Flexible Motion In-betweening with Diffusion Models

Setareh Cohan
University of British Columbia
Canada
setarehc@cs.ubc.ca

Guy Tevet
Tel-Aviv University
Israel
University of British Columbia
Canada
guytevet@mail.tau.ac.il

Daniele Reda
University of British Columbia
Canada
dreda@cs.ubc.ca

Xue Bin Peng
Simon Fraser University
Canada
NVIDIA
Canada
xbpeng@sfu.ca

Michiel van de Panne
University of British Columbia
Canada
van@cs.ubc.ca

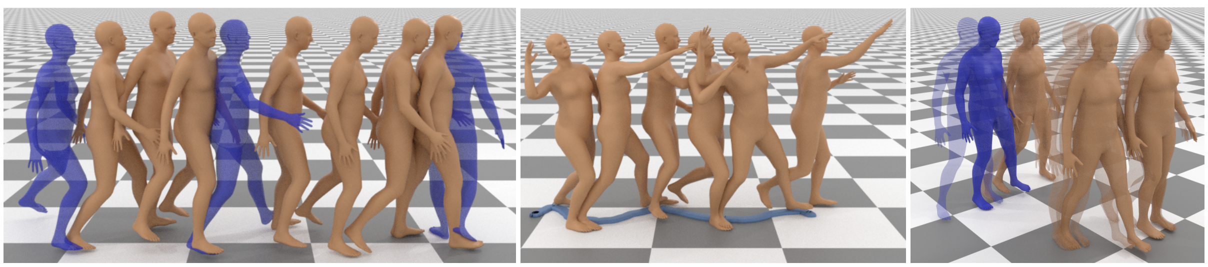


Figure 1: Flexible motion in-betweening given a text prompt and spatio-temporally sparse keyframes. From left to right: a) motion conditioned on sparse keyframes; b) motion conditioned on root trajectory and a "throwing" prompt; c) diverse motions generated for the same keyframes.

ABSTRACT

Motion in-betweening, a fundamental task in character animation, consists of generating motion sequences that plausibly interpolate user-provided keyframe constraints. It has long been recognized as a labor-intensive and challenging process. We investigate the potential of diffusion models in generating diverse human motions guided by keyframes. Unlike previous inbetweening methods, we propose a simple unified model capable of generating precise and diverse motions that conform to a flexible range of user-specified spatial constraints, as well as text conditioning. To this end, we propose Conditional Motion Diffusion In-betweening (CondMDI) which allows for arbitrary dense-or-sparse keyframe placement and partial keyframe constraints while generating high-quality motions that are diverse and coherent with the given keyframes. We evaluate the performance of CondMDI on the text-conditioned HumanML3D dataset and demonstrate the versatility and efficacy of diffusion models for keyframe in-betweening. We further explore the use of guidance and imputation-based approaches for inference-time keyframing and compare CondMDI against these methods.

SIGGRAPH Conference Papers '24, July 27-August 1, 2024, Denver, CO, USA

ACM ISBN 979-8-4007-0525-0/24/07…$15.00
https://doi.org/10.1145/3641519.3657414

CCS CONCEPTS

• **Computing methodologies → Machine learning**; **Animation**.

KEYWORDS

motion generation, motion in-betweening, diffusion models

ACM Reference Format:
Setareh Cohan, Guy Tevet, Daniele Reda, Xue Bin Peng, and Michiel van de Panne. 2024. Flexible Motion In-betweening with Diffusion Models. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers '24 (SIGGRAPH Conference Papers '24), July 27-August 1, 2024, Denver, CO, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3641519.3657414

1 INTRODUCTION

Motion synthesis stands as a central challenge in computer animation, where the precise crafting of realistic movements is essential for conveying natural and lifelike behaviors. Keyframe in-betweening is a critical component of this process, but it is well known to be a demanding and time-consuming manual task. Deep learning-based approaches have recently made significant progress

Code and visualizations are available at **`setarehc.github.io/CondMDI/`**.

on motion in-betweening, leveraging the availability of large-scale and high-quality motion capture datasets. Recurrent neural networks (RNNs) have been studied for the task of keyframe completion [Harvey et al. 2020; Holden et al. 2016; Wang et al. 2022; Zhang and van de Panne 2018], however these RNN models can struggle to accurately model long-term dependencies. Generative modeling techniques have also recently been applied to the task of motion in-betweening [He et al. 2022; Li et al. 2021; Zhou et al. 2020], with transformer-based architectures modeling the long-term dependencies for keyframe motion completion [Duan et al. 2021; Oreshkin et al. 2023].

Most recently, diffusion-based models have demonstrated promising capabilities for generating diverse and realistic human motions [Dabral et al. 2023; Tevet et al. 2023; Zhang et al. 2022]. Diffusion models stand out for their ability to seamlessly incorporate constraints into the generation process, enabling precise control over the generated outputs. Notable examples include text-to-image generation using guidance [Nichol et al. 2021], image completion using inpainting [Lugmayr et al. 2022; Saharia et al. 2022a], and offline reinforcement learning with reward guidance [Janner et al. 2022].

While diffusion models excel as robust conditional generation models, offering unique capabilities for inference-time conditioning, integrating spatial constraints, such as keyframes, into the motion generation process still has no standard solution. In this work, we present a unified and flexible method for motion in-betweening based on a masked conditional diffusion model called Conditional Motion Diffusion In-betweening (CondMDI). This method trains on randomly sampled keyframes with randomly sampled joints, together with a mask that indicates the observed keyframes and features. This then offers significant flexibility in terms of number of keyframes and their placement in time, as well as partial keyframes, i.e., providing information for a subset of the joints.

Our key contribution is a simple and unified diffusion model for motion in-betweening, offering flexible inference-time conditioning. This model is trained by sampling from the space of all possible motion in-betweening scenarios. Our model accommodates temporally-sparse keyframes and partial pose specifications, alongside text prompts. This enables generation of high-quality motion sequences aligned with the specified constraints, while maintaining fast inference speed compared to alternative diffusion-based methods. We additionally provide experimental insights into alternative design choices, including imputation and reconstruction guidance methods.

2 RELATED WORK

Kinematic methods for character animation have a long history. In the following, we first review longstanding data-driven methods, followed by more recent deep-learning based methods, and finally methods focusing specifically on motion in-betweening.

