TEDi: Temporally-Entangled Diffusion for Long-Term Motion Synthesis

Zihan Zhang
University of Chicago
USA
zzhang18@uchicago.edu

Richard Liu
University of Chicago
USA
guanzhi@uchicago.edu

Kfir Aberman
Snap Inc.
USA
kfiraberman@gmail.com

Rana Hanocka
University of Chicago
USA
ranahanocka@uchicago.edu

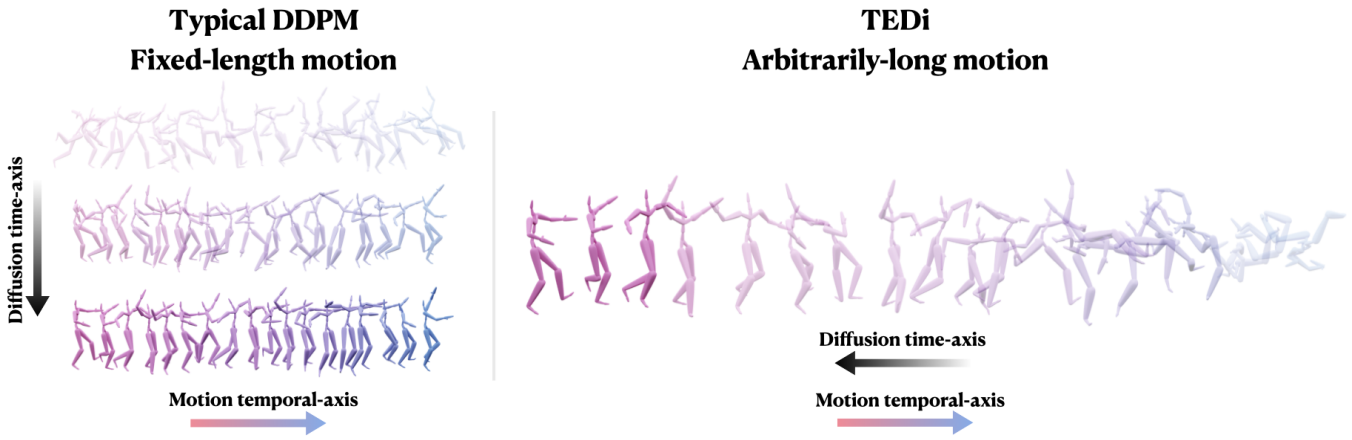

Figure 1: Inspired by the gradual nature of the diffusion process along a diffusion time-axis (left), our approach (right) entangles the temporal-axis of motion with the time-axis of the diffusion process (right), enabling a new mechanism for synthesizing arbitrarily long motion sequences.

ABSTRACT

The gradual nature of a diffusion process that synthesizes samples in small increments constitutes a key ingredient of Denoising Diffusion Probabilistic Models (DDPM), which have presented unprecedented quality in image synthesis and been recently explored in the motion domain. In this work, we propose to adapt the gradual diffusion concept (operating along a diffusion *time-axis*) into the *temporal-axis* of the motion sequence. Our key idea is to extend the DDPM framework to support *temporally varying* denoising, thereby entangling the two axes. Using our special formulation, we iteratively denoise a *motion buffer* that contains a set of increasingly-noised poses, which auto-regressively produces an arbitrarily long stream of frames. With a *stationary* diffusion time-axis, in each diffusion step we increment only the temporal-axis of the motion such that the framework produces a new, clean frame which is removed from the beginning of the buffer, followed by a newly drawn noise vector that is appended to it. This new mechanism paves the way towards a new framework for long-term motion synthesis with applications to character animation and other domains.

CCS CONCEPTS

• **Computing methodologies** → **Motion processing**; *Neural networks.*

KEYWORDS

neural motion processing, motion synthesis, denoising diffusion models

This work is licensed under a Creative Commons Attribution International 4.0 License.

SIGGRAPH Conference Papers '24, July 27–August 01, 2024, Denver, CO, USA

© 2024 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-0525-0/24/07
https://doi.org/10.1145/3641519.3657515

ACM Reference Format:
Zihan Zhang, Richard Liu, Kfir Aberman, and Rana Hanocka. 2024. TEDi: Temporally-Entangled Diffusion for Long-Term Motion Synthesis. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers '24 (SIGGRAPH Conference Papers '24), July 27–August 01, 2024, Denver, CO, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3641519.3657515

1 INTRODUCTION

Long-term generation of a motion sequence is a difficult and long standing problem in character animation with myriad applications in computer animation, motion control, human-computer interaction, and more. Generating long-term motion entails producing realistic, non-repetitive sequences which avoids degenerate outputs (i.e., frozen motion).

A promising avenue for generating high-quality motion is through Denoising Diffusion Probabilistic Models (DDPM), which have produced unprecedented quality in image synthesis [Ho et al. 2020] and have been recently adapted to motion synthesis [Kim et al. 2022; Tevet et al. 2022; Zhang et al. 2022]. A typical adaptation of DDPM to motion synthesis generates a fixed-length motion sequence (i.e., a "motion image") from randomly sampled Gaussian noise.

A fixed-length output limits long-term motion synthesis for a couple of reasons. First, there is no satisfactory approach for creating long-sequences from short-sequences outputs. Simply chaining together motions and blending them may create stitching artifacts. Second, a typical diffusion process has limited interactive controllability. Diffusion requires several hundred denoising iterations before producing a short sequence of clean motions.

We are inspired by the time-dependent nature of the diffusion process, where samples are synthesized from pure noise gradually in small time increments along the diffusion *time-axis*. In this work, we propose to adapt diffusion to the *temporal-axis* of the motion. Our method, referred to as TEDi (**T**emporally-**E**ntangled **Di**ffusion), extends the DDPM framework by enabling injection of temporally-varying noise levels during each step of the diffusion process, instead of a Gaussian noise with a fixed, temporally-invariant variance. By *entangling* the *temporal-axis* of the motion sequence with the *time-axis* of the diffusion process, we enable the production of a continuous stream of clean motion frames during each step of the diffusion process.

At the core of our framework lies a *motion buffer*, which encodes noisy future motion frames with varied noise levels. During the training phase, we add temporally varied noise to clean motion sequences, such that each frame has a random level. However, during inference the motion buffer is initialized with a *motion primer* - a sequence of clean motion frames that are being noised with increasing noise levels, such that adjacent frames contains consecutive noise levels. TEDi recursively denoises the increasingly-noised future frames. In order to constantly maintain the progressively-noised motion buffer structure during each denoising step, we *insert* a noisy frame at the end of the motion buffer and remove a single clean frame at the beginning.

This recursive mechanism enables motion sequence frames to be continuously generated, and avoids stitching problems which current motion diffusion models suffer from (see 4.4.1).

During inference, we can guide the generation with specific motions by intervening in the process and persistently injecting clean frames, called *guiding motions*. This injection enables us to control and influence the current set of generated frames to *prepare and plan* for the upcoming motion guides. This strategy causes a premeditated and calculated transition between the current frames and the future guiding motions.

Our network continues to denoise an ever-evolving motion buffer, which contains vague information about the future trajectory of the motion sequence. This formulation opens the door to more direct control, and better planning, of the generated motion via manipulation of the motion buffer. We demonstrate that our framework is capable of producing different types of long motion sequences, and due to its random nature, can provide diverse results even for the same initialization. In addition, we evaluate the model against other long-term generation models. Our experiments show that TEDi is a natural framework for generating long-term motion sequences.

