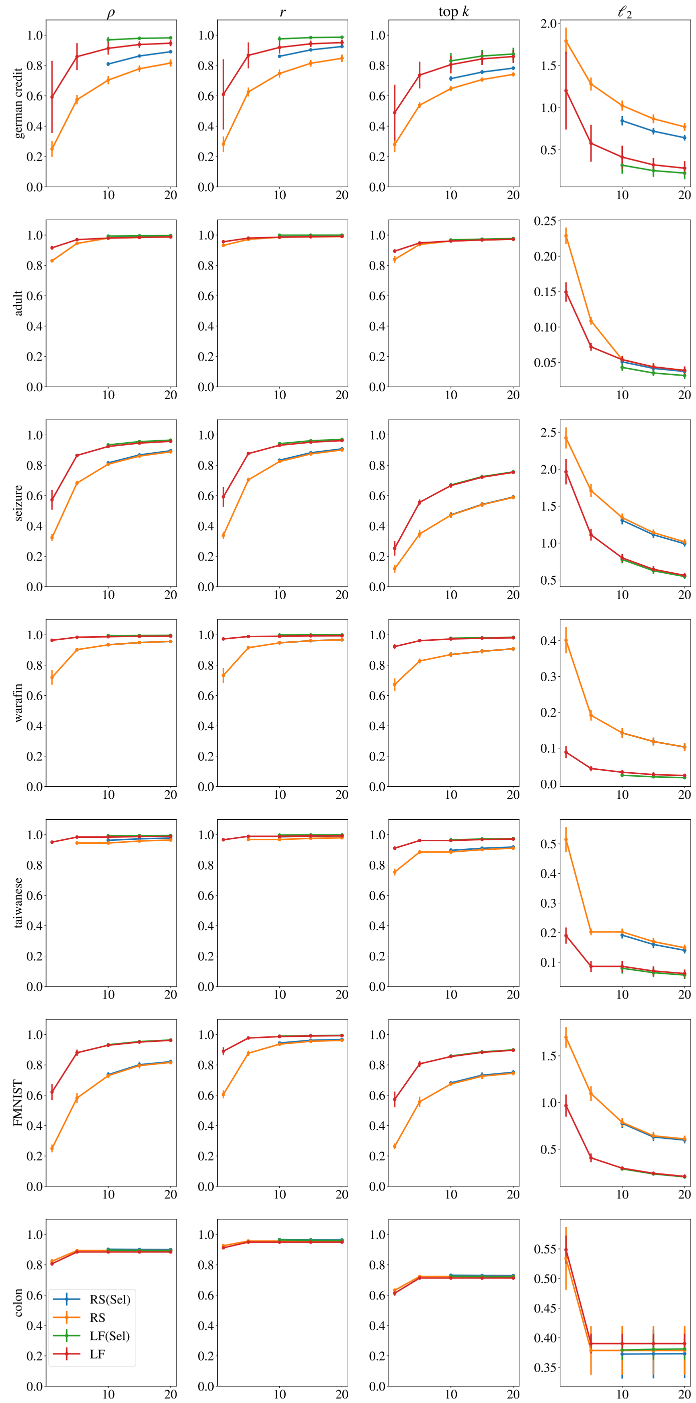
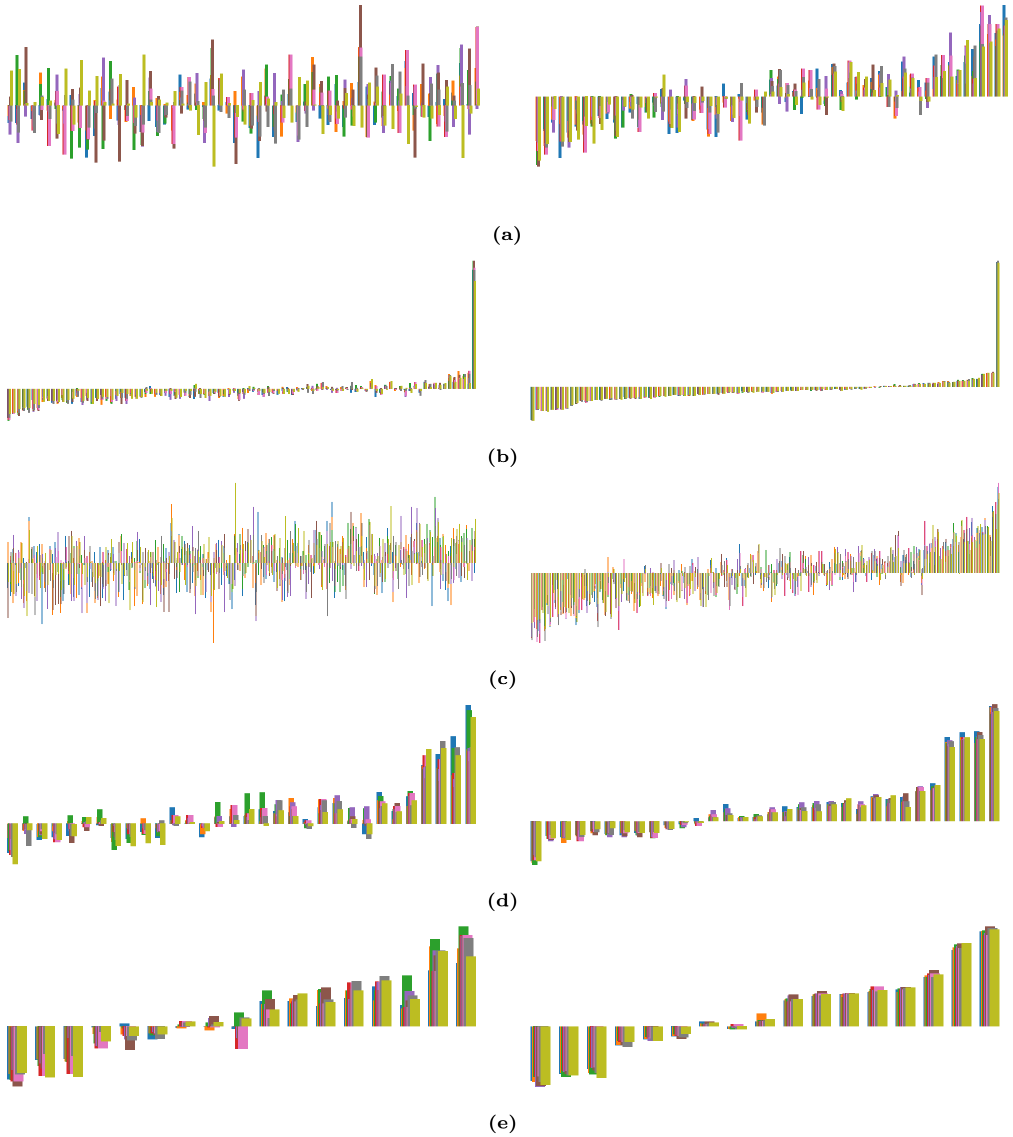


Figure B.1: We plot the average similarity across feature attributions for an individual point, averaged over 276 comparisons of feature attributions from two different models. This is aggregated across the entire validation split. The error bars represent the standard deviation over the 276 comparisons between models. Each row of plots constitutes the plots for a given dataset, noted on the far left, and each column of plots is for a given metric, noted at the top. Note that for image datasets, (FMNIST and Colon), we plot SSIM instead of Spearman's Ranking Coefficient (ρ). The x-axis is the number of models in the ensemble, starting with one, and the y-axis indicates the value of the similarity metric averaged over all 276 comparisons of individual points' in the validation split's attributions. The red and orange lines depict regular ensembles, and the green and blue represent selective ensembles.



(a)

(b)

(c)

(d)

(e)

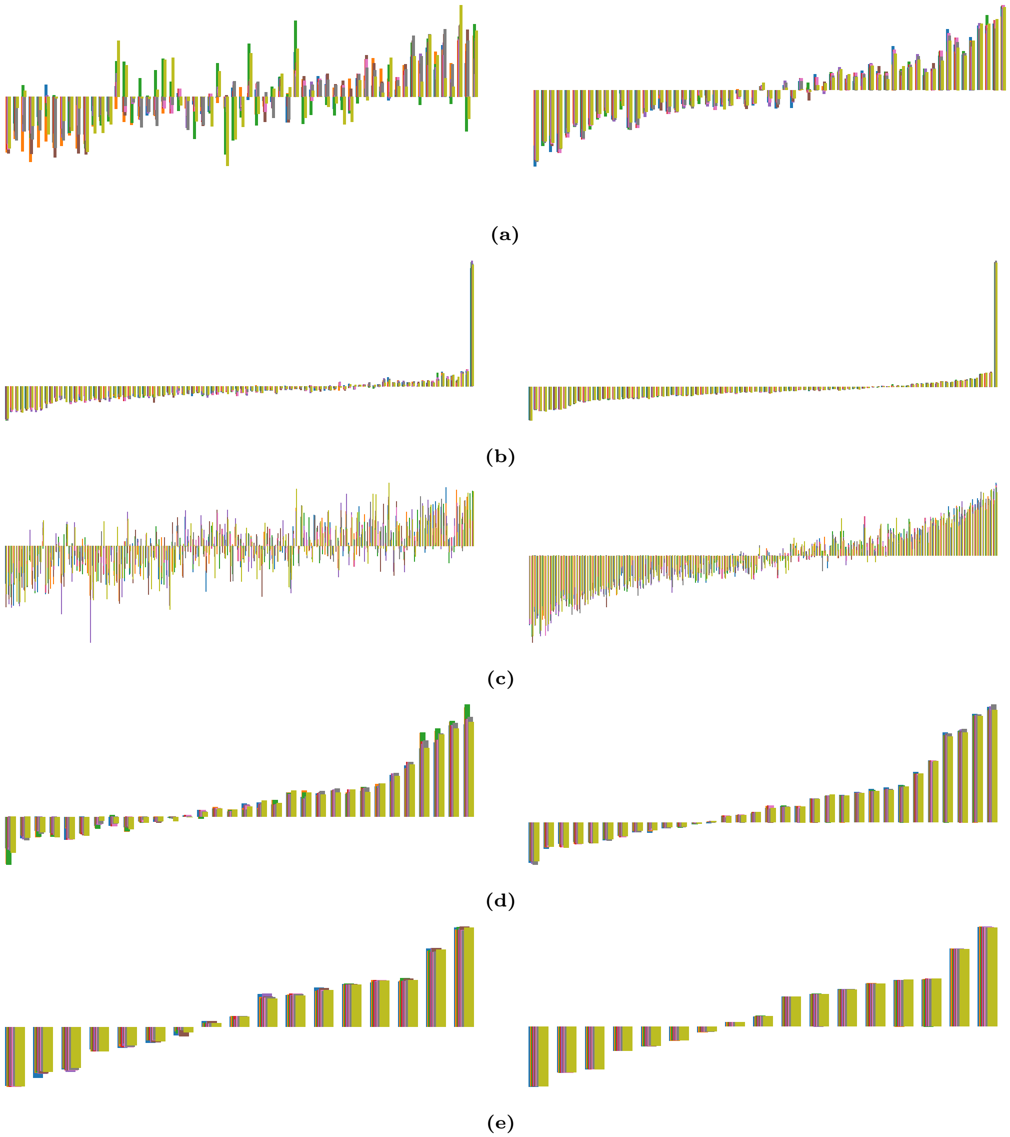
Figure B.2: Inconsistency of attributions on the same point across an individual (left) and ensembled (right) model ($n = 15$), for all datasets, over differences in random seed chosen for initialization parameters before training. The height of each bar on the horizontal axis represents the attribution score of a distinct feature, and each color represents a different model. Features are ordered according to the attribution scores of one randomly-selected model. Figure a depicts the German Credit Dataset, Figure b depicts Adult, Figure c Seizure, Figure d Taiwanese, and Figure e Warfarin. We do not include feature attribution for image datasets as the individual pixels are less meaningful than the feature attributions in a tabular dataset.

		disagreement of non-abstaining ensembles						
$\mathcal{S}$	n	Ger. Credit	Adult	Seizure	Warfarin	Tai. Credit	FMNIST	Colon
LOO	1	0.250	0.068	0.060	0.028	0.030	0.056	0.068
LOO	5	.180	.033	.028	.017	.030	.019	.022
LOO	10	.120	.023	.020	.013	.020	.025	.013
LOO	15	.065	.019	.016	.011	.013	.025	.010
LOO	20	.060	.016	.014	.010	.013	.027	.003
RS	1	0.585	0.089	0.061	0.0	0.080	0.115	0.066
RS	5	.290	.043	.028	.000	.040	.028	.0168
RS	10	.245	.030	.022	.000	.028	.029	.012
RS	15	.175	.025	.015	.000	.023	.020	.011
RS	20	.155	.021	.014	.000	.020	.018	.010

Table B.5: The percentage of points with disagreement between at least one pair of models ($p_{\text{flip}} > 0$) trained with different random seeds (RS) or leave-one-out differences in training data, for singleton models ($n = 1$) and non-selective ensembles ($n > 1$). Results presented over 10 re-samplings of different constituent models. While ensembling alone mitigates much of the prediction instability, it is unable to eliminate it as selective ensembles do.

		accuracy of non-abstaining ensembles						
$\mathcal{S}$	n	Ger. Credit	Adult	Seizure	Warfarin	Tai. Credit	FMNIST	Colon
LOO	5	.728	.844	.978	.685	.821	.918	.927
LOO	10	.728	.844	.978	.686	.821	.918	.927
LOO	15	.733	.844	.979	.685	.821	.917	.927
LOO	20	.730	.843	.979	.685	.821	.918	.927
RS	5	.745	.842	.975	.688	.822	.919	.927
RS	10	.746	.843	.975	.688	.822	.920	.928
RS	15	.750	.842	.975	.688	.822	.920	.928
RS	20	.747	.842	.975	.688	.822	.920	.938

Table B.6: The mean and standard deviation of the accuracy of non-selective ensembles for sizes n=5,10,15,20 over 10 re-samplings of different constituent models.



(a)

(b)

(c)

(d)

(e)

Figure B.3: Inconsistency of attributions on the same point across an individual (left) and ensembled (right) model ($n = 15$), for all datasets, over leave-one-out differences in the training set. The height of each bar on the horizontal axis represents the attribution score of a distinct feature, and each color represents a different model. Features are ordered according to the attribution scores of one randomly-selected model. Figure a depicts the German Credit Dataset, Figure b depicts Adult, Figure c Seizure, Figure d Taiwanese, and Figure e Warfarin. We do not include feature attribution for image datasets as the individual pixels are less meaningful than the feature attributions in a tabular dataset.

Appendix C

Consistent Counterfactuals Appendix

C.1 Proofs

C.1.1 Theorem 1 and Lemma 1

Theorem 1 *Suppose that H_1, H_2 are orthogonal decision boundaries in a piecewise-linear network $F(\mathbf{x}) = sign\{w_1^\top ReLU(W_0\mathbf{x})\}$, and let $\mathbf{x}$ be an arbitrary point in its domain. If the projections of $\mathbf{x}$ onto the corresponding halfspace constraints of H_1, H_2 are on H_1 and H_2, then there exists a point $\mathbf{x}'$ such that:*

$$1)\ d(\mathbf{x}', H_2) = 0 \qquad 2)\ d(\mathbf{x}', H_2) < d(\mathbf{x}, H_2) \qquad 3)\ d(\mathbf{x}, H_1) \leq d(\mathbf{x}', H_1)$$

where $d(\mathbf{x}, H_*)$ denotes the distance between $\mathbf{x}$ and the nearest point on a boundary H_*.

Proof. Let $u(\mathbf{x})_i = W_0\mathbf{x}$ be the pre-activation of the neuron i-th output in the hidden layer. The status of the neuron therefore will have the following two status: `ON` if $u(\mathbf{x})_i > 0$ and `OFF` otherwise. When a neuron is `ON`, the post-activation is identical to the pre-activation. Therefore, we can represent the ReLU function as a linear function of all neurons' activation status. Formally, the logit output of the network F can be written as

$$f(\mathbf{x}) = w_1^\top \Lambda W_0 \mathbf{x} \tag{C.1}$$

where Λ is a diagonal matrix $diag([\lambda_0, \lambda_1, ..., \lambda_n])$ such that $\lambda_i = \mathbb{I}(u(\mathbf{x})_i > 0)$. The network is a linear function within a neighborhood if all points in such a neighborhood have the same activation matrix Λ. For any two decision boundaries H_1 and H_2, the normal vectors of these decision boundaries can be written as $\mathbf{n}_1^\top = w_1^\top \Lambda_1 W_0$ and $\mathbf{n}_2^\top = w_1^\top \Lambda_2 W_0$, respectively, where Λ_1 and Λ_2 are determined by the activation status of internal neurons.