Since the advent of motion capture, numerous methods animate human movement by temporally stitching together captured motion clips to meet user requirements. Motion graphs can precompute feasible motion transitions [Arikan and Forsyth 2002; Kovar et al. 2002; Lee et al. 2002], which can then be used to synthesize motions via search [Kovar et al. 2002; Lee et al. 2002], dynamic programming [Arikan et al. 2003; Hsu et al. 2004; Pullen and Bregler 2002], path planning [Safonova and Hodgins 2007], and reinforcement learning [Lee and Lee 2004; Lo and Zwicker 2008; McCann and Pollard 2007]. Motion matching [Büttner and Clavet 2015] is a related method that searches for animation frames that best fit the current context. Motion blending methods further allow for interpolation of motions. Radial basis function (RBF) kernels have been used to interpolate motions of the same class [Rose et al. 1998; Rose III et al. 2001]. Some work cluster similar motions [Beaudoin et al. 2008; Kovar and Gleicher 2004] while others develop statistical models that allow the original data to be discarded, e.g., [Chai and Hodgins 2007; Mukai and Kuriyama 2005].

Deep learning methods have proliferated through animation. Human motion synthesis models are typically trained using large collections of motion capture data [Adobe Systems Inc. 2021; Guo et al. 2022; Mahmood et al. 2019a]. A large class of parametric models have been proposed for motion modeling, such as RNNs [Aksan et al. 2019; Fragkiadaki et al. 2015; Ghosh et al. 2017; Li et al. 2017], autoencoders [Guo et al. 2020; Holden et al. 2016, 2015; Li et al. 2021; Ling et al. 2020; Zhang et al. 2023], and GANs [Ahn et al. 2018; Ghosh et al. 2021]. Inspired by the success of Flow-based models for image synthesis [Dinh et al. 2014], auto-regressive normalizing networks for motion sequence modeling have also been proposed [Henter et al. 2020].

More recently, denoising diffusion models have been widely utilized for motion synthesis [Dabral et al. 2023; Kim et al. 2022; Tevet et al. 2023; Zhang et al. 2022]. Diffusion-based methods have proved to have a high capacity for modeling the complex distributions associated with motion data and have enabled new types of control over the motion generation. Notable instances are trajectory and joint control by PriorMDM [Shafir et al. 2023], GMD [Karunratanakul et al. 2023b] and OmniControl [Xie et al. 2023]; Multi-person interactions by ComMDM [Shafir et al. 2023], and InterGen [Liang et al. 2023]. The flexibility of diffusion models was also demonstrated for non-human motion synthesis in MAS [Kapon et al. 2023] and SinMDM [Raab et al. 2023].

Motion in-betweening generates a full motion sequence given a set of keyframes with their associated timing. Motion in-betweening can be cast as a motion planning problem, capable of synthesizing fairly complex motions [Arikan and Forsyth 2002; Beaudoin et al. 2008; Levine et al. 2012; Safonova and Hodgins 2007]. Effective data structures such as motion graphs made search and optimization more efficient [Kovar et al. 2002; Min and Chai 2012; Shen et al. 2017]. These methods suffer from memory and scalability issues as they either require maintaining a motion database in memory or performing search and optimization at run-time [Harvey et al. 2020]. Deep learning can overcome these limitations by utilizing large datasets for training while having a fixed computation budget at run-time [Harvey et al. 2020]. Due to the temporal nature of the task, RNN-based methods have dominated the field [Harvey and Pal 2018; Harvey et al. 2020; Zhang and van de Panne 2018]. RNN-based models can struggle with long-term dependencies and are thus often limited to generating shorter transition animations. Unlike auto-regressive models, Transformer-based [Vaswani et al. 2017] models predict the entire motion trajectory at once [Duan et al. 2021; Oreshkin et al. 2023; Qin et al. 2022]. VAEs and GANs have also been applied to motion in-betweening [He et al. 2022; Li et al. 2021; Zhou et al. 2020]. A key limitation of these methods is the models are generally limited to fixed keyframe patterns.

Diffusion-based methods allow for keyframe-based control, e.g., via imputation and inpainting methods. However when methods such as MDM [Tevet et al. 2023] are presented with inpainted full joint trajectories, the motions exhibit very significant foot sliding and unnatural movements to satisfy the constraints. PriorMDM [Shafir et al. 2023] suggests fine-tuning MDM with the observed trajectory of interest. Both methods do not allow for global or sparse-in-time constraints due to a relative-to-previous-frame representation for global root-joint translation and orientation. GMD [Karunratanakul et al. 2023b] supports sparse-in-time keyframes, but only allows for specification of the pelvis position alone rather than the full pose. Hence, sparse keyframes in this work refer to sparse positions of the root joint and their method solves a goal-reaching task rather than keyframe in-betweening. GMD proposes a two-stage pipeline: root trajectory synthesis, then full-body motion generation conditioned on the synthesized root trajectory. It relies on inference-time imputation and guidance and a specialized emphasis-projection technique to increase the importance of observed keyframes.

Closest to our own work, OmniControl [Xie et al. 2023] introduces controllable motion generation with a full-pose spatial conditioning signal, representing global positions of joints over time. While intended for joint control rather than keyframe in-betweening, it supports multiple-joint keyframes and allows for full keyframe conditioning via 3D joint positions, but not joint rotations. OmniControl uses MDM as its diffusion backbone and utilizes a trainable copy of the Transformer encoder of MDM to embed the keyframe signal and later adds them to the attention layers of MDM. In addition to requiring this separate embedding module, OmniControl relies on repeated guidance application to further enforce the constraints. In addition to adding complexity. These features notably increase the inference time of OmniControl compared to other diffusion-based motion generation models.

3 BACKGROUND

In this section, we first review diffusion probabilistic models for motion generation which we refer to as motion diffusion models. Next, we provide an overview of different conditioning approaches applicable to diffusion models for conditional motion generation.

3.1 Human Motion Generation with Diffusion Models

Diffusion models have shown incredible capabilities as generative models [Ho et al. 2020; Sohl-Dickstein et al. 2015; Song and Ermon 2019] and they are the backbone of current state-of-the-art (SOTA) image synthesis models, such as Imagen [Saharia et al. 2022b] and DALL-E2 [Ramesh et al. 2022]. Viewing motion synthesis as a sequence generation problem, diffusion probabilistic models have been recently applied to generate the entire motion sequence at one go [Tevet et al. 2023].

Given a motion dataset, diffusion models add small amounts of Gaussian noise to the samples $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ in T steps such that the marginal distribution at diffusion step T is $q(\mathbf{x}_T) \approx \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \boldsymbol{I})$. This is known as the *forward process* and is formulated as:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}) \tag{1}$$

where t is the diffusion step and $\beta_{1,\dots,T}$ is a fixed variance schedule indicating the amount of noise. To generate samples conditioned on text prompts $\mathbf{p}$, diffusion models learn the *reverse process* of removing noise from $\mathbf{x}_t$ starting from pure Gaussian noise $\mathbf{x}_T$:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{p}) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t, \mathbf{p}), \Sigma_t) \tag{2}$$

where θ are the model parameters, and Σ_t is the untrained time-dependable covariance set according to the variance schedule.