2 RELATED WORK

2.1 Deep Motion Synthesis

Before the advent of modern deep learning architectures, earlier works attempted to model motion and styles of motion with techniques such as restricted Boltzmann machines Taylor and Hinton [2009]. Later on, the seminal set of works by Holden et al. [2016; 2015] applied convolutional neural networks (CNN) to motion data and learned a motion manifold which can then be used to perform motion editing by, for instance, projection onto the motion manifold. At the same time, recurrent neural networks (RNN) have also been applied to achieve expressive modeling of the temporal dynamics of motion and have succeeded in motion prediction tasks [Fragkiadaki et al. 2015; Pavllo et al. 2018]. RNN-based works have also been applied to incorporate controllable generations such as interactive motion generation [Lee et al. 2018] and music-conditioned motion synthesis [Aristidou et al. 2021]. To address collapse in RNN-based models and better leverage the cyclic nature in motion, Holden et al. [2017] proposed a phase-functioned neural network (PFNN) for locomotion generation. This idea was extended to beyond simple human locomotion by localizing phases to each joint by Starke et al. [2020], and it was also used for quadruped motion generation [Zhang et al. 2018]. Additionally, normalizing flows have also been used to model motion dynamics [Henter et al. 2020]. Deep neural networks have been successfully applied to other motion synthesis tasks such as motion retargeting [Aberman et al. 2020a, 2019; Villegas et al. 2018], motion style-transfer [Aberman et al. 2020b; Mason et al. 2022], key-frame based motion generation [Harvey et al. 2020], motion matching [Holden et al. 2020], animation layering [Starke et al. 2021], motion in-betweening [Tang et al. 2022], and motion synthesis from a single sequence [Li et al. 2022].

2.2 Long-Term Motion Synthesis

Deep learning models for long term motion synthesis are mostly based on RNNs as they naturally enable auto-regressive generation and capture the time dependencies between animation frames. In general, RNNs have shown much success in natural language processing (NLP) for generating text [Sutskever et al. 2011], hand written characters [Gregor et al. 2015], and even captioning images [Vinyals et al. 2015]. At the same time, much progress was made to enable RNNs to better model the temporal and the spatial structure of data. For instance, Shi et al. [2015] proposed ConvLSTM which adds convolutional layers into the fully-connected LSTM, and Wang et al. [2017] proposed Spatialtemporal LSTM (ST-LSTM)

units that are able to model spatial and temporal representations simultaneously in one unified recurrent unit.

Zhou et al. [2018] addressed the error accumulation problem in autoregressive motion models by introducing acRNN, a modification to RNNs to include both the network's output and groudtruth as input during training. However, despite the modified training procedure, acRNN still fails to produce very long motions over a diverse motion dataset. One speculation is that acRNN, and RNNs in general, rely on a memory component that is being eroded with time. In contrast, our framework explicitly utilizes frames within our context window which only needs to be the same size temporally as the diffusion time-axis, producing the motion autoregressively in small increments that complies with the successful mechanism of the diffusion process.

In addition to RNN-based methods, VAE-based and phase-based models have also achieved success in long-term motion generation. In particular, VAEs can be used to model the dynamics of locomotion to enable memoryless autoregressive generation [Ling et al. 2020], and phase-based methods are able to leverage the periodic nature of motion to generate long-term motions that can adapt to complex terrains with rich user interactions [Holden et al. 2017; Starke et al. 2022]. Nevertheless, these methods are often only suited to work over a single motion category (locomotion, dance, etc.) as different categories of motion have distinct transition dynamics.

Moreover, we note that long-term motion generation can also be achieved with repeated applications of motion in-betweening and motion matching. Nevertheless, as the required transition length and time increases, motion in-betweening may degrade in quality, and motion matching systems does not model the underlying motion dynamics of the given dataset and is unable to produce novel motions.

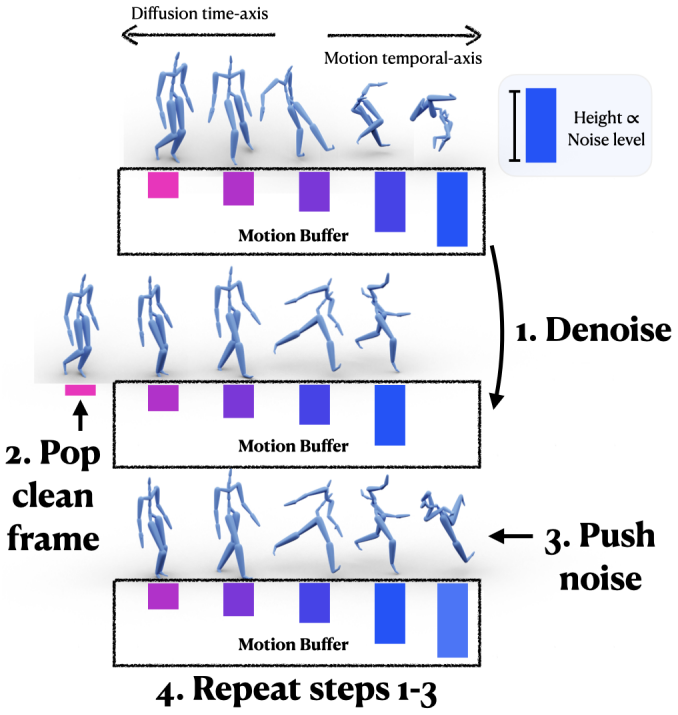

Figure 2: TEDi Recursive Generation. TEDi is capable of generating an arbitrarily long motion sequence. First, we initialize our *motion buffer* with a set of increasingly-noised motion frames. Then (step 1) we denoise the entire motion buffer, (step 2) pop the new, clean frame in the beginning of the motion buffer, and then (step 3) push noise into the end of the motion buffer. This process is repeated recursively.

2.3 Diffusion Models

Denoising diffusion probalistic models (DDPMs) and its variants [Dhariwal and Nichol 2021; Ho et al. 2020, 2022] have achieved unprecedented quality on conditional and unconditional image generation, generally surpassing GAN-based [Dhariwal and Nichol 2021] methods both in visual quality and sampling diversity. In particular, diffusion models have demonstrated remarkable fidelity and semantic control for text-to-image synthesis and editing tasks when large models are trained on text and image pairs [Hertz et al. 2022; Ramesh et al. 2022; Rombach et al. 2021; Ruiz et al. 2022; Saharia et al. 2022b]. In addition, diffusion has been successfully applied in adjacent domains such as text-to-video and image-to-image translation [Saharia et al. 2022a]. Moreover, diffusion models are beginning to see increased usage in generative tasks with 3D data. Some recent work enable 3D data generation by reducing it to a 2D task, while others directly train the entire diffusion pipeline on 3D data. More recently, in the animation domain, Zhang et al. [Zhang et al. 2022], Kim et al. [Kim et al. 2022], Tevet et al. [Tevet et al. 2022], Dabral et al. [Dabral et al. 2023], Yuan et al. [Yuan et al. 2023], and Shimada et al. [Shimada et al. 2024] have suggested adapting diffusion models for motion generation by directly applying the diffusion framework, namely by treating the entire motion as an image and denoising all frames in parallel. This adaptation can only generate fixed-length motion sequences which makes long-term generation and interactive control infeasible. Consequently, long-term generation methods for diffusion based motion synthesis models have been proposed in Shafir et al. [2023] and Tseng et al. [2022]. These methods circumvent the fixed length problem of diffusion models by dividing the long-term motion into fixed length intervals through batched generation and then stitching the results. Temporal consistency of these intervals is enforced through diffusion in-painting to ensure the long-term motion is smooth in the stitched areas. In contrast, our framework requires no stitching or post-processing to obtain long-term motions from the diffusion model, because we combine the diffusion framework with an autoregressive generation scheme, thus enabling generation of arbitrary length sequences by design.