For an input $\mathbf{x}$, if the projections of $\mathbf{x}$ onto the corresponding halfspace constraints of H_1, H_2 are on H_1 and H_2, then the distance $d(\mathbf{x}, H_1)$ and $d(\mathbf{x}, H_2)$ are given by projections as follows:

$$d(\mathbf{x}, H_1) = \frac{|\mathbf{n}_1^\top \mathbf{x}|}{||\mathbf{n}_1||_2}, \quad d(\mathbf{x}, H_2) = \frac{|\mathbf{n}_2^\top \mathbf{x}|}{||\mathbf{n}_2||_2} \tag{C.2}$$

W.L.O.G. we assume $F(\mathbf{x}) = 1$ and $\mathbf{n}_1$ and $\mathbf{n}_2$ point towards $\mathbf{x}$. Let a point $\mathbf{y}$ defined as

$$\mathbf{y} = \mathbf{y}' - \frac{|\mathbf{n}_2^\top \mathbf{y}'|\mathbf{n}_2}{||\mathbf{n}_2||_2^2} \tag{C.3}$$

$$\mathbf{y}' = \mathbf{x} + \eta \frac{\mathbf{n}_1}{||\mathbf{n}_1||_2} \tag{C.4}$$

where η is tiny positive scalar such that $F(\mathbf{y}) = F(\mathbf{x}) = 1$. We firstly show that $d(\mathbf{y}, H_2) = 0$ as follows:

$$d(\mathbf{y}, H_2) = \frac{|\mathbf{n}_2^\top \mathbf{y}|}{||\mathbf{n}_2||_2} \tag{C.5}$$

$$= \frac{|\mathbf{n}_2^\top (\mathbf{y}' - \frac{|\mathbf{n}_2^\top \mathbf{y}'|\mathbf{n}_2}{||\mathbf{n}_2||_2^2})|}{||\mathbf{n}_2||_2} \tag{C.6}$$

$$= \frac{|\mathbf{n}_2^\top \mathbf{y}' - |\mathbf{n}_2^\top \mathbf{y}'||}{||\mathbf{n}_2||_2} \tag{C.7}$$

$$= \frac{|\mathbf{n}_2^\top \mathbf{y}' - \mathbf{n}_2^\top \mathbf{y}'|}{||\mathbf{n}_2||_2} \quad (\eta \text{ is tiny so } \mathbf{n}_2 \text{ points to } \mathbf{y}') \tag{C.8}$$

$$= 0 \tag{C.9}$$

We secondly show that $d(\mathbf{y}, H_1) > d(\mathbf{x}, H_1)$ as follows:

$$d(\mathbf{y}, H_1) = \frac{|\mathbf{n}_1^\top \mathbf{y}|}{||\mathbf{n}_1||_2} \tag{C.10}$$

$$= \frac{|\mathbf{n}_1^\top (\mathbf{y}' - \frac{|\mathbf{n}_2^\top \mathbf{y}'|\mathbf{n}_2}{||\mathbf{n}_2||_2^2})|}{||\mathbf{n}_1||_2} \tag{C.11}$$

$$= \frac{|\mathbf{n}_1^\top (\mathbf{x} + \eta \frac{\mathbf{n}_1}{||\mathbf{n}_1||_2} - \frac{|\mathbf{n}_2^\top (\mathbf{x} + \eta \frac{\mathbf{n}_1}{||\mathbf{n}_1||_2})|\mathbf{n}_2}{||\mathbf{n}_2||_2^2})|}{||\mathbf{n}_1||_2} \tag{C.12}$$

$$= \frac{|\mathbf{n}_1^\top \mathbf{x} + \eta||\mathbf{n}_1||_2 - \mathbf{n}_1^\top \mathbf{n}_2 \frac{|\mathbf{n}_2^\top (\mathbf{x} + \eta \frac{\mathbf{n}_1}{||\mathbf{n}_1||_2})|}{||\mathbf{n}_2||_2^2})|}{||\mathbf{n}_1||_2} \tag{C.13}$$

$$= \frac{|\mathbf{n}_1^\top \mathbf{x} + \eta||\mathbf{n}_1||_2|}{||\mathbf{n}_1||_2} \quad (H_1 \text{ and } H_2 \text{ are orthogonal}) \tag{C.14}$$

$$\geq \frac{|\mathbf{n}_1^\top \mathbf{x}|}{||\mathbf{n}_1||_2} = d(\mathbf{x}, H_1) \tag{C.15}$$

The proof of Theorem 1 is complete. □

Lemma 1 *Let H_1, H_2, F and $\mathbf{x}$ be defined as in Theorem 1. If the projections of $\mathbf{x}$ onto the corresponding halfspace constraints of H_1, H_2 are on H_1 and H_2, but there* does not *exist a point $\mathbf{x}'$ satisfying* (2) *and* (3) *from Theorem 1, then $H_1 = H_2$.*

Note if we remove the assumption that H_1 and H_2 are orthogonal, we will show that Theorem 1 will hold by condition. Let $m(\mathbf{x}) = |\mathbf{n}_1^\top \mathbf{x} + \eta||\mathbf{n}_1||_2 - \mathbf{n}_1^\top \mathbf{n}_2 \frac{|\mathbf{n}_2^\top(\mathbf{x}+\eta\frac{\mathbf{n}_1}{||\mathbf{n}_1||_2})|}{||\mathbf{n}_2||_2^2})|$. Assume the angle between the normal vectors of H_1 and H_2 is θ such that $\mathbf{n}_1^\top \mathbf{n}_2 = ||\mathbf{n}_1||_2||\mathbf{n}_2||_2 \cos\theta$.

$$\begin{aligned}
m(\mathbf{x}) &= |\mathbf{n}_1^\top \mathbf{x} + \eta||\mathbf{n}_1||_2 - \mathbf{n}_1^\top \mathbf{n}_2 \frac{|\mathbf{n}_2^\top(\mathbf{x}+\eta\frac{\mathbf{n}_1}{||\mathbf{n}_1||_2})|}{||\mathbf{n}_2||_2^2})| && \text{(C.16)}\\
&= |\mathbf{n}_1^\top \mathbf{x} + \eta(||\mathbf{n}_1||_2 - \frac{\mathbf{n}_1^\top \mathbf{n}_2 \cdot \mathbf{n}_2^\top \mathbf{n}_1}{||\mathbf{n}_2||_2^2||\mathbf{n}_1||_2}) - \frac{\mathbf{n}_1^\top \mathbf{n}_2 \cdot \mathbf{n}_2^\top \mathbf{x}}{||\mathbf{n}_2||_2^2}| && \text{(C.17)}\\
&= |\mathbf{n}_1^\top \mathbf{x} + \eta(1-\cos^2\theta)||\mathbf{n}_1||_2 - \mathbf{n}_1\mathbf{x}\cos\theta| && \text{(C.18)}
\end{aligned}$$

Since $d(\mathbf{y}, H_1) \propto m(\mathbf{x})$ and $d(\mathbf{x}, H_1) \propto |\mathbf{n}_1^\top \mathbf{x}|$ and they share they same denominator $||\mathbf{n}_1||_2$. In order to have $m(\mathbf{x}) > |\mathbf{n}_1^\top \mathbf{x}|$, we just need $\eta(1-\cos^2\theta)||\mathbf{n}_1||_2 - \mathbf{n}_1\mathbf{x}\cos\theta > 0$, which means we need to find a η such that $\eta(1-\cos^2\theta)||\mathbf{n}_1||_2 > \mathbf{n}_1\mathbf{x}\cos\theta$. Moving terms around we have the following inequality:

$$\eta > \frac{\mathbf{n}_1\mathbf{x}\cos\theta}{(1-\cos^2\theta)||\mathbf{n}_1||_2} = \frac{||\mathbf{x}||_2}{\frac{1}{\cos\theta} - \cos\theta} \tag{C.19}$$

The RHS goes to 0 when $\theta \to \frac{\pi}{2}$, which corresponds to the situation of Theorem 1. When $\theta \to 0$ ($H_1 = H_2$), RHS goes to ∞, which means we cannot find a point $\mathbf{y}$ satisfying the Theorem 1, which completes the proof of Lemma 1.

C.1.2 Theorem 2 and Proposition 1

Theorem 2 *Let $f(\mathbf{x}) \overset{\text{def}}{=} \mathbf{w}^\top \cdot h(\mathbf{x}) + b$ be a ReLU network with a single logit output (i.e., a binary classifier), where $h(\mathbf{x})$ is the output of the penultimate layer, and denote $\sigma_\mathbf{w} = \sigma(f(\mathbf{x}))$ as the sigmoid output of the model at $\mathbf{x}$. Let $\mathcal{W} \overset{\text{def}}{=} \{\mathbf{w}' : ||\mathbf{w} - \mathbf{w}'|| \leq \Delta\}$, $\chi_{\mathcal{D}_\mathbf{x}}(\mathbf{x})$ be the distributional influence of f when weights $\mathbf{w}$are used at the top layer, and $\chi'_{\mathcal{D}_\mathbf{x}}(\mathbf{x})$ be the distributional influence of f when weights $\mathbf{w}$' are used at the top layer. If h is K-Lipschitz in the support $S(\mathcal{D}_\mathbf{x})$, the following inequality holds:*

$$\forall \mathbf{w}' \in \mathcal{W}, \quad ||\chi_{\mathcal{D}_\mathbf{x}}(\mathbf{x}; \mathbf{w}) - \chi_{\mathcal{D}_\mathbf{x}}(\mathbf{x}; \mathbf{w}')|| \leq K\sqrt{[d\sigma(\mathbf{x}; \mathbf{w})||\mathbf{w}|| + C_1]^2 + C_2}$$

where C_1 and C_2 are constants and $d\sigma(\mathbf{x}; \mathbf{w}) \overset{\text{def}}{=} \partial\sigma_\mathbf{w}/\partial f$.

Proof. Consider a ReLU network as $g(h(\mathbf{x}))$. We first write out the expression of $h(x)$:

$$h(x) = \phi_{N-1}(W_{N-1}(\cdots \phi_1(W_1 x + b_1)) + b_{N-2}) \tag{C.20}$$

where W_i, b_i are the parameters for the i-th layer and $\phi_i(\cdot)$ is the corresponding ReLU activation. By the definition of the distributional influence,

$$\begin{aligned}
\chi_\mathcal{D}(\mathbf{x}) &= \mathbb{E}_{\mathbf{z}\sim\mathcal{D}(\mathbf{x})} \frac{\partial\sigma(g(h(\mathbf{z}); \mathbf{w}))}{\partial\mathbf{z}} && \text{(C.21)}\\
&= \mathbb{E}_{\mathbf{z}\sim\mathcal{D}(\mathbf{x})} \frac{\sigma(g)}{\partial g}\frac{\partial g(h; \mathbf{w})}{\partial h}\frac{\partial h(\mathbf{z})}{\partial\mathbf{z}} && \text{(C.22)}\\
&= \mathbb{E}_{\mathbf{z}\sim\mathcal{D}(\mathbf{x})}\left[\sigma(\mathbf{z}; \mathbf{w})(1-\sigma(\mathbf{z}; \mathbf{w}))\mathbf{w}\prod_{i=1}^{N-1}(W_i\Lambda_i(\mathbf{z}))^\top\right] && \text{(C.23)}\\
& && \text{(C.24)}
\end{aligned}$$

where W_i is the weight of the layer l_i if l_i is a dense layer or the equivalent weight matrix of a convolutional layer and $\Lambda_i(\mathbf{z})$ is an diagonal matrix with each diagonal entry being 1 if the neuron is activated or 0 other wise when evaluated at the point $\mathbf{z}$.

$$||\chi_{\mathcal{D}}(\mathbf{x}) - \chi'_{\mathcal{D}}(\mathbf{x})|| \tag{C.25}$$

$$= ||\mathbb{E}_{\mathbf{z}\sim\mathcal{D}(\mathbf{x})}\left[\sigma(\mathbf{z};\mathbf{w})(1-\sigma(\mathbf{z};\mathbf{w}))\mathbf{w}\prod_{i=1}^{N-1}(W_i\Lambda_i(\mathbf{z}))^\top\right] \tag{C.26}$$

$$- \mathbb{E}_{\mathbf{z}\sim\mathcal{D}(\mathbf{x})}\left[\sigma(\mathbf{z};\mathbf{w}')(1-\sigma(\mathbf{z};\mathbf{w}'))\mathbf{w}'\prod_{i=1}^{N-1}(W_i\Lambda_i(\mathbf{z}))^\top\right]|| \tag{C.27}$$

$$= ||\mathbb{E}_{\mathbf{z}\sim\mathcal{D}(\mathbf{x})}\left[(\sigma(\mathbf{z};\mathbf{w})(1-\sigma(\mathbf{z};\mathbf{w}))\mathbf{w} - \sigma(\mathbf{z};\mathbf{w}')(1-\sigma(\mathbf{z};\mathbf{w}'))\mathbf{w}')\prod_{i=1}^{N-1}(W_i\Lambda_i(\mathbf{z}))^\top\right]|| \tag{C.28}$$

$$\leq \mathbb{E}_{\mathbf{z}\sim\mathcal{D}(\mathbf{x})}\left[||(\sigma(\mathbf{z};\mathbf{w})(1-\sigma(\mathbf{z};\mathbf{w}))\mathbf{w} - \sigma(\mathbf{z};\mathbf{w}')(1-\sigma(\mathbf{z};\mathbf{w}'))\mathbf{w}')\prod_{i=1}^{N-1}(W_i\Lambda_i(\mathbf{z}))^\top||\right] \tag{C.29}$$

(Jensen's Inequality from (33) to (34))

$$\leq \mathbb{E}_{\mathbf{z}\sim\mathcal{D}(\mathbf{x})}\left[||(\sigma(\mathbf{z};\mathbf{w})(1-\sigma(\mathbf{z};\mathbf{w}))\mathbf{w} - \sigma(\mathbf{z};\mathbf{w}')(1-\sigma(\mathbf{z};\mathbf{w}'))\mathbf{w}')|| \cdot ||\prod_{i=1}^{N-1}(W_i\Lambda_i(\mathbf{z}))^\top||\right] \tag{C.30}$$

(By the definition of matrix operator norm from (34) to (35))

To simplify the expression, we denote

$$\mathbf{a} = \sigma(\mathbf{z};\mathbf{w})(1-\sigma(\mathbf{z};\mathbf{w}))\mathbf{w} - \sigma(\mathbf{z};\mathbf{w}')(1-\sigma(\mathbf{z};\mathbf{w}'))\mathbf{w}' \tag{C.31}$$

$$\mathbf{B} = \prod_{i=1}^{N-1}(W_i\Lambda_i(\mathbf{z}))^\top \tag{C.32}$$

and now Equation (35) now becomes

$$||\chi_{\mathcal{D}}(\mathbf{x}) - \chi'_{\mathcal{D}}(\mathbf{x})|| \leq \mathbb{E}_{\mathbf{z}\sim\mathcal{D}(\mathbf{x})}\left[||\mathbf{a}|| \cdot ||\mathbf{B}||\right] \tag{C.33}$$

with Cauchy-Shwartz inequality, we find that

$$\mathbb{E}_{\mathbf{z}\sim\mathcal{D}(\mathbf{x})}\left[||\mathbf{a}|| \cdot ||\mathbf{B}||\right] \leq \sqrt{\mathbb{E}_{\mathbf{z}\sim\mathcal{D}(\mathbf{x})}||\mathbf{a}||^2} \cdot \sqrt{\mathbb{E}_{\mathbf{z}\sim\mathcal{D}(\mathbf{x})}||\mathbf{B}||^2} \tag{C.34}$$

Now we will show that these two terms $\sqrt{\mathbb{E}_{\mathbf{z}\sim\mathcal{D}(\mathbf{x})}||\mathbf{a}||^2}$ and $\sqrt{\mathbb{E}_{\mathbf{z}\sim\mathcal{D}(\mathbf{x})}||\mathbf{B}||^2}$ are bounded.

