Synthetic Motion Data Generation for Exoskeleton Personalization

Diya Dinesh

Date of Submission
December, 2025

School of Computer Science
Carnegie Mellon University
Pittsburgh PA 15213

COMMITTEE

Adrian Krieger

Changseob Song

Inseung Kang

Submitted in partial fulfillment of the requirements
for the Senior Honors Thesis

Keywords: machine learning; generative models; diffusion models; human gait; exoskeleton assistance; personalized gait synthesis; joint-angle time series; biomechanics dataset

Abstract

Human exoskeletons need subject-specific gait data across a range of walking speeds, but collecting high-quality motion capture for every new user and condition is expensive, time-consuming, and difficult to scale. This thesis explores whether a generative model can, given a short sample of one person walking at a single speed, synthesize that same person's gait at many other speeds in a way that is accurate enough to be useful for exoskeleton personalization. A diffusion-based generative model was built on top of the GaitDynamics transformer encoder and trained on the Camargo et al. biomechanics dataset of treadmill walking, using six leg joint angles (left/right hip, knee, ankle) from able-bodied subjects. The model conditions on a subject's motion at a source speed together with the desired target speed, using a BiLSTM subject encoder and FiLM-style conditioning to modulate the transformer denoiser.

The model is evaluated on held-out able-bodied subjects and on stiff-knee patients that are never seen during training. Quantitatively, the work measures joint-angle error and defines a subject personalization metric. The model consistently generates realistic, speed-appropriate gait and achieves low joint-angle errors on held-out able-bodied subjects, with subject-aware conditioning outperforming a randomized conditioning baseline. On the stiff-knee patients, a model trained only on able-bodied data can still approximate patient-specific gait trends, suggesting that diffusion-based personalized gait generation is a promising direction for exoskeleton training and control design.

1 Introduction

Human exoskeletons and gait-assistive devices increasingly rely on precise, subject-specific models of human walking. Controllers that provide too much or too little assistance, or that fail to anticipate how a particular person walks at different speeds, can reduce comfort, increase metabolic cost, or even destabilize the user. The personalization of exoskeleton control systems holds transformative potential for enhancing mobility and quality of life for individuals with movement im- pairments. In practice, personalizing exoskeletons to individuals often requires repeated motion capture sessions and experiments with systems such as the Vicon motion capture system and Theia Markerless, which are expensive, time-consuming, and difficult to scale beyond a small set of subjects and conditions. This creates a gap between the possibilities of personalized exoskeleton control and what is feasible in real-world deployment.

A natural way to bridge this gap is to train generative models on human gait that can synthesize realistic joint-angle trajectories across subjects and walking speeds. Instead of collecting full motion capture data for every new user and every new speed, a generative model would take a short sample of that user walking at a single speed and generate plausible trajectories for other speeds. For exoskeleton design and control, this kind of synthetic data could serve as a starting point for controller tuning and simulation before expensive experiments are run on hardware.

Recent advances in generative modeling have shown that it is possible to learn rich distributions over human motion sequences. Models based on transformers and diffusion processes, in particular, have demonstrated the ability to produce realistic, temporally coherent motion and to incorporate conditioning signals such as actions, text, or partial observations. At the same time, personalization in exoskeleton control has primarily been approached through subject-specific optimization or adaptive control rather than through explicit generative modeling of how an individual's gait varies with walking speed. To date, there appears to be relatively little work that combines these two perspectives to study personalized, speed-conditioned gait generation specifically in the context of exoskeletons.

This thesis focuses on the following problem: given a short sample of a subject's lower-limb joint angles at one treadmill walking speed, can a generative model synthesize that same subject's gait at other speeds with sufficient accuracy to be useful for exoskeleton personalization? Concretely, the work uses the Camargo et al. biomechanics dataset of treadmill walking, which contains multi-speed recordings from able-bodied subjects and data collected from individuals with stiff-knee gait at the MetaMobility lab. The model operates on six leg joint angles (left/right hip, knee, and ankle) and is trained only on able-bodied data, then evaluated on unseen able-bodied subjects and on stiff-knee patients.

To address this problem, the thesis builds a conditional diffusion model that takes as input a subject's motion at a source speed together with a desired target speed and generates a full joint-angle trajectory at that target speed. The model combines a subject encoder, which maps the source sequence into a subject-specific representation, with speed embeddings and a transformer-based denoiser that produces joint-angle sequences consistent with the requested speed. Training uses a standard denoising diffusion objective in joint-angle space.

The model is evaluated along two dimensions that are directly relevant to personalization. First, joint-angle accuracy is quantified using a multi-cycle mean average error (MAE) in degrees between generated and real trajectories. Second, the thesis introduces personalization metrics that test whether the model is most accurate for the correct subject. A subject-aware, speed-conditioned model is compared against a randomized-subject, speed-conditioned baseline to test whether explicit subject conditioning improves personalization. Generalization to unseen patient gait is studied by applying the able-bodied-trained model to stiff-knee subjects and analyzing whether the generated trajectories capture key trends in their joint motion.

In summary, this thesis:

1. Formulates personalized, speed-conditioned gait synthesis as a diffusion-based generative modeling problem on joint-angle time series.

2. Proposes a subject-aware conditional diffusion architecture that combines a subject encoder, speed embeddings, and a transformer-based denoiser.

3. Develops an evaluation pipeline for gait generation that includes joint-angle error metrics, multi-cycle analyses, and personalization-oriented ranking metrics.

4. Demonstrates empirically that subject-aware conditioning improves personalization over a randomized-subject baseline on held-out able-bodied subjects

5. Demonstrates a model trained only on able-bodied gait can approximate key stiffness-related trends in stiff-knee patient gait.

These results provide an initial step toward using generative models of personalized gait as a tool for exoskeleton training, simulation, and control design.

2 Background and Prior Work

I conducted an extensive review of methods used to generate synthetic motion data, the main ones are outlined in the bibliography below. These methods included diffusion models, Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), Seq2Seq models, and physics-based simulations. We found that early GAN-based models, specifically HP-GAN, helped generate realistic motion data but had difficulties with fine-tuning to produce diverse samples and had instability in training. VAE-based models, including Action2Motion and ACTOR, demonstrated improved diversity and training stability. However, complexities remained in fine-tuning to avoid producing overly smooth, less detailed motions. Transformer-based models, such as PoseGPT, leveraged causal attention and latent autoregression for strong temporal modeling, but required large datasets and lacked explicit physical constraint enforcement. More recently, diffusion models have emerged as the leading approach, with works like MotionDiffuse, HumanMotionDiffusion, and DiffMotion demonstrating more realistic and diverse outputs and increased training stability.

Based on existing literature, diffusion models were identified as the best choice for this project. The stability and quality of the model combined with its ability to address physical constraints made it ideal for generating motion data.