Most motion diffusion models use the sample-estimation reparameterization and directly predict the clean sample estimate $\hat{\mathbf{x}}_0$ instead of the mean estimate $\boldsymbol{\mu}$. In this case the final objective to optimize the diffusion model $G_\theta(\mathbf{x}_t, t, \mathbf{p})$ is:

$$\mathcal{L} := \mathbb{E}_{(\mathbf{x}_0,\mathbf{p})\sim q(\mathbf{x}_0,\mathbf{p}), t\sim[1,T]} \left[\|\mathbf{x}_0 - G_\theta(\mathbf{x}_t, t, \mathbf{p})\|^2 \right]. \tag{3}$$

Given the sample estimate $\hat{\mathbf{x}}_0$, the mean estimate $\tilde{\boldsymbol{\mu}}$ is computed as:

$$\tilde{\boldsymbol{\mu}}_\theta(\mathbf{x}_t, t, \mathbf{p}, \hat{\mathbf{x}}_0) := \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\hat{\mathbf{x}}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t \tag{4}$$

with $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_t$.

To allow for some flexibility over the relative strength of the condition, *classifier-free guidance* is typically used with text-conditioned motion generation. Classifier-free guidance proposes to train an unconditional model $G_\theta(\mathbf{x}_t, t)$ jointly with $G_\theta(\mathbf{x}_t, t, \mathbf{p})$ by setting $\mathbf{p} = \emptyset$ for a fraction of training samples, e.g., 10%. The weighted combination of the two predictions is output at inference time:

$$G_\theta(\mathbf{x}_t, t, \mathbf{p}) = G_\theta(\mathbf{x}_t, t, \emptyset) + w\left(G_\theta(\mathbf{x}_t, t, \mathbf{p}) - G_\theta(\mathbf{x}_t, t, \emptyset)\right) \tag{5}$$

where w helps trade-off between fidelity to the text prompt and diversity among samples.

3.2 Conditional Motion Generation with Diffusion Models

Incorporating spatial constraints such as keyframes into motion diffusion models can be done through two distinct approaches: 1) Training a diffusion model explicitly trained given the spatial conditioning signal as input, and 2) Leveraging a pre-trained motion diffusion model with inference-time imputation and guidance.

Explicit Conditional Models. In this approach, the spatial conditioning signal $\mathbf{c}$ will be used as an additional input to the motion diffusion model $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{p}, \mathbf{c})$. The model is trained to learn this conditional distribution.

Inference-time Imputation. Diffusion models allow for manipulating the generated samples to satisfy certain conditions at inference time. If the spatial conditioning signal $\mathbf{c}$ is an observed part of the desired motion sample (e.g. partial keyframes), *imputation* or *inpainting* [Lugmayr et al. 2022] can be used to generate samples that adhere to this observation. This is done by replacing the output of the pre-trained diffusion model $\mathbf{x}_t$ with the noisy version of the observation $\mathbf{c}$ over the observation mask $\mathbf{m}$ at each diffusion step t.

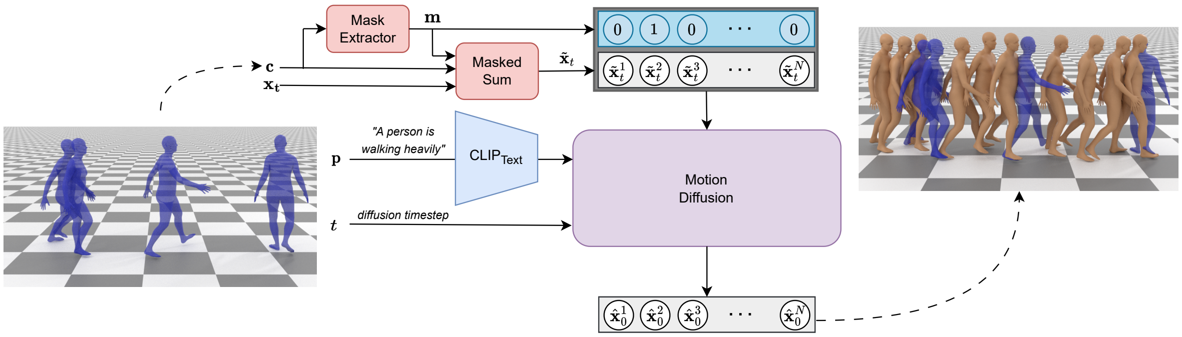


Figure 2: Conditional Motion Diffusion In-betweening (CondMDI) overview. The model is fed a noisy motion sequence $\mathbf{x}_t$, the diffusion step t, a text prompt p, and a keyframe control signal c. Text prompt p is first fed into a CLIP-based [Radford et al. 2021] textual embedder before being fed into the motion diffusion model which is based on GMD [Karunratanakul et al. 2023b]. Mask Extractor module extracts the binary mask and the Masked Sum module performs the masked addition $\tilde{\mathbf{x}}_t = \mathbf{m} \odot \mathbf{c} + (1 - \mathbf{m}) \odot \mathbf{x}_t$ and the gray box around $\tilde{\mathbf{x}}_t$ and m indicates concatenation of the two.

Inference-time Guidance. In addition, *guidance* can also be used to push the samples towards the desired spatial condition. Let $\mathcal{J}(\mathbf{x})$ be a loss function defining how much motion sequence $\mathbf{x}$ deviates from $\mathbf{c}$. With guidance, gradients of $\mathcal{J}$ can be used to guide the output of the diffusion model towards minimizing this loss [Dhariwal and Nichol 2021; Janner et al. 2022]. *Reconstruction guidance* [Ho et al. 2022] is a special form of guidance that operates on sample estimates $\hat{\mathbf{x}}_0$ and is used to improve the cohesion between observations and the generated motion. To do so, the loss function $\mathcal{J}$ is defined as the MSE over the observed spatial constraints and the diffusion model's predictions for the observations. At every denoising step t, model's predictions for the unobserved parts will be adjusted as below:

$$\hat{\mathbf{x}}^{\mathsf{p}}_{0,t} = \hat{\mathbf{x}}^{\mathsf{p}}_{0,t} - \frac{w_r\sqrt{\bar{\alpha}_t}}{2}\nabla_{\mathbf{x}^p_t}\left\| c - \hat{\mathbf{x}}^{\mathsf{o}}_{0,t}\right\|^2 \tag{6}$$

where superscripts o and p refer to the observed and predicted parts of the sequence respectively, and w_r is guidance weight.

4 METHOD

When performing motion keyframing, our goal is to produce realistic motions that adhere to a set of spatio-temporarily sparse input keyframes while maintaining coherence between these observed keyframes and the entirety of the generated motion sequence. In this section, we first provide the detailed problem setup. Then we provide a discussion of the motion data representation and how it affects our keyframe in-betweening method CondMDI, followed by a detailed description of CondMDI.