(1) Bound for the term $\sqrt{\mathbb{E}_{\mathbf{z}\sim\mathcal{D}(\mathbf{x})}||\mathbf{a}||^2}$

Consider the relation between the expectation and the variance of random variables

$$\mathbb{E}\,X^2 = (\mathbb{E}\,X)^2 + Var(X) \tag{C.35}$$

which implies that

$$\mathbb{E}_{\mathbf{z}\sim\mathcal{D}(\mathbf{x})}||\mathbf{a}||^2 = (\mathbb{E}_{\mathbf{z}\sim\mathcal{D}(\mathbf{x})}||\mathbf{a}||)^2 + Var(||\mathbf{a}||) \tag{C.36}$$

We simplify the notation by defining

$$d\sigma(\mathbf{z};\mathbf{w}') \overset{\text{def}}{=} \sigma(\mathbf{z};\mathbf{w}')(1-\sigma(\mathbf{z};\mathbf{w}')$$

and we denote $d\sigma(\mathbf{z};\mathbf{w}) = d\sigma(\mathbf{x};\mathbf{w}) + \delta(\mathbf{z};\mathbf{w})$ and $d\sigma(\mathbf{z};\mathbf{w}') = d\sigma(\mathbf{x};\mathbf{w}') + \delta(\mathbf{z};\mathbf{w}')$. Note that $\delta \leq \frac{1}{4}$ because $d\sigma \in [0, \frac{1}{4}]$. Therefore, the $\mathbb{E}_{\mathbf{z}\sim\mathcal{D}(\mathbf{x})}||\mathbf{a}||$ can be simplified as

$$\begin{aligned}
\mathbb{E}_{\mathbf{z}\sim\mathcal{D}(\mathbf{x})}||\mathbf{a}|| &= \mathbb{E}_{\mathbf{z}\sim\mathcal{D}(\mathbf{x})}||d\sigma(\mathbf{x};\mathbf{w})\mathbf{w} - d\sigma(\mathbf{x};\mathbf{w}')\mathbf{w}' + \delta(\mathbf{z};\mathbf{w})\mathbf{w} - \delta(\mathbf{z};\mathbf{w}')\mathbf{w}'|| && \text{(C.37)}\\
&\leq ||d\sigma(\mathbf{x};\mathbf{w})\mathbf{w} - d\sigma(\mathbf{x};\mathbf{w}')\mathbf{w}'|| + \mathbb{E}_{\mathbf{z}\sim\mathcal{D}}||\delta(\mathbf{z};\mathbf{w})\mathbf{w} - \delta(\mathbf{z};\mathbf{w}')\mathbf{w}'|| \quad \text{(Triangle Inequality)} && \text{(C.38)}\\
&\leq ||d\sigma(\mathbf{x};\mathbf{w})\mathbf{w} - d\sigma(\mathbf{x};\mathbf{w}')\mathbf{w}'|| + \mathbb{E}_{\mathbf{z}\sim\mathcal{D}}||\delta(\mathbf{z};\mathbf{w})\mathbf{w}|| + \mathbb{E}_{\mathbf{z}\sim\mathcal{D}}||\delta(\mathbf{z};\mathbf{w}')\mathbf{w}'|| && \text{(C.39)}\\
&\leq ||d\sigma(\mathbf{x};\mathbf{w})\mathbf{w} - d\sigma(\mathbf{x};\mathbf{w}')\mathbf{w}'|| + \mathbb{E}_{\mathbf{z}\sim\mathcal{D}}|\delta(\mathbf{z};\mathbf{w})|||\mathbf{w}|| + \mathbb{E}_{\mathbf{z}\sim\mathcal{D}}|\delta(\mathbf{z};\mathbf{w}')|||\mathbf{w}'|| && \text{(C.40)}\\
&\leq ||d\sigma(\mathbf{x};\mathbf{w})\mathbf{w} - d\sigma(\mathbf{x};\mathbf{w}')\mathbf{w}'|| + \frac{1}{4}(||\mathbf{w}|| + ||\mathbf{w}'||) \qquad (\delta \leq \frac{1}{4}) && \text{(C.41)}\\
&\leq ||d\sigma(\mathbf{x};\mathbf{w})\mathbf{w} - d\sigma(\mathbf{x};\mathbf{w}')\mathbf{w}'|| + \frac{1}{2}(||\mathbf{w}|| + \frac{1}{2}\Delta) \qquad (||w-w'|| \leq \Delta) && \text{(C.42)}\\
&\leq ||d\sigma(\mathbf{x};\mathbf{w})\mathbf{w}|| + ||d\sigma(\mathbf{x};\mathbf{w}')\mathbf{w}'|| + \frac{1}{2}(||\mathbf{w}|| + \frac{1}{2}\Delta) \quad \text{(Triangle Inequality)} && \text{(C.43)}\\
&\leq d\sigma(\mathbf{x};\mathbf{w})||\mathbf{w}|| + \frac{1}{4}(||\mathbf{w}|| + \Delta) + \frac{1}{2}(||\mathbf{w}|| + \frac{1}{2}\Delta) \quad (d\sigma \leq \frac{1}{4}) && \text{(C.44)}\\
&\leq d\sigma(\mathbf{x};\mathbf{w})||\mathbf{w}|| + \frac{3}{4}||\mathbf{w}|| + \frac{1}{2}\Delta && \text{(C.45)}
\end{aligned}$$

To summarize, we find that

$$\mathbb{E}_{\mathbf{z}\sim\mathcal{D}(\mathbf{x})}||\mathbf{a}|| \leq d\sigma(\mathbf{x};\mathbf{w})||\mathbf{w}|| + \frac{3}{4}||\mathbf{w}|| + \frac{1}{2}\Delta \tag{C.46}$$

Since both sides of the inequalities are non-negative scalars, we see that

$$(\mathbb{E}_{\mathbf{z}\sim\mathcal{D}(\mathbf{x})}||\mathbf{a}||)^2 \leq \left[d\sigma(\mathbf{x};\mathbf{w})||\mathbf{w}|| + \frac{3}{4}||\mathbf{w}|| + \frac{1}{2}\Delta\right]^2 \tag{C.47}$$

The derivation of the variance term $Var(||\mathbf{a}||)$ may not have a simple analytical form to show that it is bounded, but it is easy to find an upper bound of $||\mathbf{a}||$

$$
\begin{aligned}
||\mathbf{a}|| &= ||d\sigma(\mathbf{x};\mathbf{w})\mathbf{w} - d\sigma(\mathbf{x};\mathbf{w}')\mathbf{w}' + \delta(\mathbf{z};\mathbf{w})\mathbf{w} - \delta(\mathbf{z};\mathbf{w}')\mathbf{w}'|| \quad \text{(from Equation (43))} && \text{(C.48)}\\
&\leq ||d\sigma(\mathbf{x};\mathbf{w})\mathbf{w}|| + ||d\sigma(\mathbf{x};\mathbf{w}')\mathbf{w}'|| + ||\delta(\mathbf{z};\mathbf{w})\mathbf{w}|| + ||\delta(\mathbf{z};\mathbf{w}')\mathbf{w}'|| \quad \text{(Triangle Inequality)} && \text{(C.49)}\\
&\leq \frac{1}{4}||\mathbf{w}|| + \frac{1}{4}(||\mathbf{w}|| + \Delta) + ||\mathbf{w}|| + (||\mathbf{w}|| + \Delta) \quad (d\sigma \leq \frac{1}{4}), (||w - w'|| \leq \Delta) && \text{(C.50)}\\
&\leq \frac{5}{2}||\mathbf{w}|| + \frac{5}{4}\Delta && \text{(C.51)}
\end{aligned}
$$

which implies that $||\mathbf{a}|| \in [0, \frac{5}{2}||\mathbf{w}|| + \frac{5}{4}\Delta]$. With Popoviciu's inequality, the variance $Var(||\mathbf{a}||)$ must be bounded such that

$$
Var(||\mathbf{a}||) \leq \frac{1}{4}\left[\frac{5}{2}||\mathbf{w}|| + \frac{5}{4}\Delta\right]^2 \tag{C.52}
$$

Now so far we have derived the upper-bounds for $(\mathbb{E}_{\mathbf{z}\sim\mathcal{D}(\mathbf{x})}||\mathbf{a}||)^2$ and $Var(||\mathbf{a}||)$; put together, we show that

$$
\sqrt{\mathbb{E}_{\mathbf{z}\sim\mathcal{D}(\mathbf{x})}||\mathbf{a}||^2} \leq \sqrt{\left[d\sigma(\mathbf{x};\mathbf{w})||\mathbf{w}|| + C_1\right]^2 + C_2} \tag{C.53}
$$

where

$$
\begin{aligned}
C_1 &= \frac{3}{4}||\mathbf{w}|| + \frac{1}{2}\Delta && \text{(C.54)}\\
C_2 &= \frac{1}{4}\left[\frac{5}{2}||\mathbf{w}|| + \frac{5}{4}\Delta\right]^2 && \text{(C.55)}
\end{aligned}
$$

(2) Bound for the term $\sqrt{\mathbb{E}_{\mathbf{z}\sim\mathcal{D}(\mathbf{x})}||\mathbf{B}||^2}$

$||\mathbf{B}||$ is the operator norm, namely the spectral norm for the matrix $\mathbf{B}$. As we assume $h(\mathbf{x})$ is K-Lipschitz, we know that

$$
\begin{aligned}
\mathbb{E}_{\mathbf{z}\sim\mathcal{D}(\mathbf{x})}||\mathbf{B}|| = \mathbb{E}_{\mathbf{z}\sim\mathcal{D}(\mathbf{x})}||\left[\prod_{i=1}^{N-1}(W_i\Lambda_i(\mathbf{z}))^\top\right]|| \leq \sup_{\mathbf{z}\sim\mathcal{D}(\mathbf{x})}||\left[\prod_{i=1}^{N-1}(W_i\Lambda_i(\mathbf{z}))^\top\right]|| = K && \text{(C.56)}\\
&& \text{(C.57)}
\end{aligned}
$$

and $\forall \mathbf{z} \sim \mathcal{D}(\mathbf{x})$

$$||\mathbf{B}|| \leq \sup_{\mathbf{z}\sim\mathcal{D}(\mathbf{x})} || \left[\prod_{i=1}^{N-1} (W_i \Lambda_i(\mathbf{z}))^\top\right] || = K \tag{C.58}$$

$$\tag{C.59}$$

Therefore,

$$||\mathbf{B}||^2 \leq K^2 \tag{C.60}$$

$$\mathbb{E}_{\mathbf{z}\sim\mathcal{D}(\mathbf{x})}||\mathbf{B}||^2 \leq K^2 \tag{C.61}$$

which implies $\sqrt{\mathbb{E}_{\mathbf{z}\sim\mathcal{D}(\mathbf{x})}||\mathbf{B}||^2} \leq K$. To put together, we finish the proof and show that

$$||\chi_{\mathcal{D}}(\mathbf{x}) - \chi'_{\mathcal{D}}(\mathbf{x})|| \leq \sqrt{\mathbb{E}_{\mathbf{z}\sim\mathcal{D}(\mathbf{x})}||\mathbf{a}||^2} \cdot \sqrt{\mathbb{E}_{\mathbf{z}\sim\mathcal{D}(\mathbf{x})}||\mathbf{B}||^2} \leq K\sqrt{[d\sigma(\mathbf{x};\mathbf{w})||\mathbf{w}|| + C_1]^2 + C_2} \tag{C.62}$$

□

C.1.3 Proposition 1

Proposition 1 *Let q be a differentiable, real-valued function in $\mathbb{R}^d$ and S be the support set of Uniform$(\mathbf{0} \rightarrow \mathbf{x})$. $\forall \mathbf{x}' \in S$,*

$$||\frac{\partial q(\mathbf{x}')}{\partial \mathbf{x}'}|| \geq ||\mathbf{x}||^{-1}|\frac{\partial q(r\mathbf{x}')}{\partial r}|_{r=1}|$$

Proof. First, we show that $\forall \mathbf{x}' \in S$

$$|\frac{\partial q(\mathbf{x}')}{\partial \mathbf{x}'}^\top \cdot \mathbf{x}'| \leq ||\frac{\partial q(\mathbf{x}')}{\partial \mathbf{x}'}|| \cdot ||\mathbf{x}'|| \quad \text{(Cauchy–Schwarz)} \tag{C.63}$$

By the construction of $\mathbf{x}'$ we know $||\mathbf{x}'|| \leq ||\mathbf{x}||$; therefore,

$$|\frac{\partial q(\mathbf{x}')}{\partial \mathbf{x}'}^\top \cdot \mathbf{x}'| \leq ||\frac{\partial q(\mathbf{x}')}{\partial \mathbf{x}'}|| \cdot ||\mathbf{x}|| \tag{C.64}$$

$$||\frac{\partial q(\mathbf{x}')}{\partial \mathbf{x}'}|| \geq ||\mathbf{x}||^{-1}|\frac{\partial q(\mathbf{x}')}{\partial \mathbf{x}'}^\top \cdot \mathbf{x}'| \tag{C.65}$$

Now consider a function $p(r;\mathbf{x}') = r\mathbf{x}'$. Then we show a trick of chain rule.

$$\frac{\partial q(p)}{\partial r} = \frac{\partial q(p)}{p}^\top \cdot \frac{\partial p(r;\mathbf{x}')}{\partial r} = \frac{\partial q(p)}{p}^\top \cdot \mathbf{x}' \tag{C.66}$$

Replacing the notation p with $\mathbf{x}'$ in $\frac{\partial q(\mathbf{x}')}{\partial \mathbf{x}'}^\top$ does not change the computation of taking the Jacobian of q's output with respect to the input; therefore, we show that

$$\frac{\partial q(\mathbf{x}')}{\partial \mathbf{x}'}^\top \cdot \mathbf{x}' = \frac{\partial q(p)}{p}^\top |_{r=1} \cdot \mathbf{x}' = \frac{\partial q(p)}{\partial r}|_{r=1} = \frac{\partial q(r\mathbf{x}')}{\partial r}|_{r=1} \tag{C.67}$$

We therefore complete the proof of Proposition 1 by showing

$$||\frac{\partial q(\mathbf{x}')}{\partial \mathbf{x}'}|| \geq ||\mathbf{x}||^{-1}|\frac{\partial q(r\mathbf{x}')}{\partial r}|_{r=1}| \tag{C.68}$$

□

C.2 Experiment Details

C.2.1 Meta information for Datasets

The German Credit [67] and Taiwanese Credit [67] data sets consist of individuals financial data, with a binary response indicating their creditworthiness. For the German Credit [67] dataset, there are 1000 points, and 20 attributes. We one-hot encode the data to get 61 features, and standardize the data to zero mean and unit variance using SKLearn Standard scaler. We partitioned the data intro a training set of 700 and a test set of 200. The Taiwanese Credit [67] dataset has 30,000 instances with 24 attributes. We one-hot encode the data to get 32 features and normalize the data to be between zero and one. We partitioned the data intro a training set of 22500 and a test set of 7500.

The HELOC dataset [85] contains anonymized information about the Home Equity Line of Credit applications by homeowners in the US, with a binary response indicating whether or not the applicant has even been more than 90 days delinquent for a payment. The dataset consists of 10459 rows and 23 features, some of which we one-hot encode to get a dataset of 10459 rows and 40 features. We normalize all features to be between zero and one, and create a train split of 7,844 and a validation split of 2,615.

The Seizure [67] dataset comprises time-series EEG recordings for 500 individuals, with a binary response indicating the occurrence of a seizure. This is represented as 11500 rows with 178 features each. We split this into 7,950 train points and 3,550 test points. We standardize the numeric features to zero mean and unit variance.

The CTG [67] dataset comprises of 2126 fetal cardiotocograms processed and labeled by expert obstetricians into three classes of fetuses, healthy, suspect, and pathological. We have turned this into a binary response between healthy and other classes. We split the data into 1,700 train points and a validation split of 425. There are 21 features for each instance, which we normalize to be between zero and one.

The Warfain dataset is collected by the International Warfarin Pharmacogenetics Consortium [47] about patients who were prescribed warfarin. We removed rows with missing values, 4819 patients remained in the dataset. The inputs to the model are demographic (age, height, weight, race), medical (use of amiodarone, use of enzyme inducer), and genetic (VKORC1, CYP2C9) attributes. Age, height, and weight are real-valued and were scaled to zero mean and unit variance. The medical attributes take binary values, and the remaining attributes were one-hot encoded. The output is the weekly dose of warfarin in milligrams, which we encode as "low", "medium", or "high", following [47].

The UCI datasets are under an MIT license, and Warfarin datasets are under a Creative Commons License. [47, 67]. The license for the FICO HELOC dataset is available at the dataset challenge website, and allows use for research purposes [86].

C.2.2 Hyper-parameters and Model Architectures

The German Credit and Seizure models have three hidden layers, of size 128, 64, and 16. Models on the Taiwanese dataset have two hidden layers of 32 and 16, and models on the HELOC dataset have two deep layers with sizes 100 and 32. The Warfarin models have one hidden layer of 100. The CTG models have three layers, of sizes 100, 32, and 16. German Credit, Adult, Seizure, Taiwanese, CTG and Warfarin models are trained for 100 epochs; HELOC models are trained for 50 epochs. German Credit models are trained with a batch size of 32; Adult, Seizure, and Warfarin models with batch sizes of 128; Taiwanese Credit models with batch sizes of 512, and CTG models with a batch size of 16. All models are trained with keras' Adam optimizer with the default parameters.

C.2.3 Implementation of Baseline Methods

We describe the parameters specific to each baseline method here. Common choices of hyper-parameters are shown in Table C.1.

Min-Cost ℓ_1/ℓ_2 [247] We implement this by setting $\beta = 1.0$ for ℓ_1 (or $\beta = 0.0$ for ℓ_2) and `confidence`=0.5 for the elastic-net loss [43] in ART [184].