Most directly related to this thesis is GaitDynamics, a generative foundation model for human walking and running. GaitDynamics is trained on a large multi-study dataset of human gait and learns to jointly model center-of-mass velocities, full-body joint angles, joint angular velocities, and ground reactive forces over short time windows. The model uses a Transformer-based diffusion denoiser with a cosine noise schedule, min–max normalization per parameter, and rotary positional encodings. Conditioning is expressed primarily through known channels, for example, kinematics or COM velocity can be treated as "observed" dimensions, and the model inpaints the remaining channels. In their speed-controlled experiments, COM velocity is used as the main conditioning signal to generate kinematics and kinetics at different running speeds.

GaitDynamics demonstrates several capabilities that are highly relevant to exoskeletons such as estimating GRFs from kinematics, predicting how gait modifications affect knee loading without additional experiments, and generating plausible kinematic and kinetic changes across speeds. However, the model is trained as a population-level foundation model. While it implicitly spans a wide range of subjects and speeds, the paper does not focus on explicit personalization tasks such as: "given joint-angle trajectories for a specific subject at one speed, generate that same subject's gait at unseen speeds and quantify subject-specific accuracy relative to a randomized-subject baseline. And it also doesn't directly target exoskeleton control. Instead, it supports downstream tasks like GRF estimation and what-if analyses of gait modifications.

This thesis builds on these foundations by adapting a diffusion-based architecture in the same line as GaitDynamics to a lower-limb joint-angle setting, introducing explicit subject-aware conditioning alongside speed conditioning, and evaluating how well the model can generate personalized joint-angle trajectories across walking speeds compared to a randomized-subject baseline.

3 Methodology

3.1 Dataset

This work uses two datasets. The first is an open-source lower-limb biomechanics dataset from Camargo et al., which includes treadmill walking at multiple treadmill speeds for able-bodied subjects. This dataset includes 22 subjects walking at speeds from 0.5 m/s to 1.85 m/s in 0.05 m/s increments. The second dataset is joint angle data collected at the MetaMobility Lab of 3 Stiff-Knee patient's data at 0.4 m/s, 0.7 m/s, 1.0 m/s, 1.3 m/s. Each trial consists of approximately 30 seconds of level-ground walking with lower-limb joint kinematics derived from motion capture and inverse kinematics. For this thesis, only the joint angles for the left and right hip, knee, and ankle are used, yielding six features per time step.

The raw 30-second trials are resampled to a uniform 20 Hz sampling rate to remove noise from the data and capture just the overall gait patterns for each subject and speed pairing, resulting in sequences of 600 time steps per trial. This resampling is implemented by taking regularly spaced frames from the higher-frequency motion capture data, which reduces sequence length and computational cost while preserving smooth joint trajectories. If a trial is shorter than 600 time steps, zero padding is added at the end to reach the required length and longer trials are truncated to 600 time steps. Each one is associated with a subject ID and treadmill speed.

Before training, each joint angle is standardized using global statistics computed over the training windows: $x' = \frac{x-\mu}{\sigma}$ where ($\mu \in \mathbb{R}^6$) and ($\sigma \in \mathbb{R}^6$) are the mean and standard deviation across all training time steps and windows. These normalization statistics are stored with the trained checkpoint and reused for sampling and evaluation.

3.2 Task Definition

The core task is personalized, speed-conditioned gait generation. Formally, let:

- $s \in \mathcal{S}$ represents a subject.
- $v \in \mathcal{V}$ represents a treadmill walking speed (m/s).
- $x_{s,v} \in \mathbb{R}^{T \times F}$ represents a 30-second joint-angle sequence with $T = 600$ time steps and $F = 6$ joint-angle channels.

Given a source sequence ($x_{s,v_{\mathrm{src}}}$) of subject (s) walking at speed (v_{src}), and a target speed (v_{tgt}), the goal is to generate a sequence: ($\hat{x}_{s,v_{\mathrm{tgt}}}$) $\in \mathbb{R}^{T \times F}$ that represents subject s walking at the target speed (v_{tgt}), while preserving subject-specific gait characteristics (e.g., overall pattern of joint trajectories).

The training pairs of speeds per subject are constructed using the PairDataset wrapper over the windowed dataset:

- For each subject with at least two distinct speeds, all data is bucketed by (subject, speed).
- The training pairs are of the form: ($x_{s,v_1}, x_{s,v_2}, v_1, v_2$), where ($v_1 \neq v_2$).
- During training, both directions are included: each pair is used once with ($v_{\mathrm{src}} = v_1, v_{\mathrm{tgt}} = v_2$) and once with ($v_{\mathrm{src}} = v_2, v_{\mathrm{tgt}} = v_1$).

This defines a conditional generative modeling problem:

$$p_\theta(x_{s,v_{\mathrm{tgt}}} \mid x_{s,v_{\mathrm{src}}}, v_{\mathrm{src}}, v_{\mathrm{tgt}}),$$

where (θ) are the model parameters learned from all such pairs across all training subjects.

3.3 Model Architecture

The architecture follows a diffusion-model design with a transformer-based denoiser built on GaitDynamics, augmented with explicit subject and speed-aware conditioning.

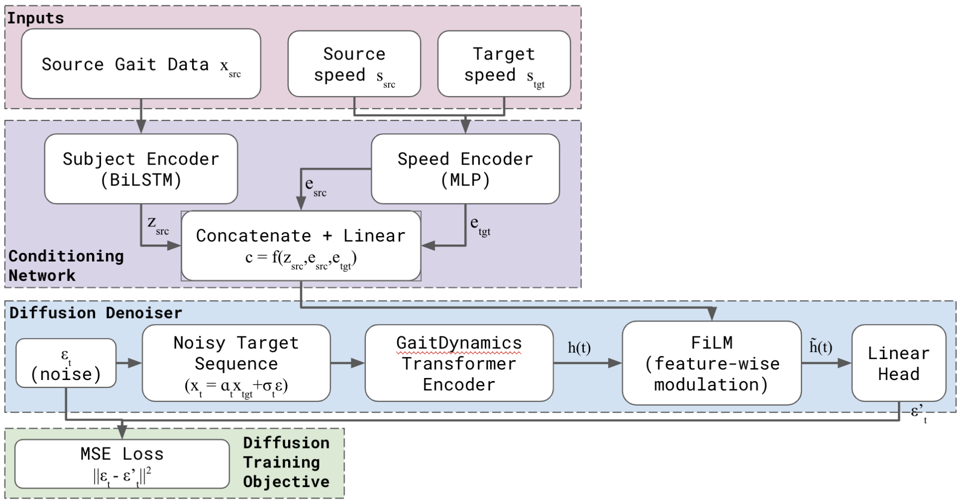


Figure 1:

3.3.1 Base Transformer Denoiser

The denoiser operates on full sequences of length (T = 600) with (F = 6) features. The GaitDynamics-style transformer encoder is adapted as the core sequence model with the following parameters:

- Hidden dimension ($d_{\text{model}} = 128$),
- 4 self-attention layers,
- 4 attention heads per layer,
- Feed-forward hidden size taken from the GaitDynamics configuration,
- Layer normalization and dropout as in the original GaitDynamics implementation.