4.1 Problem Definition

Given a text prompt $\mathbf{p}$, observation control signal $\mathbf{c} \in \mathbb{R}^{N\times J\times D}$, our goal is to generate a human motion trajectory $\mathbf{x} = \{\mathbf{x}^i\}_{i=1}^N \in \mathbb{R}^{N\times J\times D}$ where N is the number of frames. The pose of i-th frame $\mathbf{x}^i \in \mathbb{R}^{J\times D}$ is represented by a D-dimensional feature vector for the pose of J joints. For our task of keyframe in-betweening, the control signal $\mathbf{c}$ contains only an observed subset of $k \leq N$ keyframes (temporal sparsity) for a subset of $j \leq J$ joints (spatial sparsity).

4.2 Motion Representation

Common motion representations divide each motion sequence into two parts: *local motion* containing the pose of the skeleton relative to the root at every frame, and *global motion* containing the global translations and rotations of the root joint relative to the previous frame [He et al. 2022; Karunratanakul et al. 2023b]. Referring back to the problem definition above, a small portion out of the D features includes the global orientation of the root with respect to the previous frame, and the rest of the features represent the local pose with respect to the root joint. Since the root joint positions are represented as relative positions with respect to the previous frame, incorporating temporarily sparse spatial constraints such as sparse keyframes, adds an additional challenge to the sparse keyframing problem. Thus, we address this challenge by converting the relative orientation of the root to global coordinates and use this global-root representation for our model. Detailed description of this conversion is available in the supplementary material.

4.3 Conditional Motion Diffusion In-betweening

We model the conditional reverse posterior $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{p}, \mathbf{c})$ with an explicit conditional diffusion model which takes the keyframe conditioning signal $\mathbf{c}$ as input alongside the noisy motion sample $\mathbf{x}_t$ and the text prompt $\mathbf{p}$. An overview of our approach is represented in Figure 2. To incorporate the keyframe information, following [Harvey et al. 2022], we adopt a straightforward approach and replace the noisy sample $\mathbf{x}_t$ with the observed partial keyframes $\mathbf{c}$ at every observed frame and joint. To provide the model with an indication of which features are observed, we concatenate the resulting masked sample $\mathbf{x}_t$ with the observation mask as input to the diffusion model. The observation mask $\mathbf{m} \in \mathbb{R}^{N\times J\times D}$ is a binary mask with ones over the observed frames and joints and zero everywhere else, defined based on the keyframe signal $\mathbf{c}$. To allow for flexible keyframe conditioning at inference-time, our model is trained with randomly sampled partial keyframes. Algorithm 1 shows an overview of the training procedure of our conditional method. `Random Mask Generator` is the procedure in which the

ALGORITHM 1: Training

repeat
- $(\mathbf{x}_0, \mathbf{p}) \sim q(\mathbf{x}_0, \mathbf{p})$
- $\mathbf{m} \sim$ `Random Mask Generator`
- $\mathbf{p} \leftarrow \emptyset$ with probability 10% ▷ `Classifier-free Guidance`
- $\mathbf{c} \leftarrow \emptyset$ with probability 10% ▷ `Unconditioned Generation`
- $t \sim \mathrm{Uniform}(\{1, \ldots, T\})$
- $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
- $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \boldsymbol{\epsilon}\sqrt{1-\bar{\alpha}_t}$
- $\mathbf{x}_t = \mathbf{m} \odot \mathbf{x}_0 + (1 - \mathbf{m}) \odot \mathbf{x}_t$
- $\mathbf{x}_t = \langle \mathbf{x}_t, \mathbf{m} \rangle$
- Take gradient descent step on $\nabla_\theta \left\| \mathbf{x}_0 - G_\theta(\mathbf{x}_t, t, \mathbf{p}) \right\|^2$

until *converged*;

ALGORITHM 2: Sampling

Require: Guidance scale w
Require: Text prompt $\mathbf{p}$
Require: Keyframe signal $\mathbf{c}$ and observation mask $\mathbf{m}$
$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
for $t = T, \ldots, 1$ **do**
- $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
- $\mathbf{x}_t = \mathbf{m} \odot \mathbf{c} + (1 - \mathbf{m}) \odot \mathbf{x}_t$
- $\mathbf{x}_t = \langle \mathbf{x}_t, \mathbf{m} \rangle$
- $\hat{\mathbf{x}}_0 = G_\theta(\mathbf{x}_t, t, \emptyset) + w\left(G_\theta(\mathbf{x}_t, t, \mathbf{p}) - G_\theta(\mathbf{x}_t, t, \emptyset)\right)$
- $\hat{\boldsymbol{\mu}} = \tilde{\boldsymbol{\mu}}(\hat{\mathbf{x}}_0, \mathbf{x}_t)$
- $\mathbf{x}_{t-1} = \hat{\boldsymbol{\mu}} + \sigma_t \mathbf{z}$

end
return $\mathbf{x}_0$

number of keyframes k is first sampled within the length of the motion sequence, and then these k keyframes are randomly picked out of all the frames in the sequence. To provide additional flexibility over the joints, this method is extended to additionally sample the number of observed joints j, and then randomly pick the observed joints out of all J joints. Note that we set keyframe conditioning signal $\mathbf{c}$ to $\emptyset$ for 10% of training samples to make CondMDI better suited for unconditioned motion generation at inference time. Our proposed conditioning method can be applied to any backbone text-conditioned motion diffusion model G_θ, and we choose to use the motion diffusion model of GMD [Karunratanakul et al. 2023a] as our backbone diffusion model. For more details about the network architecture, refer to the supplementary material. $\odot$ is the element-wise product and $\langle\rangle$ are used to denote concatenation. The sampling procedure of our conditional method is available in Algorithm 2.

5 IMPLEMENTATION AND EVALUATION METRICS

Our method is evaluated on the human motion generation task conditioned on text prompts and a variety of keyframe control signals. In particular, we evaluate the performance of our method on text-conditioned motion generation given sparse keyframes. We also compare against inference-time conditioning methods for the task of in-betweening, including both imputation and imputation combined with reconstruction guidance. Finally, we evaluate our model on a wide range of conditioning signals to demonstrate the capabilities of our model beyond simple keyframes.

5.1 Dataset

Our model is evaluated on the HumanML3D [Guo et al. 2022] dataset which contains 14,646 text-annotated human motion sequences taken from the AMASS [Mahmood et al. 2019b] and HumanAct12 [Guo et al. 2020] datasets. Motion sequences from this dataset have variable lengths where the average motion length is 7.1 seconds and motions are padded with zeros to be a fixed length of 196 frames with a framerate of 20 fps. In this dataset, motion at every frame is represented by a 263-dimensional feature vector consisting of the relative root joint translations and rotations, plus the local pose including the joint rotations and joint positions with respect to the root joint. Detailed description of the data representation is available in the supplementary material.