Min-ϵ PGD [166]: For a given ϵ, we use 10 interpolations between 0 and the current ϵ as the norm bound in each PGD attack. The step size is set to $2 * \epsilon_c/$ `max_steps` where ϵ_c is the norm bound used. The maximum allowed norm bound is the median of the ℓ_2 norm of data points in the training set.

Pawelczyk et al. [200]: We train an AutoEncoder (AE) instead of a Variational AutoEncoder (VAE) to estimate the data manifold. Given that VAE jointly estimate the mean and the standard deviation of the latent distribution, it creates non-deterministic latent representation for the same input. In the contact with Pawelczyk et al., we are informed that we can only use the mean as the latent representation for an input; therefore, by taking out the standard deviation from a VAE, we instead train a AE that produces deterministic latent representation for each input. When searching for the latent representation of a counterfactual, we use random search as proposed by Pawelczyk et al. [200]: we randomly sample 1280 points around the latet representation of an input within a norm bound of 1.0 in the latent space. When generating random points, we use a fixed random seed 2021. If there are multiple counterfactuals, we return the one that is closest to the input. For all datasets, we use the following architecture for the hidden layers: 1024-128-32-128-1024.

Looveren et al. [243]: We use the public implementation of this method[1]. We use k-d trees with $k = 20$ to estimate the data manifold as the curre implementation only supports an AE where the input features must be between 0 and 1, while our dataset are not normalized into this range. The rest of the hyper-parameters are default values from the implementation: `theta`=100, `max_iterations`=100. This implementation

[1] `https://docs.seldon.io/projects/alibi/en/stable/methods/CFProto.html`

Hyper-parameters and Success Rate

Min ℓ_1	German Credit	Seizure	CTG	Warfarin	HELOC	Taiwanese Credit
ϵ	-	-	-	-	-	-
`step size`	0.05	0.05	0.05	0.5	0.01	0.05
`success rate`	0.35	0.14	1.00	1.00	1.00	1.00

Min ℓ_2	German Credit	Seizure	CTG	Warfarin	HELOC	Taiwanese Credit
ϵ	-	-	-	-	-	-
`step size`	0.01	0.01	0.01	0.01	0.01	0.01
`success rate`	0.84	0.71	1.00	1.00	1.00	1.00

Min ϵ PGD	German Credit	Seizure	CTG	Warfarin	HELOC	Taiwanese Credit
Max. ϵ	3.00	3.00	0.20	0.50	2.10	5.00
`step size`	`adp.`	`adp.`	`adp.`	`adp.`	`adp.`	`adp.`
`success rate`	0.90	0.86	0.51	0.85	1.00	1.00

Looveren et al.	German Credit	Seizure	CTG	Warfarin	HELOC	Taiwanese Credit
ϵ	-	-	-	-	-	-
`step size`	-	-	-	-	-	-
`success rate`	1.0	1.0	1.0	1.0	1.0	1.0

Pawelczyk et al.	German Credit	Seizure	CTG	Warfarin	HELOC	Taiwanese Credit
ϵ	0.3	1.0	1.0	1.0	1.0	1.0
`step size`	-	-	-	-	-	-
`success rate`	0.38	1.00	0.87	0.72	0.76	0.14

Table C.1: Hyper-parameters and Success Rate for each baseline methods. `adp.` denotes that the step size for each iteration is $2 * \epsilon/$`max_steps`.

only supports for non-eager mode so we turn off the eager execution in TF2 by running `tf.compat.v1.disable_eager_execution()` for this baseline.

SNS : We run SNS for 200 steps for all datasets and project the counterfactual back to a ℓ_2 ball. The size of the ball is set to be 0.8 multiplied by the largest size of the ball used for the baseline Min-ϵ PGD. For Max ℓ_1/ℓ_2 without a norm bound, we use the norm bound from Min-ϵ PGD. Similarly, the step size is set to $2 * 0.8 * \epsilon/200$.

C.2.4 Details of Retraining

We evaluate counterfactual invalidation over models with one-point differences in their training set, or different random initialization. For each dataset, we train a base model $F(\theta)$ with a specified random seed to determine initialization, and a specified train-validation split. We use this to generate all counterfactuals. We then train 100 models with one-point differences in the training set from a base model, as well as 100 models trained with different

Invalidation Rate (LOO)

Method	German Credit	Seizure	CTG	Warfarin	HELOC	Taiwanese Credit
Min. ℓ_1	0.41±0.04	-	0.07±0.09	0.44±0.02	0.30±0.03	0.30±0.03
+SNS	0.00±0.00	-	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00
Min. ℓ_2	0.36±0.05	0.64±0.06	0.48±0.17	0.35±0.02	0.55±0.05	0.27±0.05
+SNS	0.00±0.00	0.02±0.02	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00
Min. ϵ PGD	0.28±0.03	0.94±0.01	0.04±0.03	0.10±0.01	0.04±.0.01	0.04±0.00
+SNS	0.00±0.00	0.04±0.02	0.00±0.00	0.01±0.00	0.00±0.00	0.00±0.00
Looveren et al.	0.25±0.03	0.48±0.04	0.11±0.08	0.26±0.02	0.25±0.03	0.29±0.06
Pawelczyk et al.	0.20±0.13	0.16±0.14	0.00±0.00	0.02±0.00	0.05±0.06	0.02±0.01

Invalidation Rate (RS)

Min. ℓ_1	0.56±0.05	-	0.29±0.09	0.35±0.08	0.43±0.07	0.78±0.06
+SNS	0.07±0.02	-	0.01±0.00	0.00±0.00	0.00±0.00	0.04±0.02
Min. ℓ_2	0.56±0.06	0.77±0.12	0.49±0.15	0.30±0.05	0.61±0.07	0.72±0.07
+SNS	0.06±0.04	0.13±0.08	0.00±0.00	0.00±0.00	0.00±0.00	0.04±0.04
Min. ϵ PGD	0.61±0.04	0.94±0.12	0.09±0.04	0.12±0.03	0.11±0.02	0.24±0.07
+SNS	0.12±0.03	0.16±0.08	0.00±0.00	0.02±0.01	0.00±0.00	0.11±0.05
Looveren et al.	0.40±0.03	0.54±0.05	0.18±0.08	0.25±0.02	0.34±0.05	0.53±0.06
Pawelczyk et al.	0.35±0.16	0.11±0.17	0.06±0.04	0.01±0.00	0.15±0.21	0.20±0.09

Table C.2: Invalidation Rates with standard deviations for each datasets and each re-training situations. Results are aggregated over 100 models.

random initialization parameters. To do this, we randomly derive: a training set S, a set $O \subseteq S$ of size 100 that consists of points drawn randomly from test data (i.e. with which to create 100 different training sets with one point removed, $S^{(\setminus i)}$), and a test set. Then, For each $z_i \in O$, we train $F(\theta)'$ on $S^{(\setminus i)}$ by removing z_i from S. To train the 100 models with different initialization parameters, we simply change the numpy random seed directly before initializing a model.

C.2.5 Full results of IV with Standard Deviations

The full results of invalidation rates with standard deviations are shown in Table C.2.

C.2.6 ℓ_1 and ℓ_2 results

The full results of ℓ_1 and ℓ_2 costs with standard deviations are shown in Table C.3.

Counterfactual Cost (ℓ_2)

Method	German Credit	Seizure	CTG	Warfarin	HELOC	Taiwanese Credit
Min. ℓ_1	1.33±1.07	-	0.17±0.12	0.50±0.33	0.24±0.18	1.56±0.94
Min. ℓ_2	4.49±1.90	8.23±2.27	0.06±0.04	0.54±0.57	0.11±0.08	2.65±1.08
Looveren et al.	5.37±2.53	8.40±6.96	0.11±0.06	1.03±0.46	0.45±0.45	2.82±1.89
Min. ϵ PGD	1.02±0.57	1.36±0.38	0.08±0.03	0.31± 0.12	0.32±0.12	0.75±0.27
Min.ℓ_1 + SNS	3.40±0.82	-	0.25±0.08	0.80±0.29	1.71±0.12	3.50±0.91
Min.ℓ_2 + SNS	6.23±1.65	9.60±2.31	0.21±0.04	0.90±0.54	1.71±0.11	4.68±1.03
PGD + SNS	3.03±0.38	3.60±0.59	0.22±0.04	0.50±0.11	1.79±0.15	2.78±0.49
Pawelczyk et al.	7.15±2.12	13.66±7.46	1.07±0.20	2.62±0.79	1.35±0.93	4.24±2.23

Counterfactual Cost (ℓ_1)

Method	German Credit	Seizure	CTG	Warfarin	HELOC	Taiwanese Credit
Min. ℓ_1	2.09±2.00	-	0.16±0.19	0.48±0.59	0.35±0.30	2.40±2.84
Min. ℓ_2	24.70±13.14	77.89±28.38	0.18±0.12	1.17±0.91	0.43±0.31	9.76±3.93
Looveren et al.	15.93±8.93	76.99±69.93	0.16±0.12	1.75±0.99	0.97±1.05	6.11±5.12
Min. ϵ PGD	6.31±3.56	14.55±4.15	0.30±0.12	1.07±0.42	1.45±0.54	3.19±1.09
Min.ℓ_1 + SNS	13.81±2.96	-	0.58±0.14	1.61±0.52	7.11±0.72	11.04±3.17
Min.ℓ_2 + SNS	34.38±11.54	96.08±27.80	0.57±0.10	2.17±0.95	7.18±0.73	16.63±3.76
PGD + SNS	14.21±2.31	38.55±6.36	0.63±0.14	1.41±0.36	7.64±0.88	10.19±2.54
Pawelczyk et al.	38.48±12.31	145.36±77.67	3.03±0.63	±6.48±2.66	4.51±3.52	12.22±7.49

Table C.3: ℓ_1 and ℓ_2 costs of counterfactuals with standard deviations.

Appendix D

IRS Case Study Appendix

D.1 Fairness constraints and Monotonicity

In this section, we show that a selection process which achieves either equal true positive rates or equalized odds will, under certain (differing) conditions, satisfy monotonicity with respect to the ranking of bins by true misreport rate. That is, such models must choose a higher audit rate in a group with a higher rate of misreport than it chooses in a group with a lower rate of misreport. Given that, in our setting, misreport rate appears to be monotonic with respect to income, such results would imply audit rate monotonicity with respect to income as well.

For this section, we assume the following setup. There are two groups of observations G_1 and G_2 of equal size n, and they have m_1 and m_2 positive labels respectively and $r_1 = n - m_1$ and $r_2 = n - m_2$ negative labels. An auditor selects A_1 observations for audit from G_1 and A_2 from G_2 such that the total audits $A_1 + A_2$ is their audit budget A. The auditor has access to a model $\mathcal{M}$ which gives binary predictions $\hat{y} \in \{0, 1\}$. The auditor would like to select A_1 and A_2 in such a way that she maximizes true positives selected; we assume that $A << \sum_{j \in \{1,2\}} \sum_{i \in G_j} \mathcal{M}(X_i)$ - that is, the audit budget is much smaller than the total amount of positive predictions by the model.

After the auditor makes selections A_1 and A_2, we define the α_1 as the false positive rate of the audits for G_1; that is,

$$\alpha_1 = \text{FPR}_1 = \frac{\text{False Positives in } G_1 \text{ selected}}{r_1}.$$

In other words, α_1 is the false positive rate of the *composition* of whatever the auditor's selection process is with the predictions of the model (not the false positive rate of the model itself). We define α_2 similarly. Additionally, we define β_1 as the true positive rate of the audits for G_1, i.e.:

$$\beta_1 = \text{TPR}_1 = \frac{\text{True Positives in } G_1 \text{ selected}}{m_1}$$

and β_2 similarly. Finally, let $p_i = \frac{\text{True Positive Predictions for group } i}{A_i}$, often known as precision.

D.1.1 Equal TPR and Monotonicity

Our first lemma relates monotonicity to precision in the case of a selection process satisfying equal true positive rates:

Lemma 2. *Suppose that the selection process satisfies equal true positive rates. Then with* A_i*,* m_i*, and* p_i *defined as above:* $A_2 \geq A_1 \iff \frac{m_1}{m_2} \leq \frac{p_1}{p_2}$*.*

Proof. Note that:

$$p_i = \frac{\text{True Positive Predictions}}{\text{All Positive Predictions}} \implies \text{True Positives}_i = A_i p_i.$$

Then the true positive rate can be written as

$$\beta_i = \frac{\text{True Positive}_i}{\text{Positives}_i} = \frac{A_i p_i}{m_i}.$$

But by assumption, $\beta_1 = \beta_2 = \beta$, so

$$\frac{A_1 p_1}{m_1} = \frac{A_2 p_2}{m_2}.$$

But this implies that

$$\frac{A_1}{A_2} = \frac{m_1}{m_2}\frac{p_2}{p_1}.$$

Hence, $A_2 \geq A_1$ if and only if $\frac{m_1}{m_2}\frac{p_2}{p_1} \leq 1$, or in other words:

$$A_2 \geq A_1 \iff \frac{m_1}{m_2} \leq \frac{p_1}{p_2}.$$

□

To interpret this lemma, suppose that Group 2 has a higher misreport rate than Group 1 by some factor. Then the lemma states that for any selection process satisfying equal true positive rates, monotonicity with respect to misreport rate requires precision in Group 2 greater than in Group 1 by at least the same factor, and vice versa.

D.1.2 Equalized Odds and Monotonicity

The following lemma shows that, in this setting, any allocation that satisfies equalized odds (i.e. $\alpha_1 = \alpha_2 = \alpha$ and $\beta_1 = \beta_2 = \beta$) must audit the group with a *higher* misreport rate at a *higher* rate if the true positive rate is *larger* than the false positive rate; conversely, it must audit the group with a *higher* misreport rate at a *lower* rate if the true positive rate is *lower* than the false positive rate.

Lemma 3. *Suppose that the allocation* A_1, A_2 *satisfies equalized odds. That is,* $\alpha_1 = \alpha_2 = \alpha$ *and* $\beta_1 = \beta_2 = \beta$*. If* $\beta \geq \alpha$*, then* $A_2 \geq A_1 \iff m_2 \geq m_1$*; otherwise,* $A_2 \geq A_1 \iff m_1 \geq m_2$*.*

Proof. Note that A_1 is the sum of true and false positives in G_1 and A_2 is the sum of true and false positives in G_2. Since

$$\alpha = \alpha_1 = \frac{\mathrm{FP}_1}{r_1} \qquad \text{and} \qquad \beta = \beta_1 = \frac{\mathrm{TP}_1}{m_1},$$

we can observe that:

$$A_1 = r_1\alpha + m_1\beta$$

and similarly for A_2. But then:

$$\begin{aligned} A_2 - A_1 &= r_2\alpha + m_1\beta - (r_1\alpha + m_1\beta) \\ &= \alpha(r_2 - r_1) + \beta(m_2 - m_1) \\ &= \alpha((n - m_2) - (n - m_1)) + \beta(m_2 - m_1) \\ &= \alpha(m_1 - m_2) + \beta(m_2 - m_1) \\ &= (\beta - \alpha)(m_2 - m_1). \end{aligned}$$

But then we have that:

$$A_2 - A_1 > 0 \iff (\beta - \alpha)(m_2 - m_1) > 0,$$

yielding the claimed result. □

Lemma 3 shows that if the selection process as a whole satisfies equalized odds, then groups with higher misreport rates will be audited at a higher rate if and only if the process catches a larger fraction of misreporters than the fraction of non-misreporters it ensnares. In balanced settings and with good models, we might expect that generally the true positive rate will be higher than the false positive rate, and this is what provides intuition that imposing equalized odds might push the process towards monotonicity in misreport rate. But these rates interact with the overall audit budget: in the regime where the budget is very small and models are good, then it may be possible to obtain a low false positive rate but an *even lower* true positive rate. In that case, equalized odds will require that the group with higher non-compliance is audited *less.*

D.2 Further Experimental Details

In this paper, we compare LDA, Random Forest Classifier, Random Forest Regressor, Gradient Boost Classifier, and Gradient Boost Regressor models. We use the *sklearn* python package [202] to implement all models except for gradient boosted models, and search for optimal hyperparameters using *sklearn*'s *GridSearchCV* method with 5-fold cross validation. Gradient boosted models are created through the XGBoost python package, and optimal hyperparameters are also found using GridSearchCV. We use NRP data from 2010-2014 to train all models in this paper, with dollar values scaled to 2014 values. Our threshold for determining what qualifies as a tax misreport is a $200 difference between paid tax and amount owed. We winsorize amount of misreport to the 1st and 99th percentiles. We split the data into train, test, and validation sets randomly. Our train and validation sets comprise 75% of the data, with a test set of 25% of the data.