This transformer encoder is wrapped in GDEncoderWrapper and exposed as EpsPredictor, which maps noisy sequences and conditioning vectors to predicted noise:

$$\epsilon_\theta : \mathbb{R}^{T\times F} \times \mathbb{R}^{d_{\text{cond}}} \to \mathbb{R}^{T\times F}$$

.

3.3.2 Subject and Speed Conditioning

Conditioning is implemented through a separate encoder that encodes both the subject trajectory and the pair of speeds:

1. Subject–speed encoder (SubjectEncoder)
   - Input: the normalized source sequence ($x_{s,v_{\text{src}}}$) and scalar speeds (v_{src}, v_{tgt}).

- A BiLSTM model is used to generate a latent representation of the subject's gait data ($x_{s,v_{\mathrm{src}}}$).
- The scalar speeds are encoded with a small MLP and concatenated with the encoded subject gait data. This data is then passed through a linear layer.
- The output is a fixed-length conditioning vector ($c \in \mathbb{R}^{d_{\mathrm{cond}}}$) (with ($d_{\mathrm{cond}} = 64$)).

2. FiLM conditioning
   The conditioning vector (c) is injected into the transformer via Feature-wise Linear Modulation (FiLM) layers: $\mathrm{FiLM}(h_t, c) = (1 + \gamma(c)) \odot h_t + \beta(c)$, where ($h_t \in \mathbb{R}^{d_{\mathrm{model}}}$) is the transformer hidden state at time (t), and (γ, β) are learned functions of (c) implemented by a small MLP. The FiLM layers allow the conditioning signal to scale and shift the transformer's hidden features across all time steps. This ensures that the output is conditioning on the subject's personal gait data and speed requirements.

3. Output head
   After the final transformer layer, a linear projection maps the hidden states back to the joint-angle feature dimension:$\hat{\epsilon} = Wh + b \in \mathbb{R}^{T \times F}$, which is interpreted as the predicted noise ($\hat{\epsilon}$) at each time step and feature.

3.3.3 Noise Schedule

A variance-preserving (VP) diffusion schedule is used, implemented by VPScheduler:

- Number of diffusion steps ($T_{\mathrm{diff}} = 1000$)
- (α_t) and (σ_t) are precomputed and stored on device
- The forward process at time (t) is $x_t = \alpha_t x_0 + \sigma_t \epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$.

The model is trained to approximate the reverse process by predicting (ϵ) given ((x_t, c)).

3.4 Training Methodology

3.4.1 Objective

The learning objective is a weighted denoising loss on the diffusion noise:

1. A time index t is sampled uniformly from $\{0, \ldots, T_{\mathrm{diff}} - 1\}$.
2. A target sequence $x_{s,v_{\mathrm{tgt}}}$ from the pair is treated as the clean sample x_0.
3. Noise is sampled as $\epsilon \sim \mathcal{N}(0, I)$, and a noisy input is formed:

$$x_t = \alpha_t x_0 + \sigma_t \epsilon.$$

4. The conditioning vector c is computed from the source sequence $x_{s,v_{\mathrm{src}}}$ and speeds ($v_{\mathrm{src}}, v_{\mathrm{tgt}}$).

5. The model predicts $\hat{\epsilon}_\theta = \epsilon_\theta(x_t, c)$.

The loss is then:

$$\mathcal{L}(\theta) = \mathbb{E}_{t,\epsilon}\left[\frac{1}{TF}\sum_{t,f} w_f \left(\hat{\epsilon}_{t,f} - \epsilon_{t,f}\right)^2\right],$$

where w_f are per-joint weights. The knee joints are given $2\times$ **weight** relative to hip and ankle channels to emphasize accurate knee kinematics, which are particularly important for exoskeleton assistance.

3.4.2 Training Setup

This model was trained on the Able-Bodied dataset with a 60-20-20 split. 60% of the data was used for training, and 20% for testing and validation each. There were no overlaps of subjects between the three.

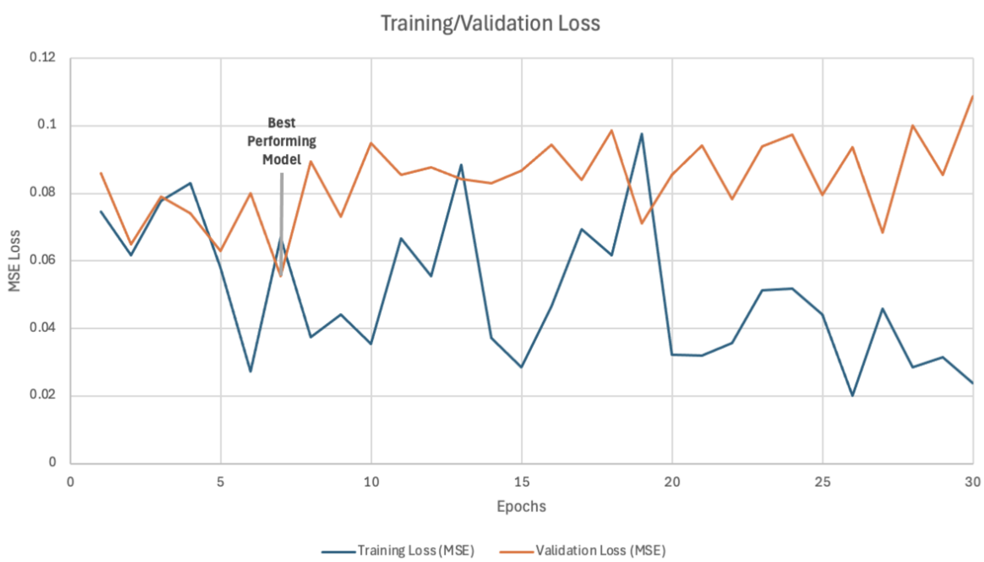


Figure 2:

Training is performed in PyTorch with the following configuration:

- Optimizer: AdamW over the combined parameters of the transformer and conditioning encoder,
- Learning rate: 2×10^{-4},
- Batch size: 32 paired sequences,
- Training/validation split:
  - Training pairs: `PairDataset(train_ds, mode="balanced_random", min_delta_speed=0.05, max_items=100000)`,

– Validation pairs: `PairDataset(val_ds, mode="all_pairs", min_delta_speed=0.05)`.

The "balanced_random" pairing mode for training ensures that for each subject, a variety of speed pairs is sampled rather than exhaustively enumerating all pairs. For validation, "all_pairs" is used to systematically measure performance across all pairs of speeds for each validation subject.

At the end of each epoch, the model is evaluated on the validation set using the same denoising loss. The checkpoint with the **lowest validation loss** is saved. All experiments reported in the Results section use this best validation checkpoint.

3.5 Sampling Methodology

At test time, the goal is to generate a target-speed sequence $\hat{x}_{s,v_{\text{tgt}}}$ for each subject and speed pair, starting from a real source sequence at another speed.