3 METHOD

We propose a new approach to synthesize long motion sequences using diffusion models. Our approach extends the classic DDPM framework to support injection of temporally-varying noise levels during the diffusion process. This extension enables entangling the *temporal-axis* of the motion sequence with the *time-axis* of the diffusion process. In the particular case where the first frame in the sequence is mapped into the lowest noise level, the last frame

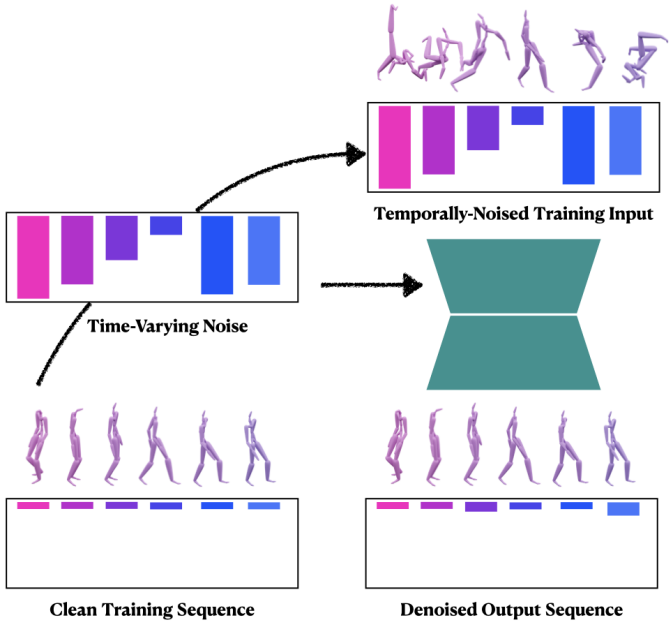

Figure 3: TEDi Training. We train our diffusion-based model to remove temporally-varying noise that is applied to clean sequences during training. In each iteration we fetch a motion sequence of K frames $[f_1, f_2, \ldots, f_K]$ from the dataset, apply noise to it according to a noise level schedule $[\beta_{t_1}, \beta_{t_2}, \ldots, \beta_{t_K}]$, and train our network to predict the clean motion sequence in a supervised fashion as described in (1).

to the highest level, and the mapping function is linear, we can continuously synthesize arbitrarily many frames during inference - akin to a *motion buffer*. In each diffusion step we get a clean frame at the beginning of the sequence, shift the frames in the stack by popping the clean frame, and append a new noisy frame (drawn from a Gaussian distribution) to the end of the sequence. Repeating this process during inference results in a new mechanism for long term motion synthesis. We describe below the motion representation (3.1), novel diffusion framework (3.2), training (3.3) and inference procedure (3.4).

3.1 Motion representation

Motion sequences are represented by a temporal set of K poses consisting of root joint displacements with respect to the xz-plane $\mathbf{O_{xz}} \in \mathbb{R}^{K\times 2}$, root joint height $\mathbf{O_y} \in \mathbb{R}^{K}$, root joint angular velocity with respect to the y-axis $\mathbf{O_w} \in \mathbb{R}^{K}$, joint rotations $\mathbf{R} \in \mathbb{R}^{K\times JQ}$, joint positions in root space $\mathbf{P} \in \mathbb{R}^{K\times 3J}$, and joint velocities in root space $\mathbf{V} \in \mathbb{R}^{K\times 3J}$, where J is the number of joints and Q is the number of rotation features. The rotations are in local coordinates with respect to joint parents, and we use the 6D rotation representation ($Q = 6$) [Zhou et al. 2019]. In addition, we include foot *contact labels* as a $K \cdot C$ binary values $\mathbf{L} \in \{0, 1\}^{K\times C}$, and we let $C = 4$, where the joints are the left(right) heels and toes. All the features are concatenated along the channel axis and we denote the full representation by $\mathbf{M} \equiv [\mathbf{O_w}, \mathbf{O_{xz}}, \mathbf{O_y}, \mathbf{P}, \mathbf{V}, \mathbf{R}, \mathbf{L}] \in \mathbb{R}^{K\times(4+J(6+Q)+C)}$.

3.2 Diffusion Models

Diffusion Denoising Probabilistic Models (DDPM) [Ho et al. 2020; Sohl-Dickstein et al. 2015] are generative models that aim to approximate a given data distribution $q(m_0)$ with an easy and intuitive sampling mechanism that is inspired by diffusion processes in physics. In the particular case of motion synthesis, the data consists of fixed-length motion sequences. During training, the process starts by sampling a clean motion sequence m_0 from the dataset, then an IID Gaussian noise is added gradually to form a sequence of noisy motions which constitute the latent variables of the process $\{m_1, \ldots, m_T\}$. The latent sequence follows $q_{1:t|0}(m_1, \ldots, m_t \mid m_0) = \prod_{i=1}^{t} q_{i|i-1}(m_i \mid m_{i-1})$, where a sampling step in the forward process (clean data to noise) is defined as a Gaussian transition $q_{t|t-1}(m_t \mid m_{t-1}) := \mathcal{N}(\sqrt{1-\beta_t}m_{t-1}, \beta_t I)$ parameterized by a schedule $\beta_0, \ldots, \beta_T \in (0, 1)$. When the total diffusion time step T is large enough, the last noise vector m_T nearly follows an isotropic Gaussian distribution.

In order to sample from the distribution $q(m_0)$, we define the dual "time-reversal" with transitions $p_{t-1|t}(m_{t-1} \mid m_t)$ from isotropic Gaussian noise m_T to data by sampling the posteriors $q_{t-1|t}(m_{t-1} \mid m_t)$. Since the intractable reverse process $q_{t-1|t}(m_{t-1} \mid m_t)$ depends on the unknown data distribution $q(m_0)$, we approximate it with a parameterized Gaussian transition network $p_\theta(m_{t-1} \mid m_t) := \mathcal{N}(m_{t-1} \mid \mu_\theta(m_t, t), \Sigma_\theta(m_t, t))$.

As suggested by [Tevet et al. 2022], instead of predicting the noise as formulated by [Ho et al. 2020], we follow [Ramesh et al. 2022] and the network predicts the signal itself while solving the following optimization problem:

$$\min_\theta L(\theta) := \min_\theta E_{m_0\sim q(m_0), w\sim N(0,I), t} \|m_0 - \mu_\theta(m_t, t)\|_2^2, \quad (1)$$

which maximizes a variational lower bound. In addition, we find that it is best to fix the variance schedule on the reverse process, namely setting $\Sigma_\theta = \beta_t I$ for all time steps, so our model only needs to learn to predict the clean motion. For more details about DDPMs please refer to [Ho et al. 2020; Sohl-Dickstein et al. 2015].