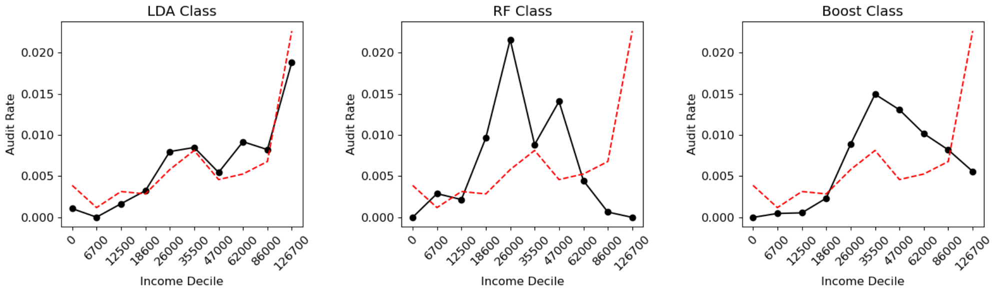


Figure D.1: Audit rate over income deciles, for LDA, Random Forest, and XGBoost classifiers trained with unweighted datasets of size 100k, subsampled from the weighted NRP data. (These allocations are in black, with oracle in red).

Model Type	Label Type	Subsampled (Data Size)	Revenue ($B)	No-Change Rate	Cost ($B)	Net Revenue ($B)	Oracle Overlap
Oracle	-	×	29.40	0.0%	0.33	29.07	1.00
LDA	Class	✓11M	6.07	12.8%	0.21	5.86	0.09
LDA	Class	✓1100k	6.61	16.0%	0.30	6.31	0.09
Random Forest	Class	×	3.05	3.5%	0.08	2.97	0.00
Random Forest	Class	✓1100k	3.19	4.5%	0.07	3.12	0.01
Grad Boost	Class	×	4.05	4.2%	0.08	3.97	0.00
Grad Boost	Class	✓1100k	3.72	4.7%	0.09	3.61	0.00

Table D.1: Revenue, No-change rate, cost, and net revenue for models trained on a subsampled dataset of size 100k. No-change rate represents the percentage of audits that were allocated to compliant tax-payers; cost reflects cost to the IRS as described in Section 5.8. These results reflect audit allocations which select the top 0.644% of taxpayers predicted most likely to misreport from each model. All metrics are reported on the test set, weighted using the sampling weights provided by the IRS to scale up to a representative sample of the US population.

We note that the IRS NRP data contains sampling weights, which are used to ensure that the NRP data is representative of the true underlying distribution of taxpayers [124]. We train all unconstrained models with sampling weights included in the NRP data using *sklearn*'s built in data-weighting feature, except LDA, whose *sklearn* implementation does not does not support training weights. For LDA, we create a representative dataset from the NRP data by randomly subsampling rows from the weighted training data according to the weights. For example, consider that each row x has a weight w, and the sum of all weights in the training set is W. Then each observation has probability $\frac{w}{W}$ of getting selected as any given row in the subsampled data. This produces an unweighted training set reflecting the same proportions as the weighted training data, with one million samples. As mentioned in Section 5.5, the *FairLearn* package [22] requires the use of the *sklearn* training weights feature to implement its in-process fairness enforcement algorithms. As a result, we also use the subsampling technique to create training sets for in-process fairness models, but with samples of $100,000$ points, as the algorithm is extremely time-intensive on large datasets (over 48 hours for one model). In order to show that the use of sampling weights during training, or the difference in training set size from 100k to 1M, does not strongly affect the results presented in the paper, we show the audit allocations and revenue, cost, and no-change rates of the LDA, Random Forest, and XGBoost classifiers in Figure D.1 and Table D.1 respectively.

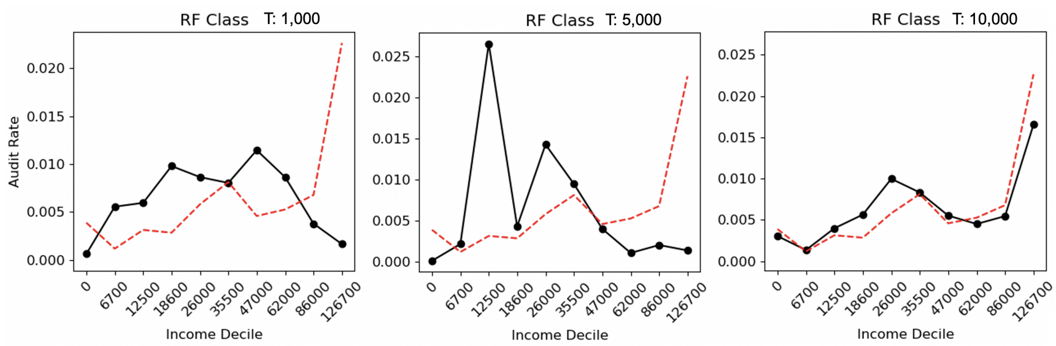


Figure D.2: Audit rate over income deciles, for random forest classification models trained with different thresholds for what consitutes a significant amount of misreport. From left to right, we have the allocation for a model trained with a threshold of $1,000, $5,000, and $10,000. (These allocations are in black, with oracle in red).

All analyses sections are produced on the test set. Cost and revenue calculations are reported by rescaling costs and revenues to reflect estimated annual values for the full population, for each year 2010-2014, and then dividing by five.

We sort taxpayers by descending order of predicted *misreport probability* from all classification models (using *sklearn*'s *predict_proba()*) method, in order to produce a ranking. We use *sklearn*'s *predict* method to return expected misreport for regression models. We use an audit rate budget of 0.644% of the taxpayer population, reflecting the average audit rate from 2010-2014, and select audits a_i by taking the top 0.644% of the taxpayer population in rank order. This 0.644% corresponds to weighted percentage of the population, computed with sampling weights, i.e. $\frac{\sum a_i w_i}{\sum w_i}$ where i is an observation in the weighted dataset, a_i is an indicator of whether to audit that observation, and w_i is the number of people the observation represents to create a representative population from the sampling data. The audit budget of 0.644% of the taxpayer population, is equivalent to 1125000 audits.

D.3 Robustness Checks on Classification Thresholds

In this section, we compare the audit allocations of high-flexibility classification models (namely, random forest classifiers) with different thresholds for what constitutes a significant adjustment. In the main text, we use a threshold of $200 to signify a significant misreport. In these experiments, we consider thresholds of $1,000, $5,000, and $10,000. Experimental setup is identical to that described in Section D.2, with the exception of the change in threshold. We display our results in Figure D.2, and Table D.2.

The results show us that changing the threshold of a significant adjustment to $1,000 does not significantly impact audit allocation compared to the results presented in the main text. A threshold of $5,000 exacerbates the classification model's excess focus on the lower end of the income spectrum, even beyond results shown in the main paper. Only a threshold of $10,000 makes a significant difference in terms of the audit allocation—shifting the focus to high income individuals almost exclusively— however, it results in an extremely high no-change rate.

Model Type	Label		Revenue	No-Change	Cost	Net Revenue
	Type	Threshold	($B)	Rate	($B)	($B)
Oracle	-	×	29.40	0.0%	0.33	29.07
LDA	Class	200	6.07	12.8%	0.21	5.86
Random Forest	Class	200	3.05	3.5%	0.08	2.97
Random Forest	Class	1,000	4.92	5.6%	0.10	2.87
Random Forest	Class	5,000	6.48	43.6%	0.15	6.35
Random Forest	Class	10,000	10.1	64.1%	.45	10.55
LDA	Class	1,000	6.3	17.4%	0.20	6.1
LDA	Class	5,000	7.52	53.3%	0.30	7.22
LDA	Class	10,000	9.0	70.8%	.47	8.53

Table D.2: Revenue, No-change rate, cost, and net revenue for models with different thresholds for what constitutes a significant misreport. No-change rate represents the percentage of audits that were allocated to compliant tax-payers; cost reflects cost to the IRS as described in Section 5.8. These results reflect audit allocations which select the top 0.644% of taxpayers (i.e. top 1125000 taxpayers) predicted most likely to misreport from each model. All metrics are reported on the test set, weighted using the sampling weights provided by the IRS to scale up to a representative sample of the US population.

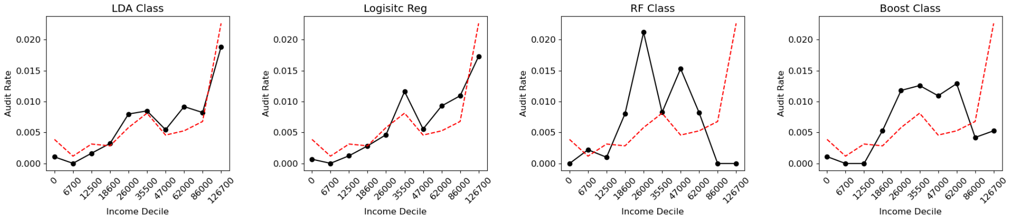


Figure D.3: Audit rate over income deciles, for LDA, Logisitc Regression, Random Forest, and XGBoost classifiers trained on NRP data. The new figure included in this graph, relative to the figures in the main paper, is the introduction of the logistic regression model. (These allocations are in black, with oracle in red).

D.4 Increased Audit Focus on Lower-and-Middle Income only in High Complexity Models

In this section, we provide results from a logistic regression model to further buttress the claim that only higher-complexity classification models result in audit allocations which exacerbate focus on lower and middle-income taxpayers. We train the Logistic Regression classification model with the same procedure outlined in Appendix D.2, with sampling weights directly included during training. The audit allocation is depicted in Figure D.3: the allocation is more monotonic than the higher complexity classification models; and is apparent in Table D.3, the no-change rate is higher, but the revenue is higher as well.

D.5 Additional Robustness Checks

As noted in the main text, we make several important choices. First, we focus on total positive income (TPI), rather than adjusted gross income (AGI; roughly corresponding to the taxpayer's total net income) because it it represents a simple measure of earnings that is less likely to be affected by audit determinations. Second, for our analysis of the status quo, we do not differentiate between EITC-specific audits for EITC claimants (e.g. qualifying

Model Type	Label Type	Subsampled (Data Size)	Revenue ($B)	No-Change Rate	Cost ($B)	Net Revenue ($B)	Oracle Overlap
Oracle	-	×	29.40	0.0%	0.33	29.07	1.00
LDA	Class	✓11M	6.07	12.8%	0.21	5.86	0.09
Random Forest	Class	×	3.05	3.5%	0.08	2.97	0.00
Grad Boost	Class	×	4.05	4.2%	0.08	3.97	0.00
Log. Reg.	Class	×	5.42	15.3%	0.19	5.23	0.06

Table D.3: Revenue, No-change rate, cost, and net revenue for models presented in the paper alongside results for a logistic regression model. No-change rate represents the percentage of audits that were allocated to compliant tax-payers; cost reflects cost to the IRS as described in Section 5.8. These results reflect audit allocations which select the top 0.644% of taxpayers predicted most likely to misreport from each model. All metrics are reported on the test set, weighted using the sampling weights provided by the IRS to scale up to a representative sample of the US population.

child eligibility) and income-centered audits (e.g. confirmation of reported small business or self-employment income). As we note above, this distinction is not relevant for the purposes of an ultimate determination as to a liability to the government, but for operational purposes, it may be meaningful to understand which type of audit is driving the vertical equity findings. Third, we focus on reported income figures rather than audit-adjusted figures. This is because, by definition, audit-adjusted income is not available to the IRS before auditing, so any policy or choice that relies on access to audit-adjusted income is unimplementable. However, audit-adjusted income may provide a better picture of distributional effects (at least for audited taxpayers).

D.5.1 Status Quo

In this section, we consider how the alternative choices (using AGI, splitting up EITC and income audits, and measuring model outcomes with respect to audit-adjusted income) in turn affect our status quo findings. We interpret these results as primarily confirming our main results.

Adjusted Gross Income First, we consider whether our motivating stylized facts — that low-income taxpayers are audited at rates about as high as very-high income taxpayers despite change rate being monotonic in income and average adjustment being much higher for high income taxpayers — is dependent on the choice of TPI rather than AGI. We thus recreate the left-most and right-most panels of Figure 5.1 with AGI as our feature in the x-axis. We use NRP data, which is selected via stratified random sampling, as before to avoid selection bias.

The left panel of Figure D.4 shows the 2014 audit rate for taxpayers in each $10,000-wide bin of AGI. The figure shows that the large spike near 0 observed with respect to TPI remains for AGI as well. However, the graph looks different in that AGI, unlike TPI, can be negative; the negative-AGI portion of the graph qualitatively resembles a (much noisier) mirror image of the non-negative-AGI porion, though negative-AGI taxpayers made up just over 1% of all taxpayers according to NRP data.

The right panel of Figure D.4 depicts change rate and average adjustment across AGI bins. Here, the bins consist of AGI deciles for non-negative AGI taxpayers augmented by a single bin for all negative-AGI taxpayers. Excluding the negative-AGI bin, the change rate and average adjustment follow a qualitatively similar trend to their counterparts observed on

TPI. That is, the change rate increases nearly monotonically, while the average adjustment is increasing overall but has a decreasing or flat portion. However, the overall difference between the average adjustment in the highest AGI bin and highest average adjustment among the lower-AGI bins is smaller than for TPI. As for the negative AGI bin, it has a relatively low (compared to other bins) change rate, but a higher average adjustment than any positive-AGI bin. Recall that AGI is income less various adjustments (e.g. for student loan interest, alimony payments, health insurance for self-employed taxpayers, etc.). As mentioned, given additional scope relative to TPI for errors, subjective determinations, or manipulation to influence ultimate AGI figures, we focus on TPI as our primary measure of income.

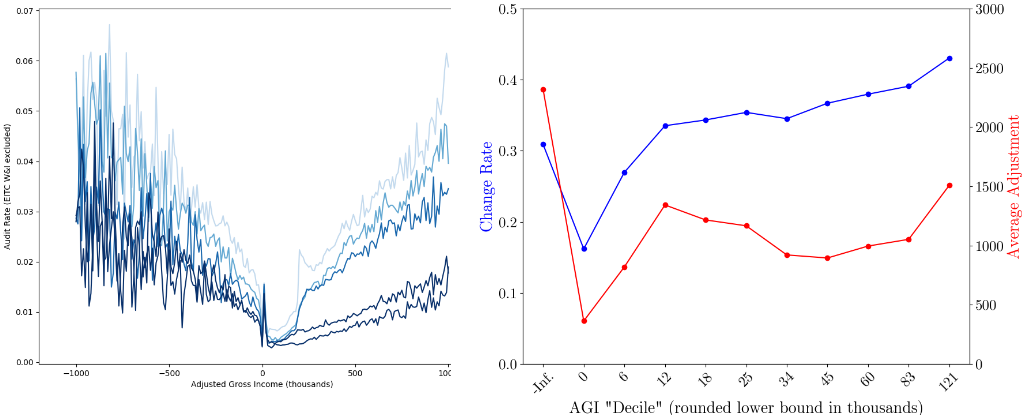


Figure D.4: Robustness checks with adjusted gross income. Left: The figure shows the audit rate by year at a given amount of adjusted gross income (discretized into bins of $10,000. Note that AGI may be negative; however, just over 1% of NRP observations submit negative AGI, so the noise in the left half of the graph is due to small sample size. Right: The figure shows outcomes in terms of misreport rate and average adjustment by AGI "deciles" (we compute deciles for observations non-negative AGI and add all negative AGI observations as an additional initial bin).

Income vs. EITC Audits Next, we explore whether the extent to which the observed non-monotonicity in audit rates by income is driven primarily by income-related audits (e.g. verifying that claimed income was truly received, that reported income presents a full picture of true income, etc.) or eligibility-related audits (e.g., whether a claimed dependent satisfies residency or relationship tests for EITC eligibility). To do this, we replicate our main audit-rate analysis after removing dependent-related audits. We do this using *project codes*. Projects codes are given to returns upon audit and correspond to a focus on particular issues. These do not necessarily map one-to-one with the income/EITC distinction — for example, some project codes correspond to a particular flag being triggered, and can result in focus on both eligibility and/or income issues depending on the return; still, careful examination of the issues considered allow us to develop an approximate measure of the intent of the audit.[1]

[1] We started with a list of project codes, project titles, and project descriptions. We examined all projects with EITC-related words in the title (e.g. "EITC" or "EIC"), as well as all projects indicated to be related to EITC by 4.19.14.4 in the Internal Revenue Manual.

We categorize EITC-related projects into three categories: most narrowly, *EITC-eligibility projects*, which only consider questions related to whether a taxpayer's EITC claim satisfies eligibility requirements; more generally, *EITC-Only* projects, which may consider more than eligibility but are still related to the EITC claim (e.g. verifiability of Schedule C income for EITC claimants); and most broadly, *EITC-mentioning* projects, which constitute any project which mentions EITC as the population of interest. So, for instance, audits about the premium tax credit within EITC claimants would be considered as part of the *EITC-mentioning* projects but not the *EITC-Only* or *EITC-eligibility* projects. Note that these categories are nested, so if we move from *excluding* only the first to the next to the last we end up with a successively narrower set of included audits. In particular, the set of audits that fall into *EITC-eligibility* projects but not *EITC-Only* projects are those which correspond strictly to eligibility questions, and so the effect of removing them shows (a lower bound on) the portion of audits which are due to eligibility and not income. (It is a *lower bound* because some projects in the *EITC-Only* do not only focus on income, but may also focus on eligibility; without further detail unavailable in our data, we cannot further distinguish between specific issues considered for each return within the same project code.)