Sparse keyframes need to be defined with global translation and orientation of the root joint. To make conditioning of diffusion models on such global keyframes more straight-forward, we first convert the dataset to have global orientations for the root joint. For each frame, this is simply done by cumulatively summing the translation and rotation of the root joint up to its previous frame. CondMDI assumes similar dimensionality for the keyframe signal and motion signal. Consequently, for each observed frame and joint, CondMDI requires all corresponding features out of 263. Additionally, as motions in the dataset are represented as root motion and pose with respect to the root, partial keyframes always include the root joint. Further details on this can be found in the supplementary material.

5.2 Evaluation Metrics

For the task of conditional motion generation, we adopt the evaluation protocol from Guo et al. [2022]. They suggest a set of neural metrics calculated in a mutual text-motion latent space based on pre-trained encoders. This includes *Fréchet Inception Distance (FID)* score, which measures the distance between the distribution of ground-truth and generated motions in the latent space of a pre-trained motion encoder. *R-Precision* measures the proximity of the motion to the text it was conditioned on, and *Diversity* measures the variability within the generated motion. The full description of these metrics is available in the supplementary material. In addition, we adopt the *Foot Skating Ratio* and the *Keyframe Error* metrics from Karunratanakul et al. [2023b]. The prior measures the proportion of frames in which either foot skids more than a certain distance (2.5 cm) while maintaining contact with the ground (foot height < 5 cm). The latter measures the mean distance between the generated motion root locations and the keyframe root locations at the keyframe motion steps.

5.3 Implementation Details

For our baseline diffusion model, we adopt the motion diffusion model of GMD [Karunratanakul et al. 2023b], which uses a UNet architecture with AdaGN [Dhariwal and Nichol 2021]. Our model uses the sample-estimation parameterization of DDPMs [Ho et al. 2020] with $T = 1000$ diffusion steps during training and inference.

Table 1: Text-to-motion evaluation on the HumanML3D test set.

	FID ↓	R-precision ↑ (Top-3)	Diversity →
Real	0.002	0.797	9.503
JL2P [Ahuja and Morency 2019]	11.02	0.486	7.676
Text2Gesture [Bhattacharya et al. 2021]	7.664	0.345	6.409
T2M [Guo et al. 2022]	1.067	0.740	9.188
MotionDiffuse [Zhang et al. 2022]	0.630	**0.782**	<u>9.410</u>
MDM	0.556	0.608	**9.446**
MLD [Chen et al. 2023]	0.473	<u>0.772</u>	9.724
PhysDiff [Yuan et al. 2023]	0.433	0.631	-
GMD $\mathbf{x}^{\text{proj}}$	**0.235**	0.652	9.726
Ours	<u>0.2538</u>	0.6450	9.7489

Table 2: Quantitative results for different keyframes on the HumanML3D test set. $K \in \{1, 5, 20\}$ means number of keyframes randomly placed along the motion trajectory. *Root Joint* and *VR Joints* mean conditioning on the root joint trajectory and the head and both wrist joints repectively.

Conditioning	FID ↓	R-precision ↑ (Top-3)	Diversity →	Foot skating ratio ↓	Keyframe err ↓
Random K=1	0.1551	0.6787	9.5807	0.0936	0.3739
Random K=5	0.1731	0.6823	9.3053	0.0850	0.1789
Random K=20	0.2253	0.6821	9.1151	0.0806	0.0754
Root Joint	0.2474	0.6752	9.4106	0.0854	0.0525
VR Joints	0.2969	0.6842	9.0659	0.0794	0.0422

Similar to GMD, we use the pre-trained CLIP model to encode the text prompts [Radford et al. 2021]. For more implementation details, refer to the supplementary material.

6 RESULTS

In this section, we present our empirical findings. In Sections 6.1 and 6.2, we provide qualitative samples for sparse-in-time and sparse-in-time-and-joints keyframes. In Section 6.3 we evaluate the performance of CondMDI on the task of text-conditioned motion synthesis without any keyframe conditioning. Section 6.4 contains evaluation results of CondMDI on the text-and-keyframe conditioned motion generation task. Finally, Section 6.5 shows the ablation results. For additional results on sample diversity and text-conditioning, refer to the supplementary material. In the qualitative samples, generated and observed keyframes are shown in yellow and blue unless otherwise stated. Additionally, in all tables, **bold** indicates best result, <u>underline</u> indicates second best, and → indicates that closer to real is better.

6.1 Sparse Keyframe In-betweening

First, we evaluate the performance of our method on the task of sparse keyframe in-betweening, a primary focus of our model. For the classical case of sparse keyframe in-betweening, we first evaluate our model by creating samples using sparse keyframes provided at fixed transitions of T frames. Figure 3 shows that the model is capable of generating high quality motions from sparse keyframes placed every $T = 20$ frames, even on dynamic and complex movements such as karate and yoga. Our qualitative results show that CondMDI can generate smooth and high quality samples that are consistent with the input keyframes, even with spacing of over 40 frames. As a more general sparse keyframing approach, instead of specifying keyframes evenly spaced in time, we provide K frames randomly spaced in time.

6.2 Partial Keyframe In-betweening: Joint Control

To further test the capabilities of our model, we define spatially-sparse keyframes, i.e. keyframes that contain a subset of the joints. Our model demonstrates good performance even when provided with a single joint trajectory. Figure 4 shows examples of the model provided with only the root joint trajectory (projected on the ground in the left figure), or with only the right wrist joint. The sample follows the input trajectory closely with natural and smooth motions. Partial keyframes also allow for other useful applications, such full-body motion reconstruction from sparse VR headsets, consisting of only the head, left wrist and right wrist joints. In the supplementary video, we show that our model is able to generate complex lower-body motions only from this sparse input.

6.3 Unconditioned Synthesis

In Table 1, we demonstrate the performance of CondMDI on the task of text-conditioned motion synthesis. This table is added as a reference to interpret the values of the rest of the quantitative evaluations, in which CondMDI also observes input keyframes. Conditioning the same model on keyframe information, should ideally lead to superior performance, as the space of solutions becomes more restricted. However, in practice, incorporating conditioning signals comes with unique challenges, generally leading to worse performance for conditioned models in terms of motion quality metrics. Therefore, a decrease in the average keyframe error while maintaining or improving the rest of the metrics shows the effectiveness of a model conditioned on keyframes.

6.4 Evaluation

For a grounded evaluation, we test our model by computing quantitative results for a range of keyframing schemes. Results are shown in Table 2. For the three cases with $K \in \{1, 5, 20\}$ randomly placed full keyframes, we can see that as the number of observed keyframes increases, the average error of the keyframes decreases due to denser conditioning, providing a stronger influence on the model. However, increasing keyframes results in worse FID values, possibly because denser signals may constrain the model too much, leading to performance degradation. Overall, all these cases demonstrate performance comparable to or better than unconditional synthesis for the motion quality metrics, while exhibiting only small errors at the keyframes. The last two rows of Table 2 show the results for partial keyframes of root joint trajectory (Root Joint) and VR joints (VR Joints). CondMDI achieves comparable performance on motion generation while keeping the keyframe error minimal.