3.3 Temporally-Entangled Diffusion

Next, we extend the DDPM framework to support injection of temporally-varying noise levels during the diffusion process. The noise level becomes a function of the frame index and we discard the notion of the diffusion time-axis during training. Effectively, we are setting $T = K$ and identifying the diffusion time-axis and the motion temporal-axis. We propose two schemes for noise injection: 1) random schedule, and 2) monotonic schedule (we avoid the term linear schedule as it is commonly used to indicate a type of variance schedule [Nichol and Dhariwal 2021]). Note that these are *not* variance schedules. Concretely, given a fixed variance schedule $\beta_{t_i} \in (0, 1), t_i \in \{0, 1, \ldots, T\}$, at each training step the random schedule is given by

$$[\beta_{t_1}, \beta_{t_2}, \ldots, \beta_{t_K}], \; t_i \sim \mathcal{U}(0, T). \quad (2)$$

On the other hand, the monotonic schedule is given by

$$[\beta_{t_1}, \beta_{t_2}, \ldots, \beta_{t_K}], \; t_i = i. \quad (3)$$

The former gives a temporally-varying noise level while the latter gives a monotonically increasing noise level.

In practice, we use a mix of these two noise injection schemes during training, so the model learns to completely denoise a motion sequence with varying noise levels across frames. This enables us to create explicit entanglement between the time axis of the diffusion process and the temporal-axis of the motion - a unique property which will be exploited during inference.

For each iteration during training, we first sample from the dataset a motion sequence of length K,

$$[f_1, f_2, \dots, f_K].$$

The model is then given the noise injected motion $[\tilde{f}_1, \tilde{f}_2, \dots, \tilde{f}_K]$ as input where $\tilde{f}_i \sim \mathcal{N}(\sqrt{\bar{\alpha}(t_i)}f_i, (1-\bar{\alpha}(t_i)I)$, for $\bar{\alpha}(t_i) = \prod_{j=1}^{t_j}(1-\beta_{t_j})$, and is tasked to predict the clean motion $[f_1, f_2, \dots, f_K]$ directly. To give the network a mixture of the two types of noise injection, we assign $[\beta_{t_j}]_{j=1}^{K}$ using the random schedule or monotonic schedule with fixed probabilities p and $1-p$. We set $p = \frac{2}{3}$ in practice.

As a remark, the training objective with the random schedule is similar to those of a pose-oriented diffusion model, where we view the entire motion sequence as a batch of poses with batch size K. If we let q_0^* be the data distribution of individual poses, then at each frame index, the model tries to learn a posterior

$$q^*_{t-1|t}(f^{t-1} \mid f^t)$$

where the superscript indicates time in the diffusion time-axis. Then, the objective with monotonic noise schedule serves to provide additional supervision to ensure smooth transitions across frames during inference.

3.3.1 Loss functions. As previously mentioned, the benefit of predicting the clean motions directly is that it gives access to regularizations that otherwise would be ill-defined for the mollified distributions. For instance, joint velocities cannot be properly regularized with loss terms for noisy motions. Due to the hierarchical nature of the human model, errors accumulate along the kinematic chains, thus errors on joint rotations should be weighted appropriately with respect to their positions in the hierarchy. Therefore, we add a positional loss defined as follows:

$$\mathcal{L}_{\text{pos}} = \frac{1}{KJ}\sum_{t=1}^{K}\left\|\text{FK}_S(\hat{\mathbf{R}}_t, \hat{\mathbf{O}}_t) - \text{FK}_S(\mathbf{R}_t, \mathbf{O}_t)\right\|_2^2, \tag{4}$$

where $\text{FK}_S : \mathbb{R}^{JQ} \times \mathbb{R}^3 \to \mathbb{R}^{3J}$ is a forward kinematics operator for a fixed skeleton S, and $\hat{\mathbf{R}}, \hat{\mathbf{O}}$ are the model predicted joint rotation and displacements and $\mathbf{R}, \mathbf{O}$ are the corresponding ground truth. In addition, since foot contact is vital to generating natural motions and enables using inverse kinematics as post-process, we further penalize errors accumulated at the foot joint with the following foot contact loss:

$$\mathcal{L}_{\text{contact}} = \frac{1}{KC}\sum_{j}\sum_{t=1}^{K-1}\left\|\text{FK}_S(\mathbf{R}_{t+1}, \mathbf{O}_{t+1})_j - \text{FK}_S(\mathbf{R}_t, \mathbf{O}_t)_j\right\|_2^2 \cdot s(\mathbf{L}_{tj}), \tag{5}$$

where $s = \frac{1}{1+e^{-12(x-0.5)}}$, and L_{tj} is the contact label for contact join j at time t as defined in Sec. 3.1. This penalizes high foot velocity while having true contact labels, thus ensuring self-consistency of the generated motions.

3.3.2 Training. In summary, our full training loss is

$$\mathcal{L} = \lambda_{\text{diff}}\mathcal{L}_{\text{diff}} + \lambda_{\text{pos}}\mathcal{L}_{\text{pos}} + \lambda_{\text{contact}}\mathcal{L}_{\text{contact}} \tag{6}$$

where $\mathcal{L}_{\text{diff}}$ corresponds to the diffusion loss specified by equation (1), and the λ parameters determine the weights of the losses.

Our diffusion network is inspired by the typical U-Net model used in the 2D image diffusion domain [Rombach et al. 2022]. In order for the network to process 1D signals, we use 1D convolutions striding over the temporal axis. We also use 1D attention blocks and skip connections so long term frame correlations are captured within the motion data.

3.4 Inference

During inference, we take advantage of the monotonic noise schedule that our model trained on. We use a typewriting-like system, as depicted in Fig. 2. Our model maintains a buffer of frames with monotonically increasing noise, where the first frame in the buffer is mapped to the lowest noise level, and and the last to the highest, as described in (3). The model is designed to generate motion autoregressively. At the beginning, the buffer is initialized with a given motion sequence that is noised with increasing variance. Then, at each iteration, the model processes all the frames in the motion sequence in parallel and produces a progressively denoised sequence. At this point, the first frame in the sequence is completely clean and can be popped from the buffer. We sample a new frame from standard Gaussian distribution and push it into the motion sequence at the end of the buffer. The model can then iteratively perform this denoising mechanism. This mode of generation can be continued indefinitely as desired, and the resulting motion frames are collected frame by frame from the model output.

Concretely, let M_θ be our model and let $I = [\tilde{f}_1, \dots, \tilde{f}_K]$ be the initialization (clean motion that is noised with increasing variance) and let F_{out} denote the (initially empty) set of output frames. At time step t, $t \in \{1, 2, \dots\}$, we have the update

$$\begin{aligned}
F_{\text{out}} &= [F_{\text{out}}, M_\theta(I)_1], \\
\tilde{f}_{i-1} &= M_\theta(I)_i, \ i \in \{2, \dots K\}, \\
\tilde{f}_K &= X \sim \mathcal{N}(0, I), \\
I &= [\tilde{f}_1, \dots, \tilde{f}_K],
\end{aligned}$$

where $M_\theta(I)_i$ denotes the i-th frame in the output of our model.

We highlight the distinction from a typical inference pass in the standard diffusion process, which samples Gaussian noise using the full motion length and repeatedly denoises the entire motion. For such a generation scheme, all the frames are required to pass through the model T times. Here, our inference scheme is able to output a new clean frame after only one forward pass of the model. At the same time, a newly sampled frame (pure noise) that gets pushed into the motion buffer will stay in the motion buffer for T iterations, going through all diffusion time steps before getting added to the output. In short, our inference method enables faster autoregressive generation yet ensures that each frame of motion goes through the full diffusion process.