Figure D.5 shows the results of this analysis for the tax year 2014. The figure depicts audit rate by TPI, but with several different lines indicating different levels of exclusions that have been made when calculating the audit rate. The shading increases with the breadth of exclusions (no exclusions, corresponding to our results in Figure 5.1, are plotted in lightest red, while the broadest exclusions, of all projects with any mention of EITC at all, are plotted in darkest red). Notice that the lightest color shows the ‘spike’ in audit rates for low income taxpayers, as displayed before, and excluding successively more returns unsurprisingly diminishes the calculated audit rate, until we are left with very few audits that are entirely unrelated to EITC claims for near-zero TPI taxpayers. Most interestingly, moving from no exclusions to excluding EITC-eligibility-specific projects decreases the audit rate at the spike from about 1.2% to about .7%. This indicates that, as a lower bound, about half of the spike is explained by EITC-eligibility-related projects.

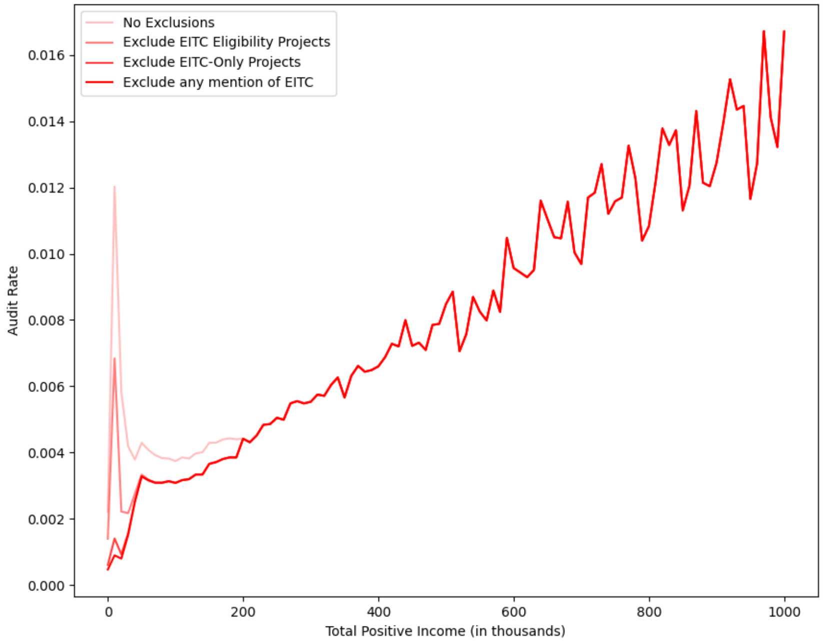


Figure D.5: Audit rate by TPI for tax year 2014 after excluding EITC-related projects of varying stringency of definition. The shades of lines move from light to dark mirroring how the consider exclusions move from very little to very broad. In particular, the lightest shade shows audit rate before any exclusions, the next shows audit rate after excluding projects related specifically to EITC eligibility, the next after excluding all projects related *only* to EITC, and the darkest after excluding all projects which mention EITC even if focused on unrelated issues.

More coarsely, we can simply look at to what extent the spike is being driven by EITC claimants at all, as indicated by claimants' *activity codes*. Activity code 270 correspond to EITC claimants with less than $25,000 of Schedule C (non-wage) income (e.g. income from self-employment), while activity code 271 captures the remainder. (Recall that income for the purposes of the EITC is not TPI, but AGI, as described above. So it is possible, though rare, for a taxpayer with high TPI to nonetheless be eligible for the EITC.) Figure D.6 displays the results of a similar exercise, moving from excluding 270 to excluding 270 and 271. The fact that the spike is essentially eliminated moving from no exclusions to excluding 270 suggests that non-monotonicity is driven by EITC claimants. (Note that this is not inconsistent with Figure D.5 because EITC claimants in 270 may be audited for non-eligibility matters, like income verification.)

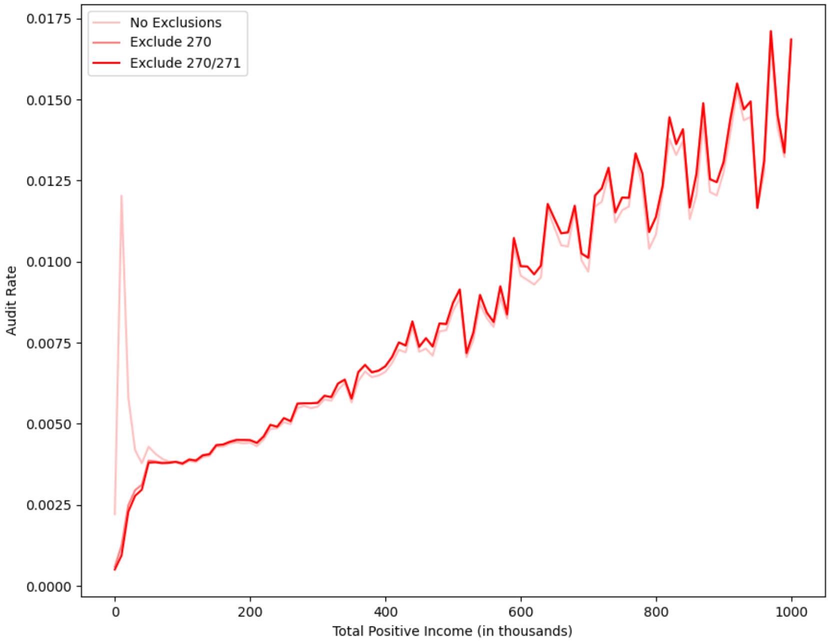


Figure D.6: Audit rate by TPI for tax year 2014 after excluding EITC-related activity codes. The lightest line corresponds to the underlying audit rate without exclusions, the next darkest to the audit rate after excluding activity code 270, and the darkest to after removing 270 and 271 (i.e. all EITC claimants).

Outcomes with respect to true TPI Finally, we recalculate no-change rates and average adjustments by corrected, rather than reported, TPI and AGI. (Note that since outcomes are measured in NRP, we have corrected incomes for nearly all taxpayers, modulo a small number of missing observations.) The outcomes are displayed in Figure D.7. Qualitatively, the TPI picture (left panel) looks similar to the right panel of Figure 5.1, but with an even clearer monotonicity pattern in average adjustment, as the downward trend in adjustments in between the 3rd-7th bins of (uncorrected) TPI is replaced by a plateau. Moreover, measured according to corrected TPI, the average adjustment is higher in the highest-income bin than according to reported TPI, but lower in the lower-income bins; in other words, the overall trend is much starker for corrected than reported TPI. The AGI picture (right panel) appears qualitatively very similar to the TPI picture, indicating that monotonicity of change rate and adjustment holds regardless of income measure, at least after correcting for the truth.

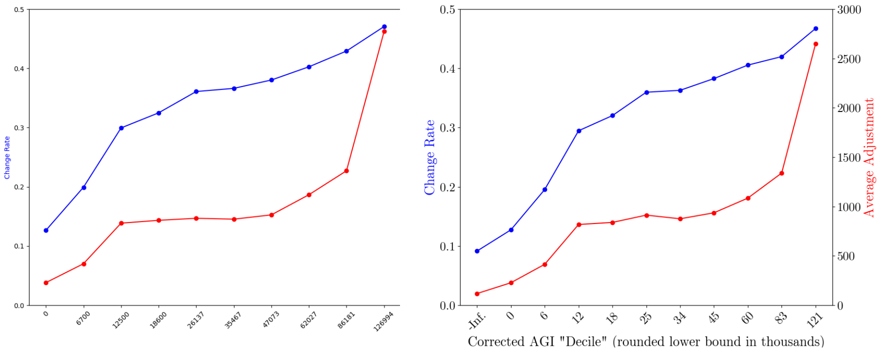


Figure D.7: The figures display outcomes — no-change rate, in blue and measured on the left y-axis, and average adjustment, in red and measured on the right y-axis — by corrected TPI (left panel) and corrected AGI (right panel).

D.5.2 Fairness methods and Modeling Choices

In this section, we display audit rate by income of classification, regression, and fairness-constrained models presented in the main paper, but with income buckets over *audit-adjusted adjusted gross income* (AA-AGI), and *audit-adjusted total positive income* (AA-TPI). This provides a robustness check to test whether models which display low audit focus on *reported* low income also do so on *true* low income populations, and if this pattern carries over to other notions of income, such as taxable (and not total) income.

Experimental Setup. For AA-TPI, we use the same income buckets as we have throughout the paper (which determine deciles on total positive income) for consistency and ease of comparison. For AA-AGI, we re-compute buckets, and also create a separate bucket for individuals with negative AGI, but note that they only make up approximately 0.7% of the population (less than 1/10 of a decile), and thus the results on this population are not directly comparable to those on the rest of the deciles due to the vastly different sample size. For both measures of income, approximately 1,000 out of 71,000 rows do not contain audit-adjusted AGI or TPI, which we exclude from the analysis.

Results The audit distributions over income deciles over AA-AGI and AA-TPI are largely similar. For AA-AGI, the boosted regressor focuses slightly less on middle-to-high income. For both AA-TPI and AA-AGI, the EO constrained classifier focuses lightly less on middle income individuals (∼47k). Regression and LDA models select a high rate for individuals with negative AA-AGI, but this is drawn from a very small percentage of the population (0.7%). Otherwise, the overall trends of audit focus for audit focus across the different classifiers remains the same.

The most notable change from reported TPI to AA-TPI and AA-AGI is the extent to which the oracle focuses on "truly" high income individuals — whereas the oracle audited up to 1% individuals with zero and middling reported TPI, from the perspective of AA-TPI and AGI, the oracle focuses almost exclusively on the upper third of the income spectrum, and most dramatically (approx 4.5%, as opposed to approx. 2% for reported TPI) on the highest income decile.

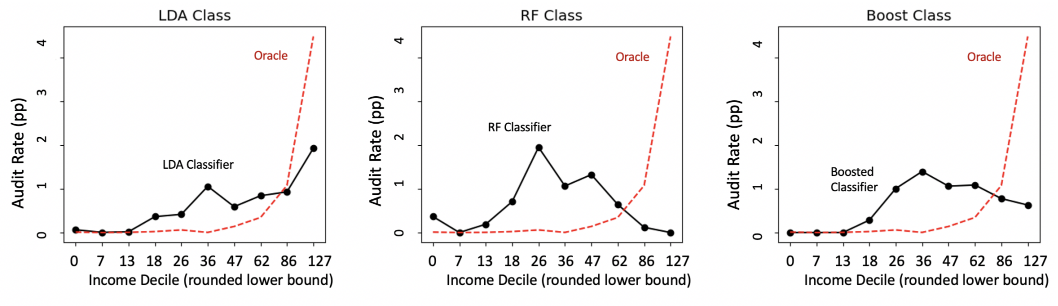


Figure D.8: Audit rate by income for classification models. From left to right: LDA classifier, Random Forest Classifier, and Boost Classifier. We use the same income deciles as presented throughout the paper for ease of comparison, but with corrected total positive income (after audit) as opposed to reported. Income decile lower bounds are given in thousands of dollars.

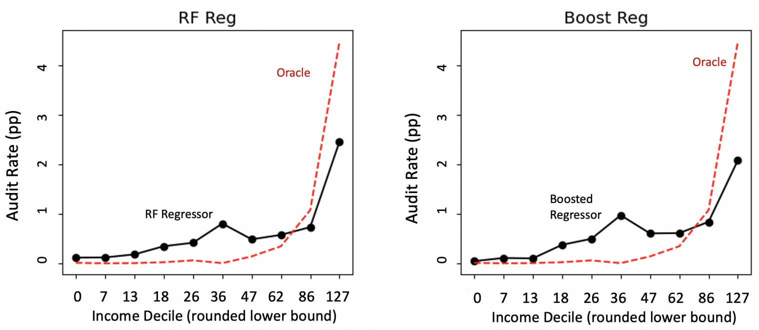


Figure D.9: Audit rate by income for regression models. We use the same income deciles as presented throughout the paper for ease of comparison, but with corrected total positive income (after audit) as opposed to reported. Income decile lower bounds are given in thousands of dollars.

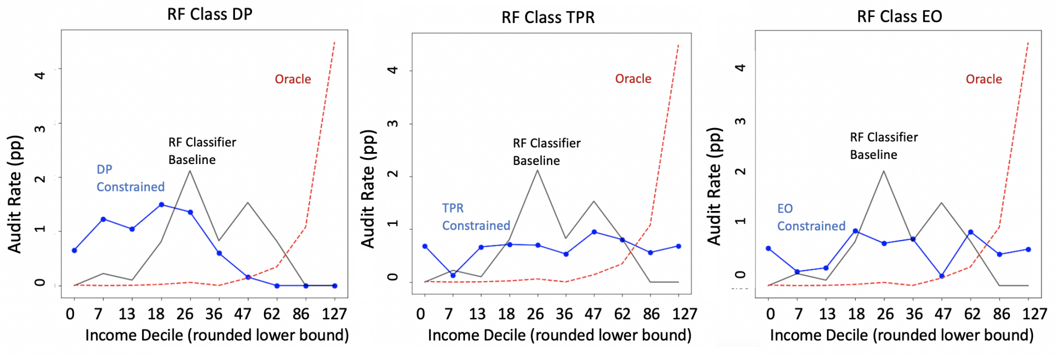


Figure D.10: Audit rate by income from in-process fairness constrained random forest models, graphed over audited corrected TPI (AA-TPI). We use the same income deciles as presented throughout the paper for ease of comparison, but with corrected total positive income (after audit) as opposed to reported.

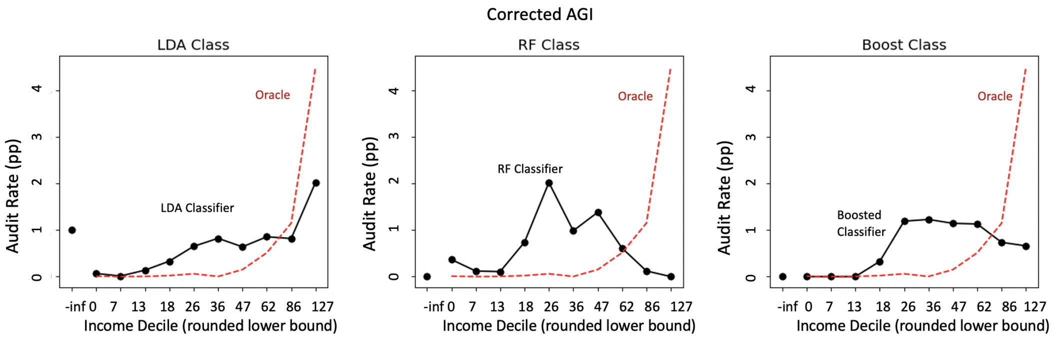


Figure D.11: Audit rate by income for classification models. From left to right: LDA classifier, Random Forest Classifier, and Boosted Classifier. We plot over 10 AGI-derived deciles (0-127k are the lower-bounds), with an additional column for the taxpayers with negative corrected AGI. Note that the first column (-inf) is not a true decile, as individuals with true negative AGI make up less than 0.7% of the population.

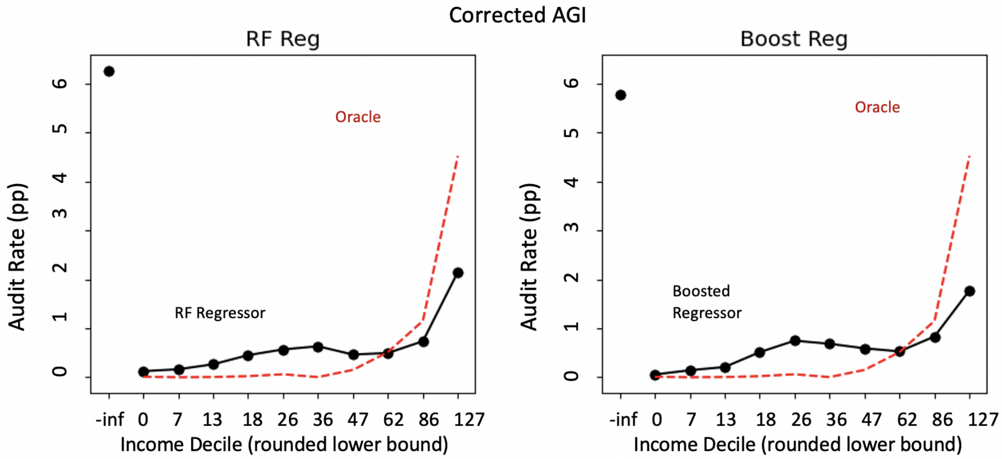


Figure D.12: Audit rate by income for regression models. We plot over 10 AGI-derived deciles (0-127k are the lower-bounds), with an additional column for the taxpayers with negative corrected AGI. Note that the first column (-inf) is not a true decile, as individuals with true negative AGI make up less than 0.7% of the population.