3.5.1 Input Preparation

For a given subject s and source speed v_{src}:

1. Read the source sequence $x_{s,v_{\text{src}}}$ from the corresponding 30-second CSV
2. If the sequence is shorter than 600 time steps, it is zero-padded, if longer, it is truncated.
3. Normalize using stored training statistics:
$$x'_{\text{src}} = \frac{x_{s,v_{\text{src}}} - \mu}{\sigma}.$$
4. Form a batch of size 1: $x'_{\text{src}} \in \mathbb{R}^{1\times T\times F}$.
5. Create scalar tensors for the source and target speeds:
$$v_{\text{src}}, v_{\text{tgt}} \in \mathbb{R}^1.$$

3.5.2 Speed-Aware DDIM Sampling

Sampling uses a deterministic DDIM-style procedure that starts from a noisy version of the *source* sequence, with the noise level chosen as a function of the difference between source and target speeds. Intuitively, when v_{tgt} is close to v_{src}, the sampler only applies a small amount of noise and performs a shallow reverse diffusion; when the speeds differ more, the sampler starts from a noisier version of the source and performs a longer reverse trajectory.

Let $\Delta v = |v_{\text{tgt}} - v_{\text{src}}|$ denote the speed difference. A maximum relevant speed difference $v_{\max}$ is set in code as `max_delta = 1.5` m/s. The fraction of the diffusion horizon to use is
$$\text{frac} = \text{clip}\left(\frac{\Delta v}{v_{\max}}, 0, 1\right),$$
and the starting diffusion step is chosen as
$$t_{\text{start}} = \lfloor (T_{\text{diff}} - 1) \cdot \text{frac} \rfloor,$$

where T_{diff} is the total number of diffusion steps (1000 in this work). A sequence of timesteps $(t_{K-1}, \ldots, t_0)$ is then constructed by linearly interpolating from t_{start} down to 0 over K sampling steps (here $K = 100$).

The sampling procedure can be summarized as follows:

1. Draw Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$ with shape $(1, T_{\text{out}}, F)$, where $T_{\text{out}} = 600$.
2. Compute $\alpha_{t_{\text{start}}}, \sigma_{t_{\text{start}}}$ from the VP scheduler.
3. Form the initial noisy state by combining the normalized source sequence x'_{src} with noise:
$$x_{t_{\text{start}}} = \alpha_{t_{\text{start}}} x'_{\text{src}} + \sigma_{t_{\text{start}}} \epsilon.$$
4. For each subsequent timestep t in the sequence $t_{K-1}, \ldots, t_0$ (from largest to smallest):
   (a) Compute (α_t, σ_t) from the scheduler.
   (b) Recompute the conditioning vector
   $$c = \text{CondEncoder}(x'_{\text{src}}, v_{\text{src}}, v_{\text{tgt}}).$$
   (c) Predict noise $\hat{\epsilon}_\theta = \epsilon_\theta(x_t, c)$.
   (d) Estimate the corresponding clean sample:
   $$\hat{x}_0 = \frac{x_t - \sigma_t \hat{\epsilon}_\theta}{\alpha_t + 10^{-8}}.$$
   (e) If $t > 0$, compute α_{t^-} for the previous timestep t^- in the DDIM schedule and update x accordingly (deterministic DDIM update); if $t = 0$, set $x_{\text{gen}} = \hat{x}_0$.

Finally, the generated sequence is denormalized to recover joint angles in the original units:
$$\hat{x}_{s,v_{\text{tgt}}} = x_{\text{gen}} \odot \sigma + \mu.$$

When Δv is small, t_{start} is close to 0, so $x_{t_{\text{start}}}$ is only mildly perturbed from x'_{src} and the reverse diffusion trajectory is short. When Δv is large, t_{start} is closer to T_{diff}, so $x_{t_{\text{start}}}$ is much noisier and the model performs a longer reverse trajectory. This design encourages the model to reuse the source sequence for nearby speeds while allowing larger changes when the target speed is substantially different.

3.5.3 Temporal Alignment and Windowing

To compare generated and real trajectories at the target speed:

1. Load the **real** 30-second sequence $x_{s,v_{\text{tgt}}}$ for the same subject and speed.
2. Extract the last **10 seconds** (i.e., the final 200 time steps at 20 Hz) from both generated and real sequences. The last 10 seconds are used to focus evaluation on the region where the subject is in steady-state walking, include multiple complete gait cycles, and roughly match the 5-second clip lengths commonly used in prior motion-generation work, while reducing variance by averaging over more cycles.

3. Perform phase alignment:

   - Detect the first prominent peak in the trajectory of each joint angle in both generated and real sequences,
   - Compute the shift needed to align the peaks,
   - Apply a circular shift to the generated sequence along the time dimension.

After alignment, both sequences are resized to a common length using linear interpolation (`length_eq`) so that the evaluation metrics can be computed elementwise.

3.6 Evaluation Metrics

The model is evaluated with metrics designed to capture per-joint accuracy at the multi-cycle-level similarity and personalization.

3.6.1 Cycle-Level Multi-Cycle MAE

To explicitly evaluate gait cycle structure without error propagation, a **multi-cycle mean absolute error (MAE)** metric is used:

1. Gait cycles are identified for each joint angle data in each sequence by finding troughs in the negated joint-angle waveform using `find_peaks`, filtering out "double" troughs based on duration statistics.

2. Choose candidate real and generated cycles which are constructed by pairing adjacent trough indices.

3. For each evaluation, 5 distinct real-generated cycle pairs are selected by matching cycles with similar duration and center time.

4. For each matched pair:

   - The real and generated cycles are individually **time-normalized** to a fixed number of points (101 samples from 0-100% gait cycle) using linear interpolation.
   - The cycle-level MAE is computed:

$$\mathrm{MAE}^{(\mathrm{cycle})} = \frac{1}{TF}\sum_{t=1}^{T}\sum_{j=1}^{F}\left|\hat{x}_{t,j}^{(\mathrm{cycle})} - x_{t,j}^{(\mathrm{cycle})}\right|.$$

5. The overall multi-cycle MAE is the average of these per-cycle MAEs, optionally aggregated over subjects and speed conditions.

This metric emphasizes the shape and phase of each gait cycle rather than only frame-wise error. This is averaged across all (subject, source speed, target speed) combinations.

3.6.2 Personalization Metrics

To quantify **personalization**, whether the generated trajectory resembles the target subject more than other subjects at the same speed, the following ranking-based metrics are used. For a fixed target speed v_{tgt}, the generated sequence $\hat{x}_{s,v_{\text{tgt}}}$ is compared against:

- the real sequence of subject s at speed v_{tgt}, and
- real sequences of other subjects at the same speed.

Using the multi-cycle MAE described above, all candidates are ranked by error. The subject personalization rank is defined as the rank of subject s's real trajectory (1 = best). This rank is averaged across speeds and test subjects.