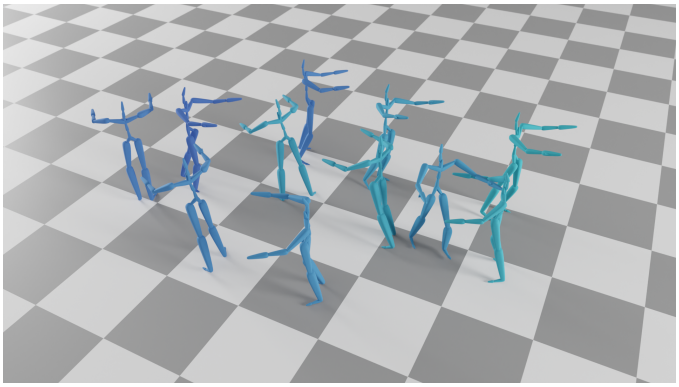

Figure 4: Long-term Generation. Our method synthesizes arbitrarily long motion sequences. In the above figure, we summarize 33 seconds of motion by visualizing the pose every 100-frames (≈3 seconds). Our model is able to generate plausible motions throughout the entire motion sequence.

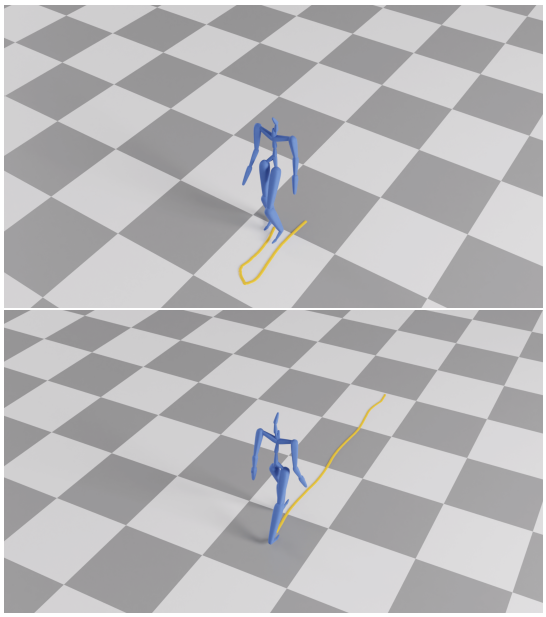

Figure 5: Trajectory Control. Similar to guided generation, given the desired trajectory information P (shown in yellow), our method can generate natural motions that adhere to the given trajectory.

4 EXPERIMENTS

In this section, we demonstrate the effectiveness of TEDi on several long-term generation tasks. We show several unique applications of our method, including the ability to plan for upcoming motion using *guided generation*. We also evaluate our method through various comparisons and ablations. For additional qualitative results, please refer to the supplemental video.

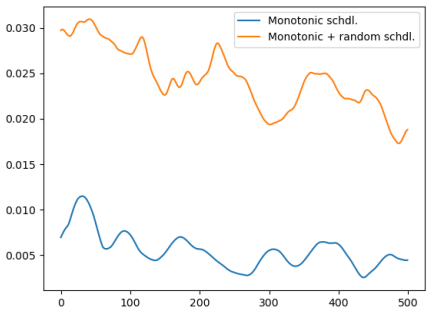

Figure 6: Ablations: Here we show the average motion variance over 500 frames for our method with and without the random schedule. It can be seen that our random schedule helps avoid motion-collapse.

4.1 Long-term Generation

Our TEDi framework is able to generate long-term motions conditioned on a clean primer motion which is used to populate the initial motion buffer. The model is given as input a primer of K frames $\{\tilde{f}_1, \tilde{f}_2, \ldots, \tilde{f}_K\}$ which are progressively noised with a monotonic noise schedule. Our iterative inference strategy can then produce an arbitrarily long sequence of new frames. We highlight some of the frames from a long-term sequence generated by our method in Fig. 4. The key to maintaining long-term generation is that at each iteration, the newly sampled noise frame ensures that our "buffer" is able to explore new potential motions in the near future, and the iterative denoising process ensures framewise consistency across the motion. In addition, we show in Fig. 7 and 8 that our method is capable of generating diverse motion sequences. Full video results can be found in the supplementary video.

4.2 Guided generation

For a character in motion, it is often desired for the character to perform a set of predefined motions which will occur at a point and time in the future. We refer to these frames are *motion guides*. Our framework maintains a motion buffer which contains information about the motions to be performed in the future. In order to *influence* the set of currently-generated frames, we directly modify the motion buffer using the motion guide. Specifically, we remove the current set of frames and replace them with a noised version of the motion guide. Then, we discard the predicted denoised frames and replace them with the noised version of the motion guide at the appropriate diffusion time.

Suppose we have a motion buffer of K frames $I = [\tilde{f}_1, \ldots, \tilde{f}_K]$ and a set of motion guides $\mathbf{Q}_1, \mathbf{Q}_2, \ldots$ each with length $l_1, l_2, \ldots$ frames that we wish to perform starting at frame number $n_1, n_2, \ldots$, $n_i \geq K$. Assuming we start with the frame number $n = 1$ (e.g. the end of the current motion buffer would be frame number K), for each predefined motion $\mathbf{Q}_i$, if any of its frames $\mathbf{Q}_{i_j}$, where $j \in \{1, 2, \ldots, l_i\}$ and $n_i \leq \mathbf{Q}_{i_j} \leq n_i + l_i$ is such that $n + 5 \leq n_i + j \leq n + K$, then we recursively replace it into the motion buffer. Note that there are five frames right before the start of the motion buffer

where we don't recursively replace. This enables the network to smooth out the transitions between the generated frames and the motion guide. We have detailed the procedure for guided generation through recursive replacement in Algorithm 1. We demonstrate guided generation in Fig. 9 and the supplementary material.

Algorithm 1 Guided generation

Require:
M_θ: Denoising model
$I = [\tilde{f}_1, \ldots, \tilde{f}_K]$: motion buffer
$\{\mathbf{Q}_1, \mathbf{Q}_2, \ldots\}$: motion guides
$\{l_1, l_2, \ldots\}$: motion guide lengths
$\{n_1, n_2, \ldots\}$: starting frame numbers for guidance
$F_{\text{out}} = \emptyset$: ouput frames
for n in $1, 2, \ldots$ **do**
  Evaluate $M_\theta(I)$
  for all frames Q_{i_j} **do**
    if $n + 5 \leq n_i + j \leq n + K$ **then**
      $M_\theta(I)_{n_i+j-n} \leftarrow Q_{i_j}$
    end if
  end for
  $F_{\text{out}} \leftarrow [F_{\text{out}}, M_\theta(I)_1]$
  $\tilde{f}_{i-1} \leftarrow M_\theta(I)_i \; \forall i \in \{2, \ldots K\}$
  $\tilde{f}_K \leftarrow X \sim \mathcal{N}(0, I)$
  $I \leftarrow [\tilde{f}_1, \ldots, \tilde{f}_K]$
end for

4.3 Trajectory Control

Our work can be applied to perform trajectory control during inference without additional training. Similar to the mechanism of guided generation in the previous section, trajectory control also utilizes the inpainting strategy by modifying the motion buffer. Specifically, let $I = [\tilde{f}_1, \ldots, \tilde{f}_K]$ be a motion buffer of K frames, and let $\mathbf{P} \in \mathbb{R}^{3\times N}$ be the trajectory information (root displacements with respect to the xz-plane and root height), where N is the desired number of frames to be generated. During inference, we recursively overwrite the trjactory information in the motion buffer with frames in $\mathbf{P}$. The detailed procedure is similar to the one presented in Algorithm 1. We demonstrate trajectory control generation in Fig. 5.