Direct comparison of CondMDI with SOTA motion diffusion models on the task of keyframe in-betweening is challenging. Some of these models are trained on relative coordinates and thus do not allow for inference-time conditioning on keyframes defined in global coordinates (MDM, PriorMD). GMD is trained with a global coordinate representation, but does not support full keyframes. OmniControl is a recent work that is intended to be used for joint

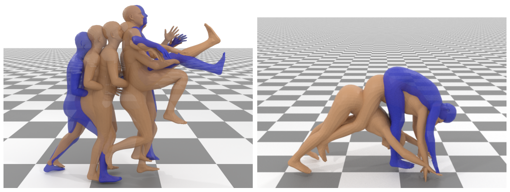


Figure 3: Our model is capable of generating high-quality motions in hard moves such as a karate kick or a yoga sun salutation pose. Check the video for the full motions.

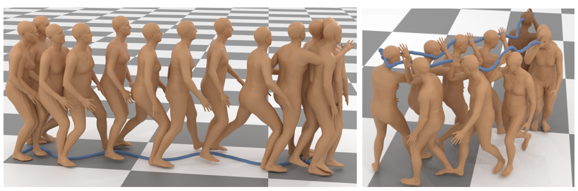


Figure 4: A walking motion conditioned only on the root joint (left) and only on the right wrist (right).

Table 3: Quantitative results for root-joint control on the HumanML3D test set. *OmniControl (on all)* means the model is trained on all joints.

Method	FID ↓	R-precision ↑ (Top-3)	Diversity →	Foot skating ratio ↓	Keyframe err ↓
Real	0.002	0.797	9.503	0.000	0.000
MDM	0.698	0.602	9.197	0.1019	0.5959
PriorMDM	0.475	0.583	9.156	0.0897	0.4417
GMD	0.576	0.665	9.206	0.1009	0.1439
OmniControl (on all)	<u>0.322</u>	**0.691**	**9.545**	**0.0571**	**0.0367**
Ours	**0.2474**	<u>0.6752</u>	<u>9.4106</u>	<u>0.0854</u>	<u>0.0525</u>

control, but according to the authors, allows for full keyframe conditioning as well. For a more complete comparison, we focus on the task of root joint trajectory control and summarize the performance statistics in Table 3. CondMDI demonstrates comparable performance on this task with respect to the SOTA OmniControl model while having a simpler architecture and a relatively faster inference speed. For more details on the inference speed, refer to the supplementary material.

6.5 Ablations

We perform a comprehensive ablation study over different conditioning methods. Table 4 shows the ablations results in which we defined $K = 5$ keyframes randomly spaced over motion sequences. Pure imputation which replaces keyframes with ground-truth values at every denoising step (*IMPC=0*) demonstrates minimal error over keyframes, which is expected when replacement is performed until the last denoising step. However, the very large FID score shows that this method leads to unnatural low-quality motions. Figure 5a shows a sample from this method, which exhibits a large jump before and after every keyframe. This shows that imputation is completely ignored by the diffusion model. Stopping imputation at denoising step 1, (*IMP*) results in high keyframe errors but near-SOTA FID score. Figure 5b shows such an example for which the model completely ignores the input keyframes but generates a reasonable motion. Adding reconstruction guidance to imputation (*IMP+RecG*) improves both the motion quality metrics and the keyframe-related errors compared to (*IMP*). Figure 5c shows a sample in which the motion both adheres well to the keyframes while being coherent with the keyframes and the generated frames, reducing the jumps seen with (*IMP*). Finally, CondMDI exhibits the best performance compared to inference-conditioning methods. Figure 5d shows a smooth motion that adheres closely to the keyframes.

Finally, we perform an ablation study over the choices of random mask generation schemes used during training. In Table 4, *(random frames)* correspond to our model trained with keyframes generated by randomly sampling the number and location of observed keyframes, while always including all the joints. Although this model has comparable FID scores and improved keyframe error compared to CondMDI on this task, it does not generalize well to partial keyframes. In general, CondMDI does better on partial keyframe in-betweening tasks, as the model is trained with partial keyframes.

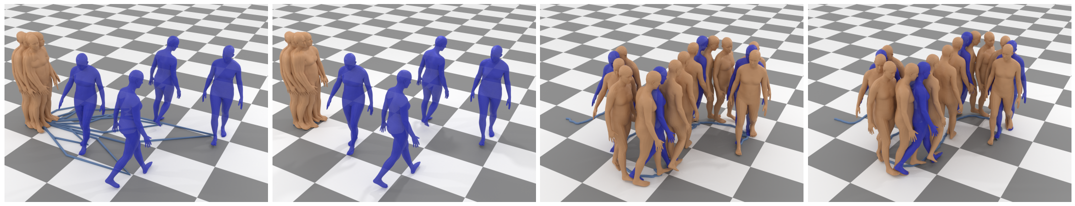
Table 4: Ablation results on the HumanML3D test set. All methods are conditioned on $K = 5$ keyframes randomly sampled from the ground truth motion trajectories with the same text prompts in the test set. *IMP* means pure imputation when replacement stops at diffusion step 1. *C=0* refers to pure imputation with replacement at every diffusion step. *RecG* refers to reconstruction guidance with the default guidance weight ($w_r = 20$). *W=5* refers to reconstruction guidance with guidance weight of $w_r = 5$. *CondMDI* is our method trained with randomly sampled frames and joints. *(random frames)* denotes training with randomly-sampled full keyframes.

Method	FID ↓	R-precision ↑ (Top-3)	Diversity →	Foot skating ratio ↓	Keyframe err ↓
Real	0.002	0.797	9.503	0.000	0.000
IMPC=0	8.6204	0.5710	6.3448	0.1499	**0.0034**
IMP	0.3600	**0.6837**	9.0170	0.1198	0.5150
IMP+RecG	1.7072	0.6498	8.0083	0.1720	**0.0034**
IMP+RecGW=5	4.4881	0.6193	7.0836	0.1728	**0.0034**
CondMDI (random frames)	0.1822	0.6821	9.2648	0.0920	<u>0.1165</u>
CondMDI	**0.1731**	<u>0.6823</u>	**9.3053**	**0.0850**	0.1789

7 CONCLUSION

We have presented a simple and flexible diffusion-based method for keyframe motion completion. It allows for flexibility at inference time and has motion quality comparable to the current state-of-the-art for diffusion-based models. Our method can be used with any backbone motion diffusion model with minimal changes and can therefore readily take advantage of continuing improvements there. We demonstrate our mask-conditioned method with sparse and dense keyframes, partial keyframes, and text conditioning, and show its ability to generate diverse samples. In addition to comparing to related work on the HumanML3D dataset, we give empirical results for several ablations and alternative inference-time conditioning variations.