In the Results section, these metrics are used to compare:

- the subject-aware conditioned model, and
- a randomized-subject (no subject identity conditioning) variant where the subject information in the conditioning path is replaced or shuffled, to assess the value of explicit personalization.

4 Results

This section evaluates the diffusion model on held-out able-bodied subjects and on stiff-knee patient gait subjects, using the metrics and protocols defined in Methods. All quantitative results are computed using the best validation checkpoint and the deterministic speed-aware sampling procedure described in the Methods

4.1 Evaluation Protocol

Unless otherwise noted, all experiments use:

- Held-out able-bodied test subjects: AB09, AB14, AB24, AB27, and AB23.
- Treadmill speeds: the discrete set of speeds present in the test CSV files, typically spanning 0.5 m/s to 1.85 m/s in increments of 0.05 m/s.
- Ordered speed pairs $(v_{\text{src}}, v_{\text{tgt}})$ with $v_{\text{src}} \neq v_{\text{tgt}}$ and $|v_{\text{tgt}} - v_{\text{src}}| \geq 0.05$ m/s.
- Last 10 seconds (200 time steps at 20 Hz) of each sequence for evaluation, focusing on steady-state walking.

For each test subject and each valid speed pair $(v_{\text{src}}, v_{\text{tgt}})$:

1. The real source-speed sequence $x_{s,v_{\text{src}}}$ is passed to the conditioning encoder together with $(v_{\text{src}}, v_{\text{tgt}})$.
2. A target-speed sequence $\hat{x}_{s,v_{\text{tgt}}}$ is generated by the speed-aware DDIM sampler.

3. The last 10 seconds of $\hat{x}_{s,v_{\text{tgt}}}$ and $x_{s,v_{\text{tgt}}}$ are aligned and resampled as described in the Methods.

4. Per-joint and overall cycle-level errors, and personalization ranks are computed.

All joint-angle errors are computed in radians and converted to degrees for reporting.

4.2 General Accuracy - Able-Bodied

Here is an example of the Able-Bodied generated vs. real data.

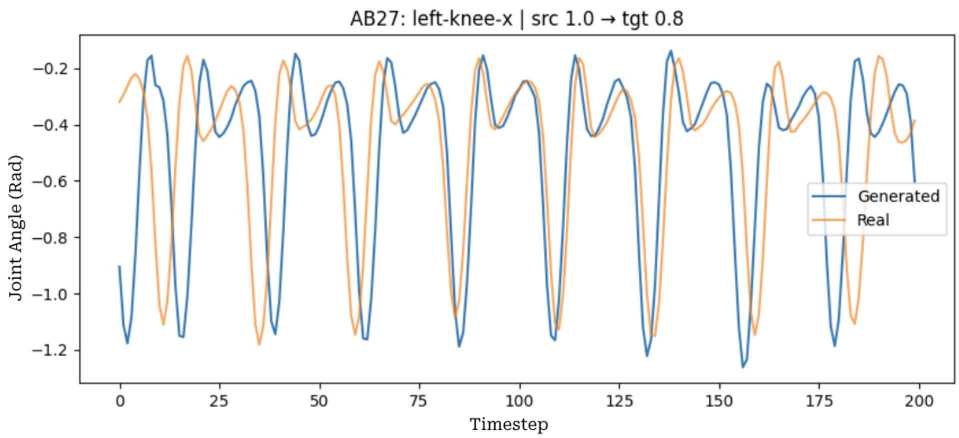


Figure 3:

As shown, smaller errors in each gait cycle generation can propagate and shift the phase of the overall cycle. This is why the multi-cycle approach described earlier in the Methods section was used to calculate the MAE.

An example of this cycle detection is shown here:

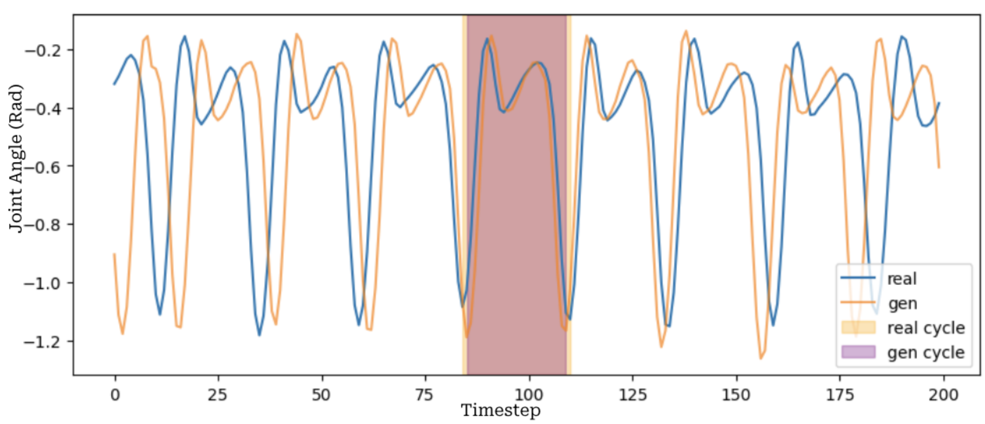


Figure 4:

With this metric and the personalization rank metric, an average across all (subject, source speed, target speed) combinations was taken to produce the results in Table 1

Average Multi-Cycle MAE	4.7°
Average Subject Personalization	2.3

Table 1: Able Bodied Test Metrics

This research is just the first step in a longer pipeline for implementation to the exoskeleton control, so while there are established metrics for evaluating these sequences on downstream exoskeleton tasks in terms of the torque profile, there are no official metrics for evaluating these generated sequences in their current joint angle state. Clinically, a 3-4 ° MAE is aimed for, so our work set a standard of aiming for <5° MAE on average, which can be revisited as necessary in the future.

4.3 Speed Pairings

To further investigate the effects of different (source, target) speed pairings, a heatmap was generated of the average MAE values across subjects for each (source, target) speed pairing.