4.4 Comparison and Ablation

We next evaluate our approach against alternative baselines, and assess our framework through an ablation study. We refer the reader to the supplementary video attached to this work to assess the results qualitatively. For quantitative evaluation, we assess our ability to avoid collapses in the motion sequence by measuring the variance across all generated frames. In order to measure how non-stationary generated motions are, and to detect the time-point where they collapse, we measure the average variance of poses in a local window.

4.4.1 Comparison. In this experiment, we focus on comparing our framework to other works on the task of long-term generation. We compare our method with ACRNN [Zhou et al. 2018] and the Human Motion Diffusion Model (MDM) [Tevet et al. 2022]. In particular, ACRNN is an RNN-based work that receives part of the model's output frames during training, to imitate the inference setting and mitigate motion collapse. ACRNN has the potential to be trained over diverse motion datasets for long-term generation and has been used as baseline in other recent works such as GANimator [Li et al. 2022] and GMM [Li et al. 2023]. MDM is an adaptation of the classic DDPM network for motion generation. While ACRNN is designed to be trained has long-term generation as default for inference, MDM does not have a default implementation for long term generation. Thus we use a pretrained checkpoint for MDM and implement an inpainting-based scheme to enable long-term generation for MDM. This implementation is the same as the popular "outpainting" technique used in 2D image generation, where we take the latter part of the generated motion and in-paint it to the first part of the generated motion on the next iteration. As in Fig. 10, it can be seen that ACRNN is not able to perform well on a large and diverse dataset, producing motions that quickly collapse after initialization. In contrast, TEDi can produce infinitely long sequences that is robust to collapses. On the other hand, MDM produces significant stitching artifacts along the in-painting boundary. Please refer to the supplemental video for more details.

4.4.2 Ablation. In Fig. 6, we demonstrate the advantage of our training scheme, by training a version of our model without temporally-variant noise levels. The running variance over a window of 50 frames is calculated for motions generated by the two models. Without temporally varying noise, the network shows sign of mode collapse and diminishes in diversity of long-range motion generation.

4.5 Perceptual Study

We conduct a perceptual study to evaluate the perceived diversity and quality of the generated motions. In addition to MDM and ACRNN (both trained on the same CMU dataset as our model), we also add Motion VAE [Ling et al. 2020], a recent autoregressive motion generation model with VAE, as a baseline comparison. Following the setup of DALLE-2 [Ramesh et al. 2022], we show users 3x3 grids of randomly sampled 500-frame motions from our model (excluding the primer frames), MDM, Motion VAE, and ACRNN, and ask them to choose 1) the set with the most diverse motions and 2) the set with the highest quality motions (only from ours, MDM, or ACRNN). Visual examples of the generated motions in the perceptual study is provided in Fig. 11 and Fig. 12.

We had 55 respondents for our study, and we report the results in Tab. 1. We conclude from our perceptual study that our method produces motion of equivalent or better quality compared to MDM while significantly outperforming in terms of diversity.

Table 1: Perceptual study results for our method and baselines.

	Ours	MDM	ACRNN	MVAE
Diversity	**34**	12	8	1
Quality	**33**	17	5	-

5 CONCLUSION

In this paper, we proposed TEDi, an adaptation of diffusion models for motion synthesis which entangles the motion temporal-axis with the diffusion time-axis. This mechanism enables synthesizing arbitrarily long motion sequences in an autoregressive manner using a U-Net architecture. A unique aspect of our work is the notion of a *stationary* motion buffer. Our framework continues to produce clean frames (i.e., progressing along the diffusion-time axis), without *actually* incrementing the diffusion time. The ability of our pipeline to continually generate motion along the diffusion axis is what enables our framework to robustly and continuously produce novel frames. Interestingly, the ability to naturally use diffusion in such an autoregressive fashion may have implications for other types of sequential data beyond motion, such as audio and video, or modalities where a sequential order can be defined, such as a patch-by-patch order for images.

Our system enables partially-clean-frame to be immediately (or near immediately) popped-off the motion buffer stack. However, a current limitation of our system is that computing a clean from from pure noise requires going through the chain of denoising diffusion. In the future we are interested in leveraging ideas from DDIM [Song et al. 2020] to skip ahead during the denoising process to achieve even lower latency. In addition, our framework may enable future research in long-term text-conditioned motion generation. We are interested in exploring how high-level control may be coupled with low-level user-guidance for the task of long-term generation. Finally, as a result of representing pose with root displacements instead of absolute coordinate (Sec 3.1), our method cannot place the guiding motions in a fixed-world location. One possibility is to specify a trajectory (a set of displacements) that leads to a global position, but this will limit the diversity of different possible motions that can be generated towards the target position.

ACKNOWLEDGMENTS

We thank the 3DL lab for their invaluable feedback and support. This work was supported in part through Uchicago's AI Cluster resources, services, and staff expertise. This work was also partially supported by the NSF under Grant No. 2241303, and gifts from Google and Snap Research.

REFERENCES

Kfir Aberman, Peizhuo Li, Dani Lischinski, Olga Sorkine-Hornung, Daniel Cohen-Or, and Baoquan Chen. 2020a. Skeleton-aware networks for deep motion retargeting. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 62–1.

Kfir Aberman, Yijia Weng, Dani Lischinski, Daniel Cohen-Or, and Baoquan Chen. 2020b. Unpaired motion style transfer from video to animation. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 64–1.

Kfir Aberman, Rundi Wu, Dani Lischinski, Baoquan Chen, and Daniel Cohen-Or. 2019. Learning Character-Agnostic Motion for Motion Retargeting in 2D. *ACM Trans. Graph.* 38, 4 (2019), 75.

Andreas Aristidou, Anastasios Yiannakidis, Kfir Aberman, Daniel Cohen-Or, Ariel Shamir, and Yiorgos Chrysanthou. 2021. Rhythm is a Dancer: Music-Driven Motion Synthesis with Global Structure. *arXiv preprint arXiv:2111.12159* (2021).

Rishabh Dabral, Muhammad Hamza Mughal, Vladislav Golyanik, and Christian Theobalt. 2023. MoFusion: A Framework for Denoising-Diffusion-based Motion Synthesis. In *Computer Vision and Pattern Recognition (CVPR)*.

Prafulla Dhariwal and Alexander Nichol. 2021. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems* 34 (2021), 8780–8794.

Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. 2015. Recurrent network models for human dynamics. In *Proceedings of the IEEE International Conference on Computer Vision*. 4346–4354.

Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Rezende, and Daan Wierstra. 2015. Draw: A recurrent neural network for image generation. In *International conference on machine learning*. PMLR, 1462–1471.

Félix G Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher Pal. 2020. Robust motion in-betweening. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 60–1.

Gustav Eje Henter, Simon Alexanderson, and Jonas Beskow. 2020. Moglow: Probabilistic and controllable motion synthesis using normalising flows. *ACM Transactions on Graphics (TOG)* 39, 6 (2020), 1–14.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. 2022. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626* (2022).

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems* 33 (2020), 6840–6851.

Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. 2022. Cascaded Diffusion Models for High Fidelity Image Generation. *J. Mach. Learn. Res.* 23 (2022), 47–1.

Daniel Holden, Oussama Kanoun, Maksym Perepichka, and Tiberiu Popa. 2020. Learned motion matching. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 53–1.

Daniel Holden, Taku Komura, and Jun Saito. 2017. Phase-functioned neural networks for character control. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–13.

Daniel Holden, Jun Saito, and Taku Komura. 2016. A deep learning framework for character motion synthesis and editing. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 1–11.

Daniel Holden, Jun Saito, Taku Komura, and Thomas Joyce. 2015. Learning motion manifolds with convolutional autoencoders. In *SIGGRAPH Asia 2015 Technical Briefs*. 1–4.

Jihoon Kim, Jiseob Kim, and Sungjoon Choi. 2022. Flame: Free-form language-based motion synthesis & editing. *arXiv preprint arXiv:2209.00349* (2022).

Kyungho Lee, Seyoung Lee, and Jehee Lee. 2018. Interactive character animation by learning multi-objective control. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–10.

Peizhuo Li, Kfir Aberman, Zihan Zhang, Rana Hanocka, and Olga Sorkine-Hornung. 2022. Ganimator: Neural motion synthesis from a single sequence. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–12.

Weiyu Li, Xuelin Chen, Peizhuo Li, Olga Sorkine-Hornung, and Baoquan Chen. 2023. Example-based Motion Synthesis via Generative Motion Matching. *ACM Transactions on Graphics (TOG)* (2023).

Hung Yu Ling, Fabio Zinno, George Cheng, and Michiel Van De Panne. 2020. Character controllers using motion VAEs. *ACM Transactions on Graphics* (2020).

Ian Mason, Sebastian Starke, and Taku Komura. 2022. Real-Time Style Modelling of Human Locomotion via Feature-Wise Transformations and Local Motion Phases. *arXiv preprint arXiv:2201.04439* (2022).

Alex Nichol and Prafulla Dhariwal. 2021. Improved Denoising Diffusion Probabilistic Models. (2021). arXiv:2102.09672 [cs.LG]

Dario Pavllo, David Grangier, and Michael Auli. 2018. Quaternet: A quaternion-based recurrent model for human motion. *arXiv preprint arXiv:1805.06485* (2018).

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125* (2022).

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2021. High-Resolution Image Synthesis with Latent Diffusion Models. arXiv:2112.10752 [cs.CV]

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10684–10695.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. 2022. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. *arXiv preprint arXiv:2208.12242* (2022).

Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. 2022a. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*. 1–10.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. 2022b. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. *arXiv preprint arXiv:2205.11487* (2022).

Yonatan Shafir, Guy Tevet, Roy Kapon, and Amit H. Bermano. 2023. Human Motion Diffusion as a Generative Prior. arXiv:2303.01418 [cs.CV]

Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun WOO. 2015. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (Eds.), Vol. 28. Curran Associates, Inc.

Soshi Shimada, Franziska Mueller, Jan Bednarik, Bardia Doosti, Bernd Bickel, Danhang Tang, Vladislav Golyanik, Jonathan Taylor, Christian Theobalt, and Thabo Beeler. 2024. MACS: Mass Conditioned 3D Hand and Object Motion Synthesis. In *International Conference on 3D Vision (3DV)*.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*. PMLR, 2256–2265.

Jiaming Song, Chenlin Meng, and Stefano Ermon. 2020. Denoising Diffusion Implicit Models. In *International Conference on Learning Representations*.

Sebastian Starke, Ian Mason, and Taku Komura. 2022. DeepPhase: Periodic Autoencoders for Learning Motion Phase Manifolds. *ACM Trans. Graph.* 41, 4 (2022). https://doi.org/10.1145/3528223.3530178

Sebastian Starke, Yiwei Zhao, Taku Komura, and Kazi Zaman. 2020. Local motion phases for learning multi-contact character movements. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 54–1.

Sebastian Starke, Yiwei Zhao, Fabio Zinno, and Taku Komura. 2021. Neural animation layering for synthesizing martial arts movements. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–16.

Ilya Sutskever, James Martens, and Geoffrey E Hinton. 2011. Generating text with recurrent neural networks. In *ICML*.

Xiangjun Tang, He Wang, Bo Hu, Xu Gong, Ruifan Yi, Qilong Kou, and Xiaogang Jin. 2022. Real-time controllable motion transition for characters. *ACM Transactions on Graphics* 41, 4 (jul 2022), 1–10. https://doi.org/10.1145/3528223.3530090

Graham W Taylor and Geoffrey E Hinton. 2009. Factored conditional restricted Boltzmann machines for modeling motion style. In *Proceedings of the 26th annual international conference on machine learning*. 1025–1032.

Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H Bermano. 2022. Human motion diffusion model. *arXiv preprint arXiv:2209.14916* (2022).

Jonathan Tseng, Rodrigo Castellon, and C. Karen Liu. 2022. EDGE: Editable Dance Generation From Music. arXiv:2211.10658 [cs.SD]

Ruben Villegas, Jimei Yang, Duygu Ceylan, and Honglak Lee. 2018. Neural kinematic networks for unsupervised motion retargetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8639–8648.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3156–3164.

Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and Philip S Yu. 2017. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. *Advances in neural information processing systems* 30 (2017).

Ye Yuan, Jiaming Song, Umar Iqbal, Arash Vahdat, and Jan Kautz. 2023. PhysDiff: Physics-Guided Human Motion Diffusion Model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

He Zhang, Sebastian Starke, Taku Komura, and Jun Saito. 2018. Mode-adaptive neural networks for quadruped motion control. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–11.

M. Zhang, Z. Cai, L. Pan, F. Hong, X. Guo, L. Yang, and Z. Liu. 2022. MotionDiffuse: Text-Driven Human Motion Generation With Diffusion Model. *IEEE Transactions on Pattern Analysis; Machine Intelligence* 01 (jan 2022), 1–15. https://doi.org/10.1109/TPAMI.2024.3355414

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. 2019. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5745–5753.

Yi Zhou, Zimo Li, Shuangjiu Xiao, Chong He, Zeng Huang, and Hao Li. 2018. Auto-Conditioned Recurrent Networks for Extended Complex Human Motion Synthesis. In *International Conference on Learning Representations*.

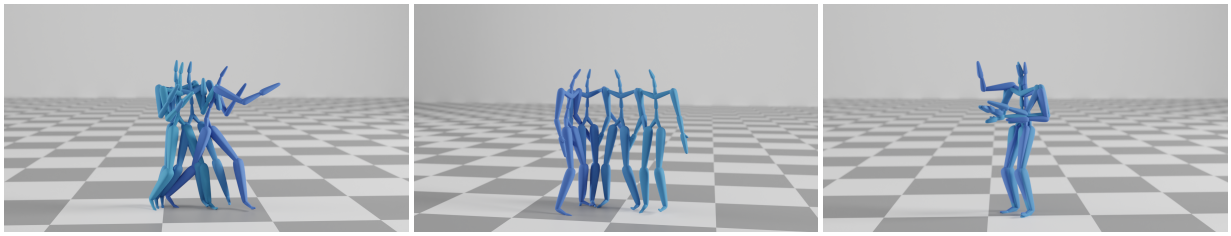

Figure 7: Diverse Motions. Our method is capable of producing a wide variety of long motion sequences. From left to right: Boxing, shuffling, and hand-gestures.