Our work comes with a number of limitations and related future work. The resulting motions still exhibit some minor footskate and



(a) Imputation C=0 (b) Imputation C=1 (c) Imputation with guidance (d) Pure conditional model

Figure 5: Ablation results on a simple S-walking motion, with keyframes equally spaced = 30 frames apart. While *Imputation* alone fails to follow the keyframes, *Imputation with guidance* is able to do so but suffers from jitters and inconsistencies. C indicates the denoising step in which replacement stops. For a better look please refer to the supplementary video.

motion jitter for highly dynamic motions, which could likely be addressed with an appropriate footskate or smoothness loss or by leveraging a physics-based simulation to track the generated motion. The HumanML3D dataset used for training includes skating and swimming data, and thus removing these outlier motions from the dataset, or providing extra contextual information about them, may also help reduce remaining footskate artefacts. Our current keyframe selection algorithm used during training is fully randomized. We are interested in improving the algorithm by grounding it to combinations that are most used in practice. Finally, our model works with keyframes represented with the same representation as the HumanML3D dataset. This redundant data representation introduces a challenge when conditioning on partial keyframes because spatial constraints may correspond to a small number of features, resulting in the model treating these sparse observed values as noise. Therefore, we are interested in extending our framework to address the issues resulting from uneven representation of different features.

ACKNOWLEDGMENTS

We thank Saeid Naderiparizi for his valuable insights and feedback during the early stages of the work. This work was supported by the NSERC grant RGPIN-2020-05929 and was enabled in part by technical support and computational resources provided by Digital Research Alliance of Canada (www.alliancecan.ca).

REFERENCES

Adobe Systems Inc. 2021. Mixamo. https://www.mixamo.com Accessed: 2021-12-25.

Hyemin Ahn, Timothy Ha, Yunho Choi, Hwiyeon Yoo, and Songhwai Oh. 2018. Text2action: Generative adversarial synthesis from language to action. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 5915–5920.

Chaitanya Ahuja and Louis-Philippe Morency. 2019. Language2pose: Natural language grounded pose forecasting. In *2019 International Conference on 3D Vision (3DV)*. IEEE, 719–728.

Emre Aksan, Manuel Kaufmann, and Otmar Hilliges. 2019. Structured prediction helps 3d human motion modelling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 7144–7153.

Okan Arikan and David A Forsyth. 2002. Interactive motion generation from examples. *ACM Transactions on Graphics (TOG)* 21, 3 (2002), 483–490.

Okan Arikan, David A Forsyth, and James F O'Brien. 2003. Motion synthesis from annotations. In *ACM SIGGRAPH 2003 Papers*. 402–408.

Philippe Beaudoin, Stelian Coros, Michiel Van de Panne, and Pierre Poulin. 2008. Motion-motif graphs. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics symposium on computer animation*. 117–126.

Uttaran Bhattacharya, Nicholas Rewkowski, Abhishek Banerjee, Pooja Guhan, Aniket Bera, and Dinesh Manocha. 2021. Text2gestures: A transformer-based network for generating emotive body gestures for virtual agents. In *2021 IEEE virtual reality and 3D user interfaces (VR)*. IEEE, 1–10.

Michael Büttner and Simon Clavet. 2015. Motion matching-the road to next gen animation. *Proc. of Nucl. ai* 1, 2015 (2015), 2.

Jinxiang Chai and Jessica K Hodgins. 2007. Constraint-based motion optimization using a statistical dynamic model. In *ACM SIGGRAPH 2007 papers*. 8–es.

Xin Chen, Biao Jiang, Wen Liu, Zilong Huang, Bin Fu, Tao Chen, and Gang Yu. 2023. Executing your Commands via Motion Diffusion in Latent Space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 18000–18010.

Rishabh Dabral, Muhammad Hamza Mughal, Vladislav Golyanik, and Christian Theobalt. 2023. Mofusion: A framework for denoising-diffusion-based motion synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9760–9770.

Prafulla Dhariwal and Alexander Nichol. 2021. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems* 34 (2021), 8780–8794.

Laurent Dinh, David Krueger, and Yoshua Bengio. 2014. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516* (2014).

Yinglin Duan, Tianyang Shi, Zhengxia Zou, Yenan Lin, Zhehui Qian, Bohan Zhang, and Yi Yuan. 2021. Single-shot motion completion with transformer. *arXiv preprint arXiv:2103.00776* (2021).

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.

Anindita Ghosh, Noshaba Cheema, Cennet Oguz, Christian Theobalt, and Philipp Slusallek. 2021. Synthesis of compositional animations from textual descriptions. In *Proceedings of the IEEE/CVF international conference on computer vision*. 1396–1406.

Partha Ghosh, Jie Song, Emre Aksan, and Otmar Hilliges. 2017. Learning human motion models for long-term predictions. In *2017 International Conference on 3D Vision (3DV)*. IEEE, 458–466.

Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. 2022. Generating diverse and natural 3d human motions from text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5152–5161.

Chuan Guo, Xinxin Zuo, Sen Wang, Shihao Zou, Qingyao Sun, Annan Deng, Minglun Gong, and Li Cheng. 2020. Action2motion: Conditioned generation of 3d human motions. In *Proceedings of the 28th ACM International Conference on Multimedia*. 2021–2029.

Félix G Harvey and Christopher Pal. 2018. Recurrent transition networks for character locomotion. In *SIGGRAPH Asia 2018 Technical Briefs*. 1–4.

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.

William Harvey, Saeid Naderiparizi, Vaden Masrani, Christian Weilbach, and Frank Wood. 2022. Flexible diffusion modeling of long videos. *Advances in Neural Information Processing Systems* 35 (2022), 27953–27965.

Chengan He, Jun Saito, James Zachary, Holly Rushmeier, and Yi Zhou. 2022. Nemf: Neural motion fields for kinematic animation. *Advances in Neural Information Processing Systems* 35 (2022), 4244–4256.

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.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems* 33 (2020), 6840–6851.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. 2022. Video Diffusion Models. arXiv:2204.03458 [cs.CV]

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.

Eugene Hsu, Sommer Gentry, and Jovan Popović. 2004. Example-based control of human motion. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*. 69–77.

Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. 2022. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991* (2022).

Roy Kapon, Guy Tevet, Daniel Cohen-Or, and Amit H Bermano. 2023. MAS: Multi-view Ancestral Sampling for 3D motion generation using 2D diffusion. *arXiv preprint arXiv:2310.14729* (2023).

Korrawe Karunratanakul, Konpat Preechakul, Supasorn Suwajanakorn, and Siyu Tang. 2023a. GMD: Controllable Human Motion Synthesis via Guided Diffusion Models. *arXiv preprint arXiv:2305.12577* (2023).