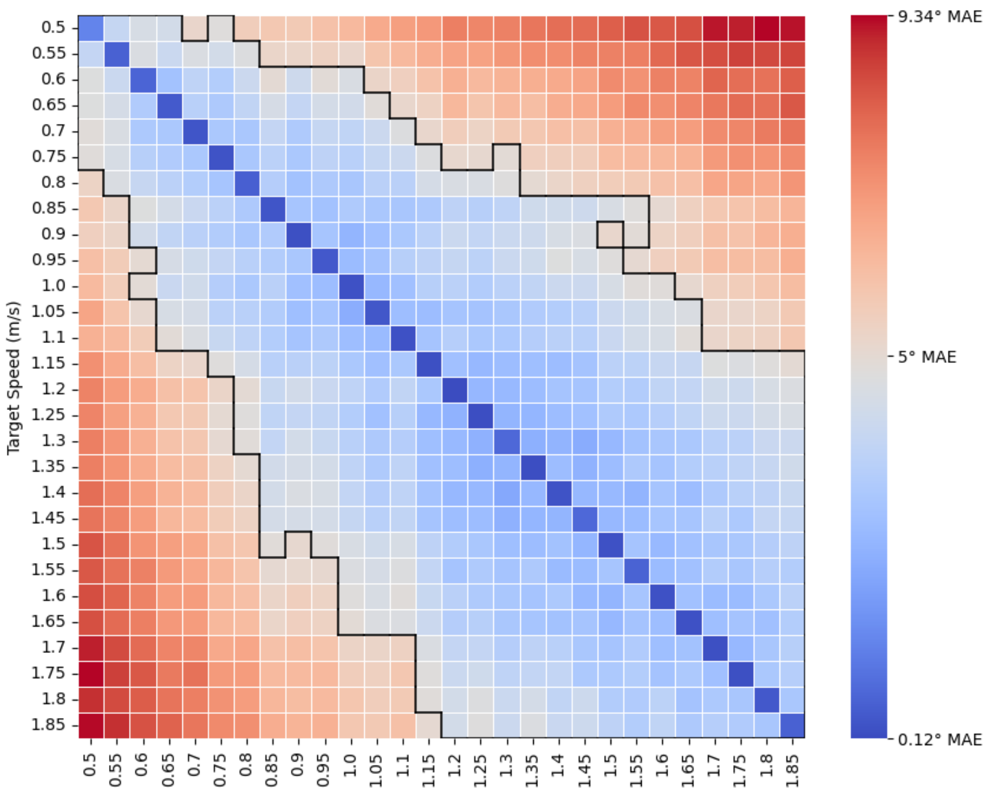


Figure 5:

As shown in figure 5, where the errors less than 5° MAE are outlined, the diffusion model generates sequences better for target speeds closer to the source speed. But, the error is particularly high for extreme speed differences and this is because conceptually, a person's

Average Multi-Cycle MAE	3.2°
Average Subject Personalization	1.4

Table 2: Able Bodied

Average Multi-Cycle MAE	4.96°
Average Subject Personalization	1.5

Table 3: Patient

gait pattern changes at some threshold between 0.5 m/s and 1.85 m/s. An example of this is shown in 6:

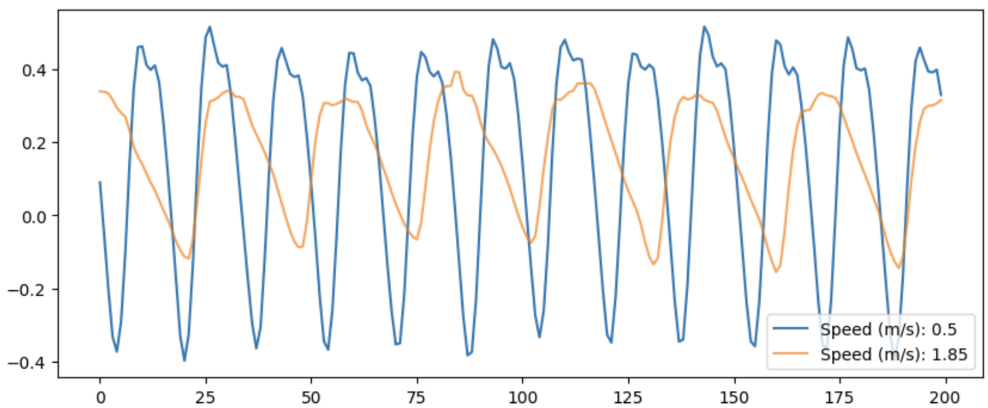


Figure 6:

Figure 7 provides a clearer picture of this with the counts of the number of target speeds with $< 5°$ MAE for each source speed.

4.4 Performance Metrics

With the speeds considerations from the previous section in mind, the source speeds were narrowed down to 0.7 and 1.3 as they provided the largest number of target speeds with error $< 5°$ MAE. For speeds in $[0.5, 1.0]$, a source speed of 0.7 m/a was used, and a 1.3 m/s speed was used otherwise. With these considerations, the metrics for able bodied data is shown in Table 2.

4.5 Patient Data

Furthermore, this model was tested on the stiff-knee subjects as shown in Table 3

And, an example of the stiff knee data generation with one of the detected cycles highlighted is shown in figure 8.

This shows that although the stiff knee data is out of distribution for the able-bodied data that the model was trained on, it adapts well to unseen data. This is likely because we condition on the subject gait data through both the conditioning vector and in the initial sampling.

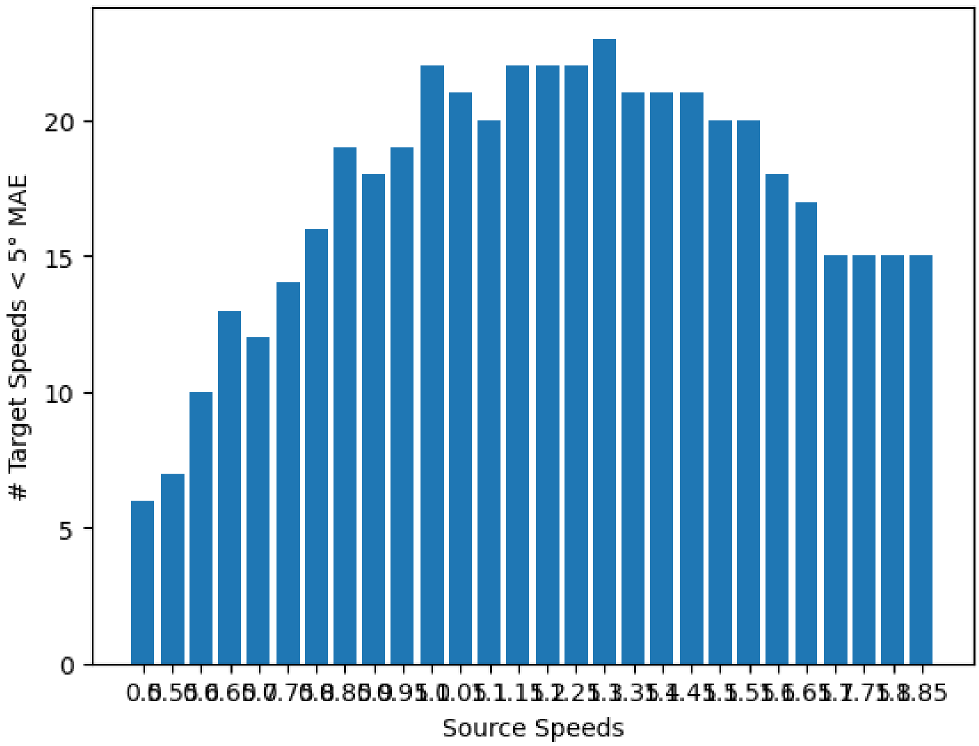


Figure 7:

Furthermore, here is another example of the personalization to stiff knee data. As shown in Figure 9, able-bodied data has a similar gait pattern for both left and right knees. However, for stiff knee data, one knee has gait cycles with a much smaller magnitude that the other knee, which the diffusion model adapts well to.