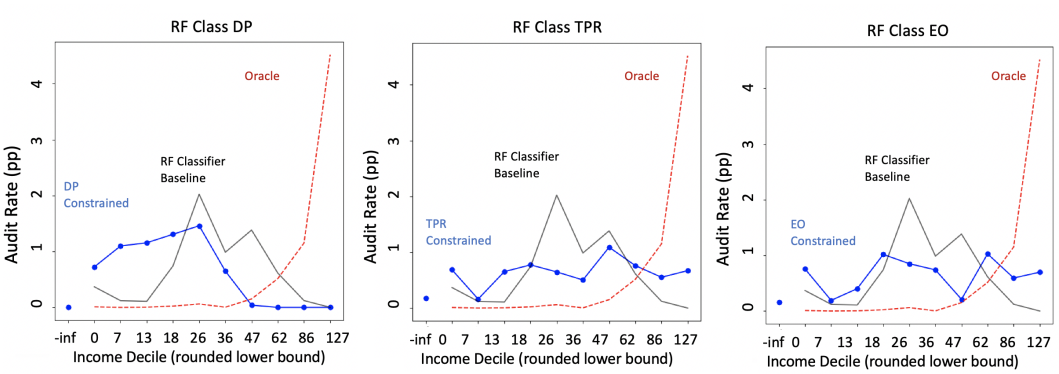


Figure D.13: Audit rate by income from in-process fairness constrained random forest models, graphed over audited corrected AGI. We plot over 10 AGI-derived deciles (0-127K are the lower-bounds), with an additional column for the taxpayers with negative corrected AGI. Note that the first column (-inf) is not a true decile, as individuals with true negative AGI make up less than 0.7% of the population. Income decile bounds are given in thousands.

D.6 Further Fairness Results

In this section, we present complete in-processing results, and also show results from another technique, specifically, post-processing techniques for enforcing fairness constraints. We also discuss why pre-processing techniques, and perhaps counterintuitively, fair ranking methods are not well-suited to our setting.

D.6.1 In-processing

As noted in Section 5.5, the in-processing results do not result in audit allocations which respect the fairness constraints the models are trained to obey, partially due to the fact that the audit allocation focuses only on the top 0.644% of predictions. First, we present (i) numerical evidence that in-process fairness constrained models do not produce allocations which respect the constraints they are trained to satisfy (Tables D.4 and D.5), (ii) we show evidence that the in-processing results did perform according to expectation, i.e., they do produce models which satisfy their respective constraints over the *full suite of predictions* on the training set, in Table D.6.

We present only numeric clarification for the fact that the allocations do not satisfy the constraints which are enforced on the model for true positive and false positive rates, as the fact that selection rate parity is not upheld is clear from the graph of the allocation (as an allocation which satisfies selection rate parity would have equal audit rate across all income groups).

We note that we present the true and false positive rates calculated over the *weighted* population—i.e. calculating all metrics taking into account the sample weight of each row—as well as over the unweighted raw data. This is due to the fact that the algorithm used to implement these results do not offer any guarantees over weighted data [8]. However, we find that the results are qualitatively similar.

	In-Process Fairness Method: False Positive Rates							
	Unweighted				Weighted (W)			
Income Bucket	Uconstr.	SR PAR	TPR PAR	EO	Unconstr. W	SR PAR W	TRP Par W	EO W
0	0.000	0.008	0.002	0.011	0.000	0.006	0.000	0.005
7	0.000	0.008	0.003	0.001	0.000	0.009	0.002	0.004
13	0.000	0.012	0.002	0.006	0.000	0.010	0.001	0.003
18	0.000	0.016	0.000	0.002	0.000	0.010	0.000	0.007
26	0.006	0.009	0.002	0.006	0.004	0.007	0.000	0.006
36	0.000	0.003	0.005	0.015	0.000	0.002	0.001	0.003
47	0.000	0.000	0.000	0.007	0.000	0.000	0.000	0.009
62	0.000	0.000	0.004	0.010	0.000	0.000	0.003	0.012
86	0.000	0.000	0.005	0.008	0.000	0.000	0.004	0.018
126	0.000	0.000	0.000	0.008	0.000	0.000	0.000	0.007

	Post-Process Fairness Method: False Positive Rates							
Income Bucket	Unconstr.	SR PAR	TPR PAR	EO	Unconstr. W	SR PAR W	TRP Par W	EO W
0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
7	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
13	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
18	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
26	0.006	0.006	0.006	0.000	0.004	0.004	0.004	0.000
36	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
47	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
62	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
86	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
126	0.000	0.000	0.000	0.020	0.000	0.000	0.000	0.022

Table D.4: We present the false positive rates by income bucket for the audit allocations generated from unconstrained and fairness-constrained random forest classifier models on the *test* set, where an audit allocation corresponds to the highest ranked predictions from each model up to a budget of 0.644% of the taxpayer population, or 1125000 audits. Unconstr. refers to an unconstrained model, SR PAR to selection rate parity, TPR PAR to true positive rate parity, and EO to equalized odds. We note that the algorithms implemented in *Fairlearn*[22] only guarantee satisfying fairness constraints in expectation on the training set, over the entire set of predictions (i.e. not simply the top 0.64%). Also note that the only column where we would expect to see equalized false positive rates is the equalized odds (EO) column(s). The top table represents results from in-process fairness methods, and the lower table from post-process fairness enforcement methods. The numbers in the left side (left four columns) of the table corresponds to the calculation on the raw data, without sample weights, and the right four columns display the calculation weighted by the sample weights, denoted with W. We present the unweighted calculation as the fairness methods do not guarantee equalized false positive rates over the weighted data, but rather only on the unweighted—however, false positive rates are not equalized with either calculation method.

	In-Process Fairness Method: True Positive Rates							
	Unweighted				Weighted (W)			
Income Bucket	Uconstr.	SR PAR	TPR PAR	EO	Unconstr. W	SR PAR W	TRP Par W	EO W
0	0.000	0.015	0.021	0.014	0.000	0.011	0.034	0.014
7	0.015	0.029	0.010	0.010	0.012	0.032	0.011	0.010
13	0.008	0.015	0.015	0.013	0.007	0.011	0.020	0.013
18	0.018	0.024	0.015	0.022	0.027	0.025	0.015	0.022
26	0.045	0.019	0.016	0.009	0.056	0.022	0.018	0.009
36	0.019	0.015	0.016	0.013	0.025	0.011	0.014	0.013
47	0.027	0.000	0.026	0.006	0.040	0.000	0.030	0.006
62	0.007	0.000	0.018	0.018	0.012	0.000	0.015	0.018
86	0.001	0.000	0.017	0.013	0.002	0.000	0.009	0.013
126	0.000	0.000	0.009	0.010	0.000	0.000	0.016	0.010

	Post-Process Fairness Method: True Positive Rates							
	Unweighted				Weighted (W)			
Income Bucket	Unconstr.	SR PAR	TPR PAR	EO	Unconstr. W	SR PAR W	TRP Par W	EO W
0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
7	0.015	0.015	0.015	0.000	0.012	0.012	0.012	0.000
13	0.008	0.008	0.008	0.000	0.007	0.007	0.007	0.000
18	0.018	0.018	0.018	0.000	0.027	0.027	0.027	0.000
26	0.045	0.045	0.045	0.000	0.056	0.056	0.056	0.000
36	0.019	0.019	0.019	0.000	0.025	0.025	0.025	0.000
47	0.027	0.027	0.027	0.000	0.040	0.040	0.040	0.000
62	0.007	0.007	0.007	0.000	0.012	0.012	0.012	0.000
86	0.001	0.001	0.001	0.000	0.002	0.002	0.002	0.000
126	0.000	0.000	0.000	0.092	0.000	0.000	0.000	0.117

Table D.5: We present the true positive rates by income bucket for the audit allocations generated from unconstrained and fairness-constrained random forest classifier models on the *test* set, where an audit allocation corresponds to the highest ranked predictions from each model up to 0.644% of the taxpayer population (i.e. around 1.1M audits). Unconstr. refers to an unconstrained model, SR PAR to selection rate parity, TPR PAR to true positive rate parity, and EO to equalized odds. Note that the only column where we would expect to see equalized true positive rates are the true positive rate parity (TPR PAR) equalized odds (EO) columns. The top table represents results from in-process fairness methods, and the lower table from post-process fairness enforcement methods. The numbers in the left side (left four columns) of the table corresponds to the calculation on the raw data, without sample weights, and the right four columns display the calculation weighted by the sample weights, denoted with W. We present the unweighted calculation as the fairness methods do not guarantee equalized true positive rates over the weighted data, but rather only on the unweighted—however, true positive rates are not equalized over income deciles in either calculation scheme. Income buckets are given in thousands.

	DP Enforc.	TPR Enforc.	EO Enforc.	EO Enforc.
Income Bucket	SRP	TPR	TPR	FPR
0	0.348	0.979	0.981	0.006
7	0.348	0.981	0.980	0.009
	0.349	0.983	0.982	0.013
18	0.348	0.985	0.985	0.007
26	0.367	0.986	0.986	0.006
36	0.350	0.982	0.982	0.005
47	0.368	0.993	0.993	0.004
62	0.368	0.996	0.995	0.004
86	0.368	0.996	0.996	0.004
126	0.366	0.990	0.991	0.003

Table D.6: We present a verification of the fact that in-process fairness techniques work as billed. From left to right, we have the selection rate by income bucket in the equalized selection rate model, the true positive rate by income bucket in the true positive parity constrained model, and the true and false positive rates by income bucket in the equalized odds constrained model. All results are presented over *all predictions* in the *training set*, not over an allocation the size of 0.644% of taxpayer population (i.e. about 1.1M audits), as in the majority of the paper. This is in order to verify the guarantees the in-processing method implemented in *FairLearn* actually provides, which is that the model will satisfy the fairness constraint desired *in expectation on the training set*, within error $2(\epsilon + \text{best\_gap})$, where best_gap is a determined at run-time and not released to the model users, and ϵ is a user-set slack parameter. We set the slack parameter to 1% in our implementation. Note that for each metric presented, all rates across income buckets are within 2% of each other. Thus, the fairness metrics are satisfied within the expected parameters of $2(\epsilon) \leq 2(\epsilon + \text{best\_gap})$. Income buckets are given in thousands.

D.6.2 Post-processing

Post-processing involves intervening at prediction time by developing group-specific thresholds for positive predictions on top of the original model to ensure a model's predictions satisfy the relevant fairness constraints. We use a method developed by Hardt et al [106] to implement this technique.

Implementation. In post-processing methods, the base random forest model is trained exactly as described in Section D.2. We again use *FairLearn* [22] to implement the post-processing technique based upon Hardt et al. [105]. Post-processing methods as implemented in *FairLearn* are not engineered to return a ranking but only a binary prediction, thus in order to accommodate creating a ranking from predictions, we multiply the binary predictions of the fair classifier (which satisfy the desired metric across groups) by the predicted probabilities from the baseline classifier in order to be able to meaningfully rank the output.

Results Figure D.14 displays audit rate by income for post-processed Random Forest classifiers to respect each of the three fairness metrics. Again, the constrained model's audit rates are in blue, the unconstrained in black, and the oracle in red dashed. The revenue, no-change rate, and cost of each are also displayed in Table 5.1.

A key takeway is that post-processing techniques are ill-fit to the audit allocation problem as they often result in minimal changes to prediction on the most confidently predicted points, which can leave aggregate audit allocations *unchanged* from the unconstrained model. Figure

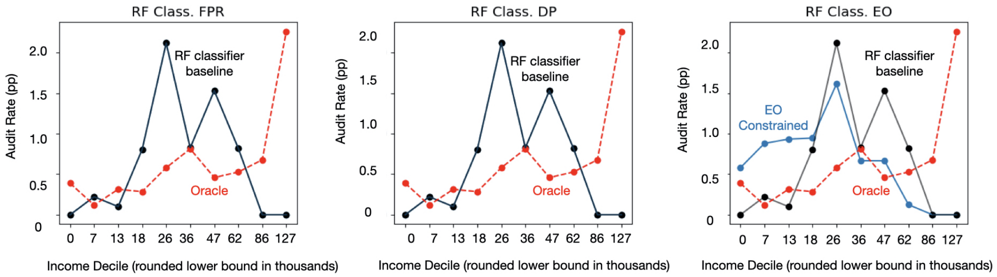


Figure D.14: Post-process fairness techniques imposed on a random forest model. From left to right: enforcing Equal True Positive Rates (FP), Demographic Parity (DP), and Equalized Odds (EO). Each blue graph depicts of the results of enforcing a fairness constraint, the black graph is the original allocation.

D.14 shows that the audit selection from post-processed models often lead to no change in aggregate audit rates (demographic parity, true positive rate parity). This is likely due to the fact that re-drawing group-specific thresholds to determine a final prediction which satisfies a fairness constraint is less likely to affect the most confidently predicted points, which we select for the top 0.644%. This is by design to keep error to a minimum, and to keep the post-processed model as similar to the original model as possible [105].

In terms of the equalized odds allocations suggested by the post-processed random forest model, it is unclear what benefits enforcing these constraints provides, as they do not satisfy the respective fairness definitions on the top 0.644% of predictions, as is noticable from the demographic parity allocation (which does not change from the baseline model). Additionally, enforcing equalized odds actually substantially increases audit focus on the lower end of the income distribution through this method, so we do not reduce audit focus on lower income individuals.

Thus, post-processing techniques are technically mismatched for the budgeted audit selection setting, and we argue, do not lead to an increase in equity.

Fair Ranking and Pre-Processing. We omit two major alternative categories of methods: *pre-processing* and *fair ranking*. Pre-processing methods alter the data before model training; this may be as simple as re-sampling the data or as involved as learning alternative representations of data that obfuscate any correlation between outcomes and sensitive features. Such methods tend to have sharp tradeoffs with accuracy [146], and often sacrifice interpretability, which may limit applicability in this setting. Fair ranking methods attempt to achieve fairness guarantees in settings where the *ranking* of individuals matter.[37], [228] While this may appear related to the audit problem, an important distinction is that in the fair ranking problem, the relative placement of items matters even beyond the decision to include or exclude them from some selection set. This is a more difficult setting than the audit problem as defined in Section 5.1, in which the precise ranking *within* audited taxpayers and separately *within* non-audited taxpayers does not matter[2] to the IRS (nor does it matter to the taxpayers). Hence, methods aimed at fair ranking are 'overkill' for our setting.

[2] This may be less true if the *budget* is not known in advance, but we do not consider such a scenario here.

D.7 Revenue-Optimal Problem as Fractional Knapsack

Given audit variables a_i, net revenues r_i, costs c_i and weights w_i, and a budget A, the revenue-optimal selection of audits is described by the following LP:

$$\begin{aligned} \text{maximize} \quad & \sum_{j=1}^{m} a_j r_j^{net} \\ \text{subject to} \quad & \sum_{j=1}^{m} a_j c_j \leq A \\ & \quad a_i \in [0, w_i], \quad \forall a_i \end{aligned}$$

Note that this is simply an instantiation of the fractional knapsack problem, which is often intuitively described as, given an option of several items with different values and weights, choosing a subset of x items to put into a "knapsack" in order to maximize the value in the knapsack given the constraint of how much a person can carry (where, in the fractional approximation, one is allowed to put a fraction of the item in the knapsack). The analogues here is the audit allocation is our knapsack, taxpayers are items to put in the knapsack, total net revenue is the value, and the cost of each taxpayer audit to the IRS is the weight. The optimal solution to this problem is a greedy selection of the objects with the best value per unit weight, i.e., in our setting, taxpayers in order of the ratio of their net tax liability returned to the IRS over the cost to the IRS to audit that individual.

D.8 Cost Calculations

We base our estimate of cost off of:
(examiner time spent on an audit)*(cost per time unit of that grade examiner)[3] averaged over income decile and *activity code* groups, which approximately corresponds to groupings of individuals based upon what tax forms they have filled out. Importantly, we base our calculation of audit cost off of *operational* IRS audits, i.e., not audits completed as a part of the National Research Program (NRP), but rather those conducted explicitly to enforce the tax code and reclaim misreported revenue. This is due to the fact that audits used for NRP are conducted differently, using more time-consuming methods, and thus relying on these cost estimates may provide a skewed picture of monetary cost to the IRS. We winsorize cost to 1st and 99th percentiles. To calculate a dollar audit budget, we calculate the yearly cost of audits using our cost metrics from operational audit data from 2010-2014, and then we average this result by five to get the average dollar cost per year in amounts proportional to our conception of cost.