Korrawe Karunratanakul, Konpat Preechakul, Supasorn Suwajanakorn, and Siyu Tang. 2023b. Guided Motion Diffusion for Controllable Human Motion Synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2151–2162.

Jihoon Kim, Jiseob Kim, and Sungjoon Choi. 2022. FLAME: Free-form Language-based Motion Synthesis & Editing. *arXiv preprint arXiv:2209.00349* (2022).

Lucas Kovar and Michael Gleicher. 2004. Automated extraction and parameterization of motions in large data sets. *ACM Transactions on Graphics (ToG)* 23, 3 (2004), 559–568.

Lucas Kovar, Michael Gleicher, and Frédéric Pighin. 2002. Motion graphs. Vol. 21. 473–482.

Jehee Lee, Jinxiang Chai, Paul SA Reitsma, Jessica K Hodgins, and Nancy S Pollard. 2002. Interactive control of avatars animated with human motion data. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*. 491–500.

Jehee Lee and Kang Hoon Lee. 2004. Precomputing avatar behavior from human motion data. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*. 79–87.

Sergey Levine, Jack M Wang, Alexis Haraux, Zoran Popović, and Vladlen Koltun. 2012. Continuous character control with low-dimensional embeddings. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 1–10.

Jiaman Li, Ruben Villegas, Duygu Ceylan, Jimei Yang, Zhengfei Kuang, Hao Li, and Yajie Zhao. 2021. Task-generic hierarchical human motion prior using vaes. In *2021 International Conference on 3D Vision (3DV)*. IEEE, 771–781.

Zimo Li, Yi Zhou, Shuangjiu Xiao, Chong He, Zeng Huang, and Hao Li. 2017. Auto-conditioned recurrent networks for extended complex human motion synthesis. *arXiv preprint arXiv:1707.05363* (2017).

Han Liang, Wenqian Zhang, Wenxuan Li, Jingyi Yu, and Lan Xu. 2023. InterGen: Diffusion-based Multi-human Motion Generation under Complex Interactions. *arXiv preprint arXiv:2304.05684* (2023).

Hung Yu Ling, Fabio Zinno, George Cheng, and Michiel Van De Panne. 2020. Character controllers using motion vaes. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 40–1.

Wan-Yen Lo and Matthias Zwicker. 2008. Real-time planning for parameterized human motion. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 29–38.

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. 2022. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11461–11471.

Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. 2019a. AMASS: Archive of Motion Capture as Surface Shapes. In *International Conference on Computer Vision*. 5442–5451.

Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. 2019b. AMASS: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF international conference on computer vision*. 5442–5451.

James McCann and Nancy Pollard. 2007. Responsive characters from motion fragments. In *ACM SIGGRAPH 2007 papers*. 6–es.

Jianyuan Min and Jinxiang Chai. 2012. Motion graphs++ a compact generative model for semantic motion analysis and synthesis. *ACM Transactions on Graphics (TOG)* 31, 6 (2012), 1–12.

Tomohiko Mukai and Shigeru Kuriyama. 2005. Geostatistical motion interpolation. In *ACM SIGGRAPH 2005 Papers*. 1062–1070.

Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. 2021. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741* (2021).

Boris N Oreshkin, Antonios Valkanas, Félix G Harvey, Louis-Simon Ménard, Florent Bocquelet, and Mark J Coates. 2023. Motion In-Betweening via Deep Δ-Interpolator. *IEEE Transactions on Visualization and Computer Graphics* (2023).

Katherine Pullen and Christoph Bregler. 2002. Motion capture assisted animation: Texturing and synthesis. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*. 501–508.

Jia Qin, Youyi Zheng, and Kun Zhou. 2022. Motion in-betweening via two-stage transformers. *ACM Transactions on Graphics (TOG)* 41, 6 (2022), 1–16.

Sigal Raab, Inbal Leibovitch, Guy Tevet, Moab Arar, Amit H Bermano, and Daniel Cohen-Or. 2023. Single Motion Diffusion. *arXiv preprint arXiv:2302.05905* (2023).

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.

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* 1, 2 (2022), 3.

Charles Rose, Michael F Cohen, and Bobby Bodenheimer. 1998. Verbs and adverbs: Multidimensional motion interpolation. *IEEE Computer Graphics and Applications* 18, 5 (1998), 32–40.

Charles F Rose III, Peter-Pike J Sloan, and Michael F Cohen. 2001. Artist-directed inverse-kinematics using radial basis function interpolation. In *Computer graphics forum*, Vol. 20. Wiley Online Library, 239–250.

Alla Safonova and Jessica K Hodgins. 2007. Construction and optimal search of interpolated motion graphs. In *ACM SIGGRAPH 2007 papers*. 106–es.

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 L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. 2022b. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems* 35 (2022), 36479–36494.

Yonatan Shafir, Guy Tevet, Roy Kapon, and Amit H Bermano. 2023. Human motion diffusion as a generative prior. *arXiv preprint arXiv:2303.01418* (2023).

Yijun Shen, He Wang, Edmond SL Ho, Longzhi Yang, and Hubert PH Shum. 2017. Posture-based and action-based graphs for boxing skill visualization. *Computers & Graphics* 69 (2017), 104–115.

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.

Yang Song and Stefano Ermon. 2019. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems* 32 (2019).

Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-or, and Amit Haim Bermano. 2023. Human Motion Diffusion Model. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=SJ1kSyO2jwu

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

Hongsong Wang, Liang Wang, Jiashi Feng, and Daquan Zhou. 2022. Velocity-to-velocity human motion forecasting. *Pattern Recognition* 124 (2022), 108424.

Yiming Xie, Varun Jampani, Lei Zhong, Deqing Sun, and Huaizu Jiang. 2023. Omni-Control: Control Any Joint at Any Time for Human Motion Generation. *arXiv preprint arXiv:2310.08580* (2023).

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*. 16010–16021.

Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Shaoli Huang, Yong Zhang, Hongwei Zhao, Hongtao Lu, and Xi Shen. 2023. T2m-gpt: Generating human motion from textual descriptions with discrete representations. *arXiv preprint arXiv:2301.06052* (2023).

Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. 2022. Motiondiffuse: Text-driven human motion generation with diffusion model. *arXiv preprint arXiv:2208.15001* (2022).

Xinyi Zhang and Michiel van de Panne. 2018. Data-driven autocompletion for keyframe animation. In *Proceedings of the 11th ACM SIGGRAPH Conference on Motion, Interaction and Games*. 1–11.

Yi Zhou, Jingwan Lu, Connelly Barnes, Jimei Yang, Sitao Xiang, et al. 2020. Generative tweening: Long-term inbetweening of 3d human motions. *arXiv preprint arXiv:2005.08891* (2020).