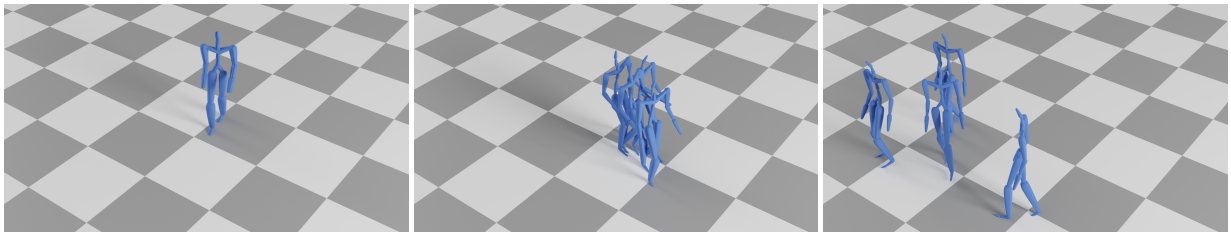

Figure 8: Motion Variations. Due to the stochastic nature of diffusion models, our method is able to generate variations using the same motion primer as input. We show four motions generated from a single primer, from left to right, we can see that the motions begins to differ significantly as time goes on.

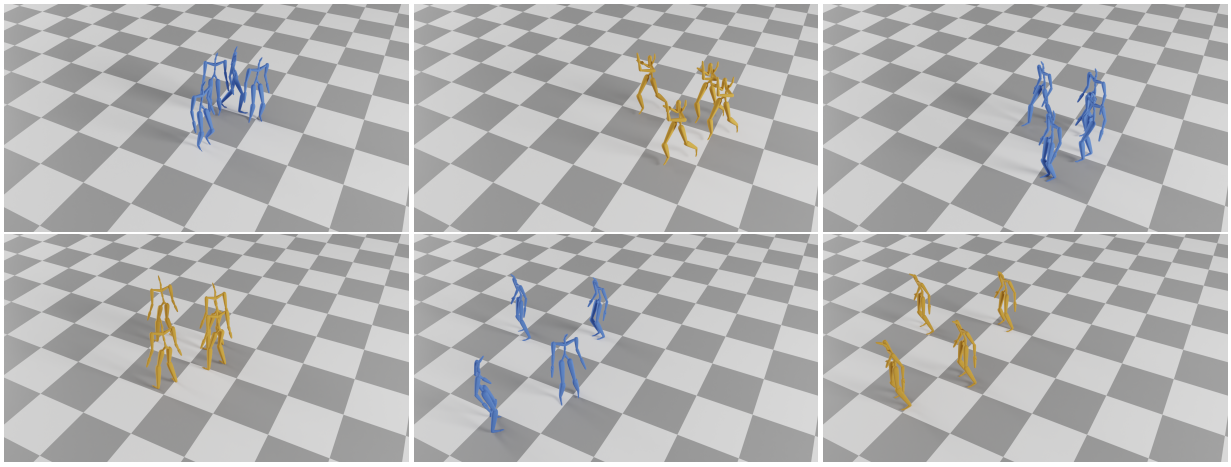

Figure 9: Guided Generation. Given a set of *motion guides* Q_i (shown in yellow), we are able to perform them in sequence at desired points while generating plausible motion in the interactively generated frames (blue). From top-left to bottom-right, our method generates an entire motion sequence that contains the desired motion guides and the interactively synthesized motion. The interactively generated motions will "prepare and plan" for the upcoming motion guides. See the supplementary video.

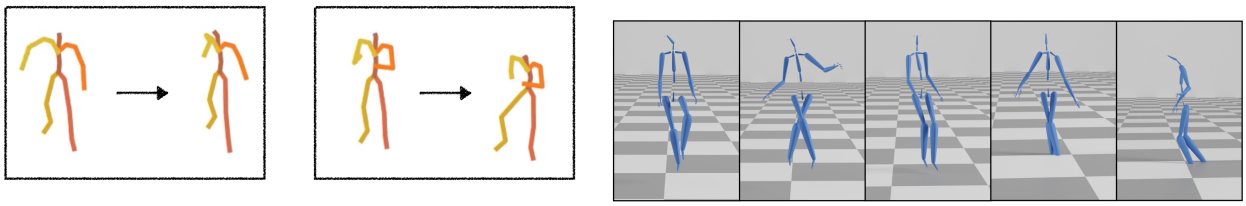

Figure 10: Long-term motion synthesis baseline comparisons. Left: We show two pairs of *consecutive* frames generated through an in-painting implementation with MDM [Tevet et al. 2022]. Classic in-painting shows visible discontinuity that happens along the border of in-painting. Right: ACRNN [Zhou et al. 2018] when trained on a large dataset is not stable, as seen by the foot levitation and penetration artifacts.

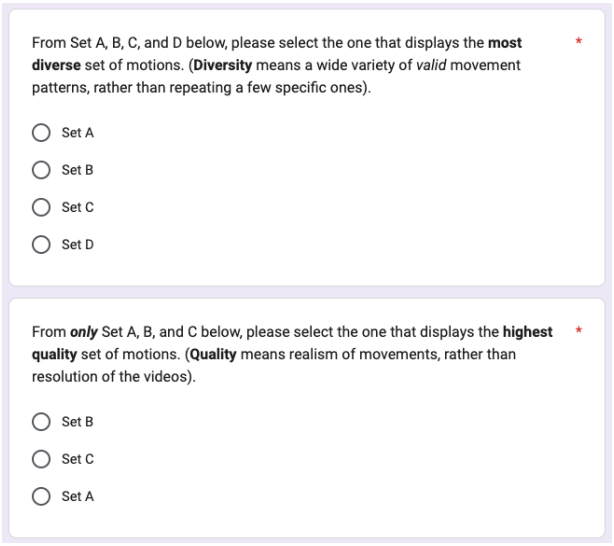

Figure 11: Questions from perceptual study.

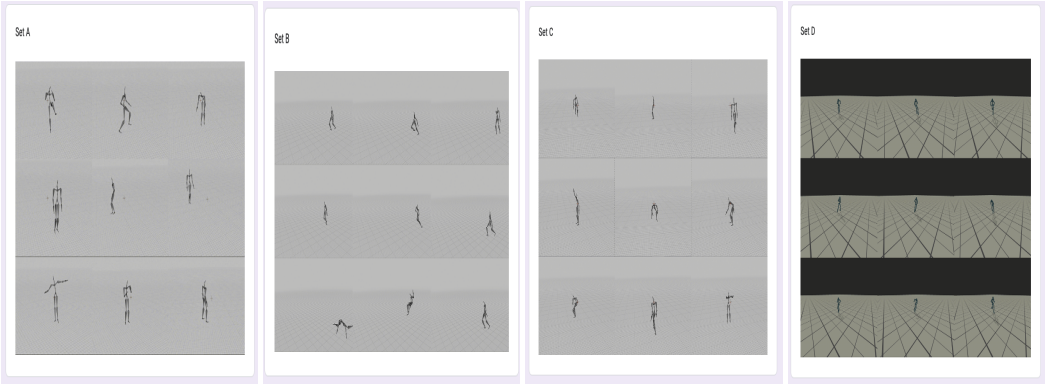

Figure 12: Example motions from perceptual study. *From top to bottom, left to right*: Ours, ACRNN, MDM, and Motion VAE.