[3] We note that this data recorded is grade of the lead examiner, but in some cases multiple people of different grades are involved. This is a shortcoming of the data for determining cost.

Appendix E

Model Multiplicity Appendix

E.1 A Formal Model of Multiplicity

Here, we formalize the relationship between individual-level disagreement in models and standard formulations of the bias-variance trade-off. Standard models of the bias-variance trade-off decompose loss into three components: bias, variance, and noise (e.g., [139]). Bias refers to the difference between the mean predictor (or in the case of 0-1 loss, which is simplest, the mode predictor) and the Bayes optimal predictor. Variance refers to the difference between any particular model and the mode predictor. Noise refers to the expected loss of the Bayes optimal model. For simplicity, we focus on the case of binary classification with 0-1 loss, though similar approaches could be used to characterize continuous models.

We begin by providing basic definitions in Section E.1.1. We explore the relationship between predictive multiplicity and accuracy in Section E.1.2, showing a fairly loose connection. We conclude by showing a tighter connection between predictive multiplicity and variance in Section E.1.3.

E.1.1 Basic Definitions

Suppose binary classification models come from some fixed distribution $\mathcal{M}$ (e.g., the distribution induced by random seeds, inclusion of data, etc.). Let $\mathcal{D}$ be the distribution over data. In a slight abuse of notation, we will denote a random data point as $x \sim \mathcal{D}$, and a random (data point, label) pair as $(x, y) \sim \mathcal{D}$. Let m^* be the Bayes optimal model, and let $\overline{m}$ be the mode predictor, defined as

$$\overline{m}(x) \triangleq \begin{cases} 1 & \Pr_{m \sim \mathcal{M}}[m(x) = 1] \geq \frac{1}{2} \\ 0 & \text{otherwise} \end{cases},$$

i.e., $\overline{m}$ assigns x the most probable label over the distribution of models $\mathcal{M}$. The 0-1 loss is defined as

$$L(y_1, y_2) \triangleq \begin{cases} 0 & y_1 = y_2 \\ 1 & \text{otherwise} \end{cases}.$$

For a given data point x, define

$$N(x) \triangleq \mathbb{E}_{y \mid x}\left[L(m^*(x), y)\right] \qquad \text{(noise)}$$
$$B(x) \triangleq L(\overline{m}(x), m^*(x)) \qquad \text{(bias)}$$
$$V_m(x) \triangleq L(m(x), \overline{m}(x)) \qquad \text{(variance)}$$

Note that of these three, variance is the only one that depends on the particular model m. The expected error (also known as the "loss") of a model m on dataset $\mathcal{D}$ is

$$\mathrm{err}(m, \mathcal{D}) \triangleq \mathbb{E}_{(x,y)\sim D}\left[L(m(x), y)\right].$$

Define the disagreement between two models m_1 and m_2 as

$$d(m_1, m_2) \triangleq \Pr_{x\sim\mathcal{D}}[m_1(x) \neq m_2(x)] = \mathbb{E}_{x\sim\mathcal{D}}\left[L(m_1(x), m_2(x))\right],$$

i.e., $d(m_1, m_2)$ is the probability that m_1 and m_2 disagree on a randomly drawn data point. Intuitively, predictive multiplicity flips decisions for more people as d grows. Note that $d(\cdot,\cdot)$ is symmetric and satisfies the triangle inequality:[1]

$$d(m_1, m_2) \leq d(m_1, m_3) + d(m_2, m_3)$$

A natural way to formalize predictive multiplicity for a distribution $\mathcal{M}$ over models is the expected pairwise disagreement over the distribution of models: if two models are randomly selected, how many points do they disagree on in expectation?

$$I(\mathcal{M}) \triangleq \mathbb{E}_{m_1, m_2\sim\mathcal{M}}\left[d(m_1, m_2)\right].$$

Thus, I is a formal measure of the predictive multiplicity present in a distribution $\mathcal{M}$ over models. Note that I is task-specific, since it depends on the data distribution $\mathcal{D}$. For the remainder of this paper, we will assume that $\mathcal{M}$ refers to the Rashomon set, i.e., all models in $\mathcal{M}$ have the same error L^*.

E.1.2 Predictive Multiplicity and Accuracy

Here, we present results relating predictive multiplicity to error. If all models in $\mathcal{M}$ have error L^*, Theorem E.1 upper-bounds predictive multiplicity $I(\mathcal{M})$ by $2L^*$. We will revisit this bound in Section E.1.3 to derive a tighter bound. Theorem E.2 shows that under certain assumptions, as L^* decreases (models in $\mathcal{M}$ become more accurate), predictive multiplicity approaches 0.

Theorem E.1.

$$I(\mathcal{M}) \leq 2L^*$$

Proof. We formalize the observation that two models that only make mistakes with probability

[1] In fact, d is a metric, since it is nonnegative and $d(m, m) = 0$.

p can only disagree with one another with probability at most $2p$:

$$\begin{aligned}
&\Pr_{x\sim\mathcal{D}}[m_1(x) \neq m_2(x)] \\
&\leq \Pr_{(x,y)\sim\mathcal{D}}[(m_1(x) = y \cap m_2(x) \neq y) \cup (m_1(x) \neq y \cap m_2(x) = y)] \\
&= \Pr_{(x,y)\sim\mathcal{D}}[m_1(x) = y \cap m_2(x) \neq y] + \Pr_{(x,y)\sim\mathcal{D}}[m_1(x) \neq y \cap m_2(x) = y] \\
&\leq \Pr_{(x,y)\sim\mathcal{D}}[m_2(x) \neq y] + \Pr_{(x,y)\sim\mathcal{D}}[m_1(x) \neq y] \\
&= 2L^*
\end{aligned}$$

Since this holds for any $m_1, m_2 \in \mathcal{M}$, it holds in expectation over for random $m_1, m_2 \sim \mathcal{M}$. □

Note that this characterization is essentially tight: consider the case where the true label is always $y = 1$ and $\mathcal{M}$ assigns equal probability to each of k models, where $m_i(x_i) = 0$ and $m_i(x) = 1$ for all $x \neq x_i$. Then, each model makes exactly one error (on x_i), and models m_i and m_j disagree in exactly two points (x_i and x_j). Thus, the expected number of disagreements between two randomly selected models is $2L^*(1 - {}^1\!/_k)$, taking into account the probability that the same model is selected twice.

Uniqueness. Next, we show that optimal models are in some sense unique. Our intuition is that the Bayes-optimal model is unique. This isn't strictly true: if $\Pr[y = 1] = 1/2$ given an x, then all models are Bayes-optimal, since they all have loss $1/2$. But our intuition should still hold for "predictable" problems, where y can be predicted from x better than random chance.

Assume $|\Pr_{y\sim\mathcal{D}\mid x}[y = 1] - 1/2| > c$, i.e., y can be predicted better than 50-50 chance for every x. We will show that predictive multiplicity goes to 0 as model performance approaches the Bayes risk. As before, let L^* be the loss of any model in $\mathcal{M}$. Then, the following theorem shows that as the models in $\mathcal{M}$ approach the Bayes risk R on $\mathcal{D}$, predictive multiplicity goes to 0.

Theorem E.2. *If* $|\Pr_{y\sim\mathcal{D}\mid x}[y = 1] - 1/2| > c$, *then*

$$I(\mathcal{M}) \leq \frac{L^* - R}{c}.$$

Proof.

$$
\begin{aligned}
&\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[L(m(x),y)\right]\\
&= \Pr_{(x,y)\sim\mathcal{D}}[m(x)\neq y]\\
&= \Pr_{(x,y)\sim\mathcal{D}}[m^*(x)\neq y] + \Pr_{(x,y)\sim\mathcal{D}}[m(x)\neq m^*(x)]\\
&- 2\Pr_{(x,y)\sim\mathcal{D}}[m(x)\neq m^*(x)\cap m^*(x)\neq y]\\
&= R + d(m,m^*)\\
&- 2\Pr_{(x,y)\sim\mathcal{D}}[m(x)\neq m^*(x)]\Pr_{(x,y)\sim\mathcal{D}}[m^*(x)\neq y \mid m(x)\neq m^*(x)]\\
L^* &= R + d(m,m^*)\left(1-2\Pr_{(x,y)\sim\mathcal{D}}[m^*(x)\neq y \mid m(x)\neq m^*(x)]\right)\\
d(m,m^*) &= \frac{L^*-R}{1-2\Pr_{(x,y)\sim\mathcal{D}}[m^*(x)\neq y\mid m(x)\neq m^*(x)]}\\
&\leq \frac{L^*-R}{1-2(1/2-c)}\\
&\qquad (m^* \text{ is wrong with probability at most } 1/2-c \text{ by assumption})\\
&= \frac{L^*-R}{2c}
\end{aligned}
$$

Thus, the distance between any $m\sim\mathcal{M}$ and the Bayes optimal model m^* is bounded. We can use this to bound predictive multiplicity as follows:

$$
\begin{aligned}
I(\mathcal{M}) &= \mathbb{E}_{m_1,m_2\sim\mathcal{M}}\left[d(m_1,m_2)\right]\\
&\leq \mathbb{E}_{m_1,m_2\sim\mathcal{M}}\left[d(m_1,m^*)+d(m_2,m^*)\right]\\
&\leq \frac{2(L^*-R)}{2c}\\
&= \frac{L^*-R}{c}
\end{aligned}
$$

□

E.1.3 Predictive Multiplicity and Variance

Next, we show a tight connection between predictive multiplicity and variance. Theorem E.3 shows that predictive multiplicity and variance are within a factor of 2 of one another. As a corollary, we show that reducing loss can *increase* predictive multiplicity (Corollary E.3.1). Theorem E.4 uses this result to sharpen the bound in Theorem E.1.

Let the expected variance of a model distribution $\mathcal{M}$ be

$$V(\mathcal{M}) \triangleq \mathbb{E}_{m\sim\mathcal{M},x\sim\mathcal{D}}\left[V_m(x)\right].$$

Theorem E.3.

$$\frac{1}{2}V(\mathcal{M}) \leq I(\mathcal{M}) \leq 2V(\mathcal{M})$$

Proof. We begin with an upper bound on predictive multiplicity.

$$\begin{aligned} I(\mathcal{M}) &= \mathbb{E}_{m_1,m_2\sim\mathcal{M}}\left[d(m_1,m_2)\right] \\ &\leq \mathbb{E}_{m_1,m_2\sim\mathcal{M}}\left[d(m_1,\overline{m}) + d(m_2,\overline{m})\right] \\ &= 2\mathbb{E}_{m\sim\mathcal{M}}\left[d(m,\overline{m})\right] \\ &= 2\mathbb{E}_{m\sim\mathcal{M}}\left[\mathbb{E}_{x\sim\mathcal{D}}\left[L(m,\overline{m})\right]\right] \\ &= 2V(\mathcal{M}) \end{aligned}$$

We derive a lower bound with the following observation: if m disagrees with the mode predictor $\overline{m}$ on an instance x, then m must disagree on x with at least half of the models in $\mathcal{M}$.[2] Formally, we can write this as

$$m(x) \neq \overline{m}(x) \implies \Pr_{m'\sim\mathcal{M}}[m(x) \neq m'(x)] \geq \frac{1}{2},$$

which implies

$$\Pr_{x\sim\mathcal{D}}[m(x) \neq \overline{m}(x)] \leq 2 \Pr_{m'\sim\mathcal{M},x\sim\mathcal{D}}[m(x) \neq m'(x)]. \tag{E.1}$$

Using this, we have

$$\begin{aligned} V(\mathcal{M}) &= \mathbb{E}_{m\sim\mathcal{M},x\sim\mathcal{D}}\left[V_m(x)\right] \\ &= \mathbb{E}_{m\sim\mathcal{M}}\left[\Pr_{x\sim\mathcal{D}}[m(x) \neq \overline{m}(x)]\right] \\ &\leq 2\mathbb{E}_{m\sim\mathcal{M}}\left[\Pr_{m'\sim\mathcal{M},x\sim\mathcal{D}}[m(x) \neq m'(x)]\right] && \text{(by (E.1))} \\ &= 2\mathbb{E}_{m,m'\sim\mathcal{M}}\left[\Pr_{x\sim\mathcal{D}}[m(x) \neq m'(x)]\right] \\ &= 2\mathbb{E}_{m,m'\sim\mathcal{M}}\left[d(m,m')\right] \\ &= 2I(\mathcal{M}) \end{aligned}$$

Putting this together, we have

$$\frac{1}{2}V(\mathcal{M}) \leq I(\mathcal{M}) \leq 2V(\mathcal{M}).$$

□

This shows a fairly tight connection between model variance and predictive multiplicity, which can help our intuition in a few ways. First, we see that increasing accuracy by increasing variance and reducing bias (e.g., using a more complex model class) can actually *increase* predictive multiplicity, consistent with empirical findings [24, 26]. Second, we see that efforts to decrease model variance (e.g., more data) should *reduce* predictive multiplicity. This yields the following result:

Corollary E.3.1. *Reducing loss by decreasing bias and increasing variance can increase predictive multiplicity.*

[2] By "half," we mean models that account for at least half the probability mass of $\mathcal{M}$.

Deriving a tighter relationship between predictive multiplicity and accuracy. We can use this insight on the connection between predictive multiplicity and accuracy to improve the bound in Theorem E.1. As before, let L^* be the loss of any model in $\mathcal{M}$. Let R be the Bayes risk, and let B be the bias of $\mathcal{M}$ (i.e., the error of the mode predictor $\overline{m}$).

Theorem E.4.

$$I(\mathcal{M}) \leq 2[L^* - R(1 - 2B)]$$

Proof. We begin with decomposition of error into noise, bias, and variance from [62].

$$\begin{aligned}
L^* &= \mathbb{E}_{(x,y)\sim\mathcal{D},m\sim\mathcal{M}}\left[L(m(x), y)\right] \\
&= (2 \cdot \Pr_{x\sim D}[\overline{m}(x) = m^*(x)] - 1)\mathbb{E}_{x\sim D}\left[N(x)\right] + \mathbb{E}_{x\sim\mathcal{D}}\left[B(x)\right] \\
&+ \mathbb{E}_{x\sim\mathcal{D},m\sim\mathcal{M}}\left[(-1)^{\mathbb{1}_{\overline{m}(x)\neq m^*(x)}}V_m(x)\right] \\
&= (2(1 - d(\overline{m}, m^*) - 1)R + d(\overline{m}, m^*) + \mathbb{E}_{x\sim\mathcal{D},m\sim\mathcal{M}}\left[(1 - 2B(x))V_m(x)\right] \\
&= (1 - 2B)R + B + V(\mathcal{M}) - 2\mathbb{E}_{x\sim\mathcal{D},m\sim\mathcal{M}}\left[B(x)V_m(x)\right] \\
&= (1 - 2B)R + B + V(\mathcal{M}) - 2\Pr_{x\sim\mathcal{D},m\sim\mathcal{M}}[B(x) = 1 \cap V_m(x) = 1] \\
&= (1 - 2B)R + B + V(\mathcal{M}) - 2\Pr_{x\sim\mathcal{D}}[B(x) = 1]\Pr_{m\sim\mathcal{M}}[V_m(x) = 1 \mid B(x) = 1]
\end{aligned}$$

Note that for any x, $\Pr_{m\sim\mathcal{M}}[V_m(x) = 1] \leq \frac{1}{2}$ because by definition of the mode predictor, the probability a random model disagrees with the mode predictor on a given x as at most a half. Since this holds for every x, this is true conditioned on $B(x) = 1$, so

$$\Pr_{m\sim\mathcal{M}}[V_m(x) = 1 \mid B(x) = 1] \leq \frac{1}{2}.$$

Thus, we have

$$\begin{aligned}
L^* &= (1 - 2B)R + B + V(\mathcal{M}) - 2\Pr_{x\sim\mathcal{D}}[B(x) = 1]\Pr_{m\sim\mathcal{M}}[V_m(x) = 1 \mid B(x) = 1] \\
&\geq (1 - 2B)R + B + V(\mathcal{M}) - \Pr_{x\sim\mathcal{D}}[B(x) = 1] \\
&= (1 - 2B)R + B + V(\mathcal{M}) - B \\
&= (1 - 2B)R + V(\mathcal{M}) \\
&\geq (1 - 2B)R + \frac{1}{2}I(\mathcal{M}) && \text{(By Theorem E.3)}
\end{aligned}$$

Rearranging yields the desired result:

$$I(\mathcal{M}) \leq 2[L^* - R(1 - 2B)].$$

□

Note that $B \leq \frac{1}{2}$, since a model that deterministically predicts the more likely class achieves loss at most $\frac{1}{2}$. As a result, Theorem E.4 immediately implies Theorem E.1.