4.6 Joint Angle Performance

The joint angles average MAEs for each type of subject is shown in figures 10 and 11. As shown, the knee joints have a higher MAE than the remaining joints which is consistent with the findings from prior work.

5 Comparison to Prior Work

5.1 Personalization

To assess how well the diffusion model developed in this research captures subject-specific gait patterns, the personalization metric described in the methods section is evaluated on subject-aware samples and randomized-subject samples.

Definition Consider a test subject s's gait data $x_{s,v_{\text{src}}}$, a target speed v_{tgt}, and a feasible source speed $v_{\text{src}} \neq v_{\text{tgt}}$. The aim is to generate $\hat{x}_{s,v_{\text{tgt}}}$.

For the subject-aware testing, the gait data for subject s ($x_{s,v_{\text{src}}}$), is passed in as well as

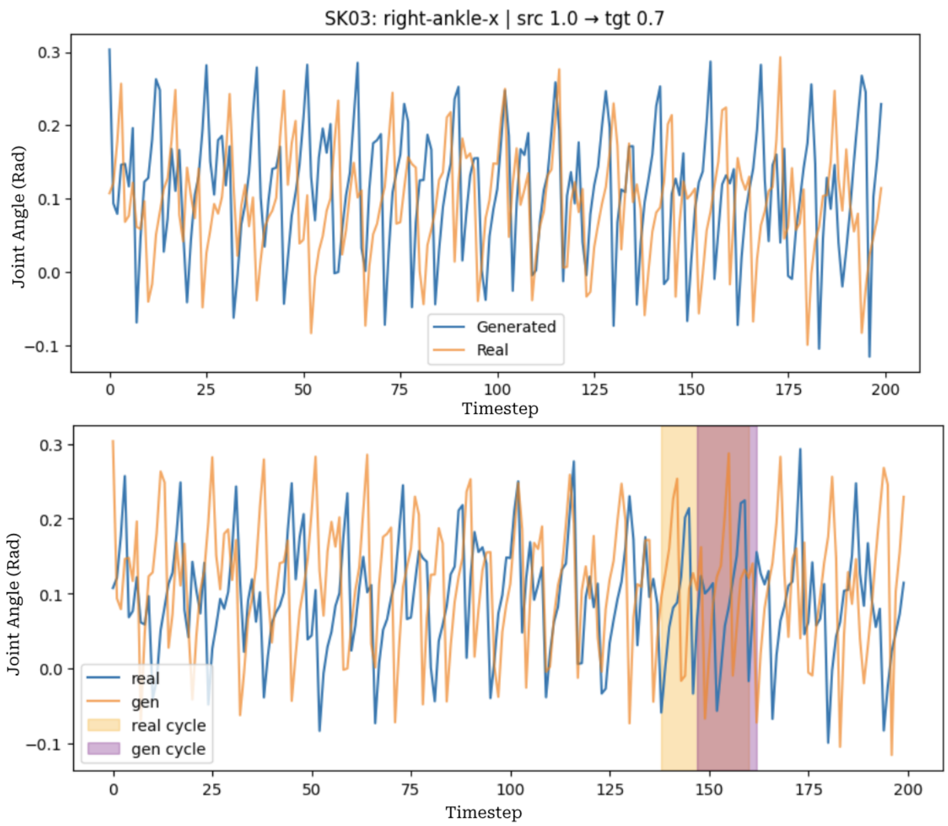
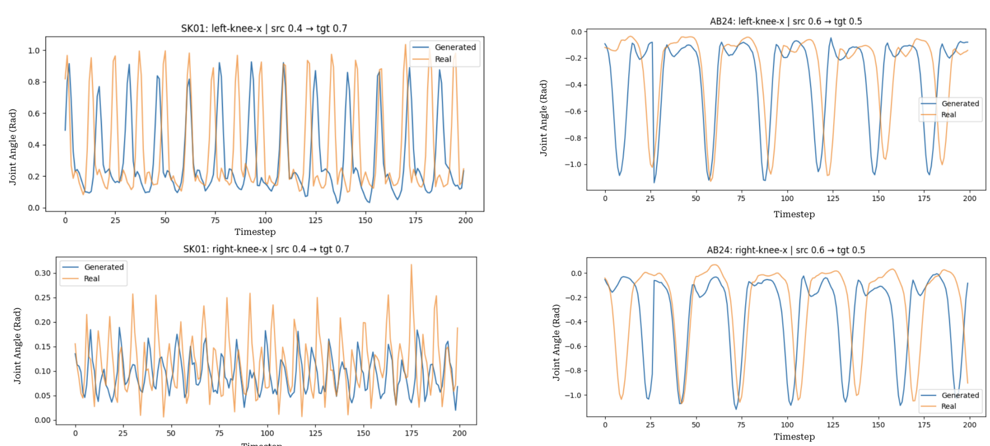


Figure 8:



Patient Example

Able-Bodied Example

Figure 9:

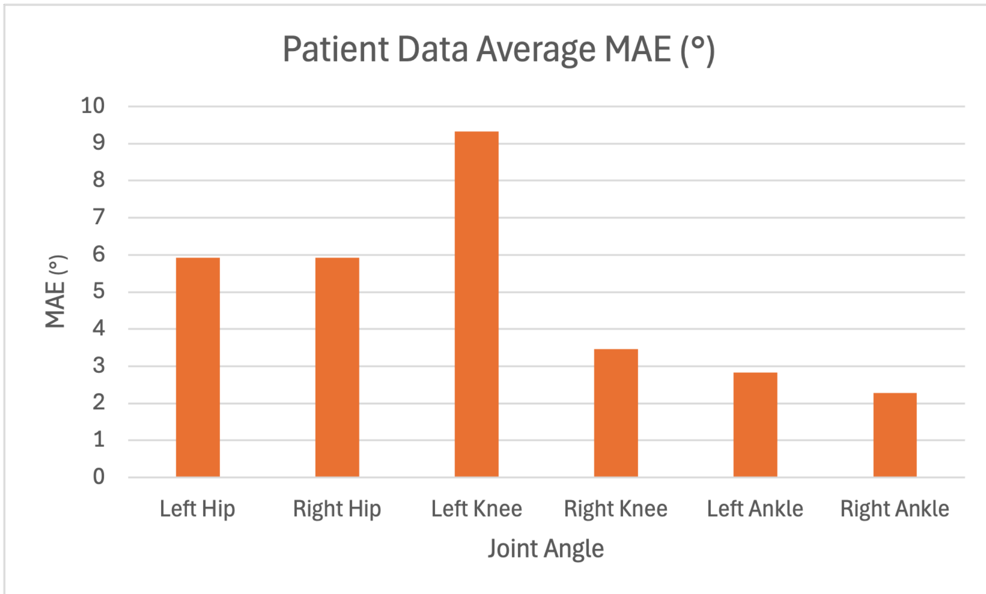


Figure 10:

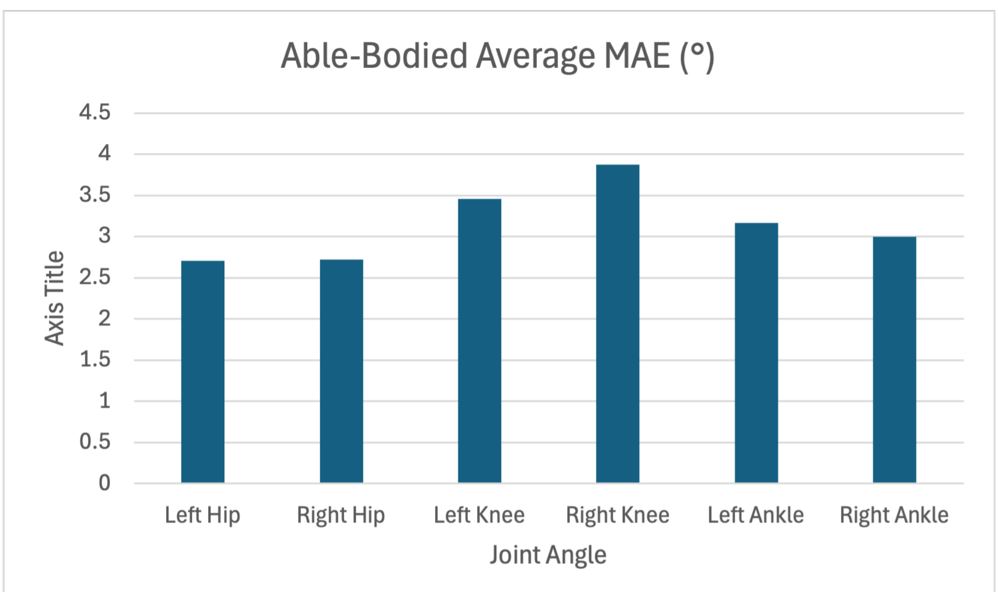


Figure 11:

Δ Average Multi-Cycle MAE °	4.12
Δ Average Subject Personalization	3.9

Table 4: Personalization Ablations

the source and target speeds. The output is then compared to the real data for subject s at the target speed.

For the randomized-subject testing, a randomized subject $s' \neq s$ is chosen and the gait data for that subject at the source speed ($x_{s',v_{\text{src}}}$) is passed in with the source and target speeds. The output is then compared to the real data for subject s at the target speed.

The goal of this is to test if the personalization conditioning added to the GaitDynamics Transformer Encoder actually improves the personalization to each subject of the generated data or if the model is just learning an "average" gait that can be applied across the distribution.

And, the results are shown in figure 12 as well as Table 4. And as shown, the randomized encoding resulted in an increase in average multi-cycle MAE and subject personalization across both able-bodied and stiff-knee subjects, showing that the personalization encoding made an improvement in terms of prediction on both fronts.

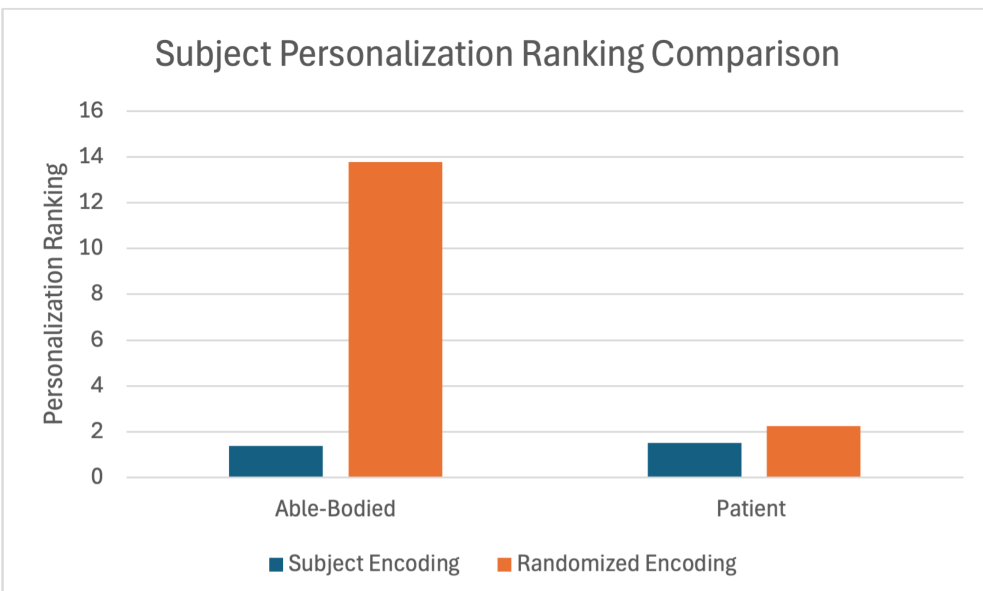


Figure 12:

5.2 Comparison to Linear Scaling

A common technique used besides generative modeling in prior work is linearly scaling the source gait data for a subject to obtain the target data. That is, given a test subject s's gait data $x_{s,v_{\text{src}}}$, a target speed v_{tgt}, and a feasible source speed $v_{\text{src}} \neq v_{\text{tgt}}$. The generated speed is $\hat{x}_{s,v_{\text{tgt}}} = (v_{\text{tgt}}/v_{\text{src}}) * x_{s,v_{\text{src}}}$.

	Diffusion Model	Linear Scaling
Average Multi-Cycle MAE °	3.2	7.1
Average Subject Personalization	1.4	11

Table 5: Comparison with Linear Scaling

Using this method, this error data was collected across all (subject, source speed, target speed) combinations for the Able-Bodied data. This is shown below in Table **??**. And as shown, the MAE did increase for linear scaling, but more significantly, personalization got significantly worse for Linear scaling, showing that the diffusion model captures the subtle changes in gait pattern across speeds and personalizes for subjects better than existing linear scaling pipelines.

6 Future Work

Going forward, there are four main changes this work will extend in.

1. **Extend beyond current conditions**. The current model focuses on only treadmill conditions at level-ground and 6 joint angles. In the future, this model could be extended to other environments, inclines, and extended angle set.

2. **Knee Joint MAE**. As mentioned earlier, the knee joints tend to have a higher MAE compared to the other joint angles, which lines up with what is shown in prior work. However, going forward, the model can be trained with a weighted MSE loss where a weight proportional to the error of each angle is place on the MSE of each angle. This would place more of a focus on the knee joints and ensure a higher accuracy.

3. **Segmented Gait Cycles**. As shown earlier, smaller errors in each gait cycle generation can propagate and lead to an inaccurate read of the error which was why the MAE multi-cycle metric was used. However, going forward, the model can instead consider the input as segmented gait cycles rather than time series data to more accurately predict and evaluate these generated samples.

4. **Stroke Dataset**. The current model has been tested on only stiff-knee replicated data collected at the CMU MetaMobility Lab. Going forward, this model should be extended to more types of patients, starting with testing on a real stroke patients dataset.

References