Hierarchical Generation of Human-Object Interactions with Diffusion Probabilistic Models

Huaijin Pi[1], Sida Peng[1], Minghui Yang[2], Xiaowei Zhou[1], Hujun Bao[1*]
[1] State Key Lab of CAD&CG, Zhejiang University [2] Ant Group

Abstract

This paper presents a novel approach to generating the 3D motion of a human interacting with a target object, with a focus on solving the challenge of synthesizing long-range and diverse motions, which could not be fulfilled by existing auto-regressive models or path planning-based methods. We propose a hierarchical generation framework to solve this challenge. Specifically, our framework first generates a set of milestones and then synthesizes the motion along them. Therefore, the long-range motion generation could be reduced to synthesizing several short motion sequences guided by milestones. The experiments on the NSM, COUCH, and SAMP datasets show that our approach outperforms previous methods by a large margin in both quality and diversity. The source code is available on our project page https://zju3dv.github.io/hghoi.

1. Introduction

Scene-aware motion generation [19] aims to synthesize 3D human motion given a 3D scene model to enable virtual humans to naturally wander around scenes and interact with objects, which has a variety of applications in AR/VR, filmmaking, and video games.

Unlike traditional motion generation methods for character control which aim to generate short or repeated motion on the fly guided by a user's control signals [56], we focus on the setting of generating long-term human-object interactions [56, 19, 76] given a starting position of the human and a target object model. This setting brings in new challenges. First, the entire approaching process and the human-object interaction should be coherent, which requires the capability of modeling long-range interaction between the human and the object. Second, in the context of content generation, the generative model should be able to synthesize diverse motions as there are many plausible ways for a real human to approach and interact with the target object.

*Corresponding author.

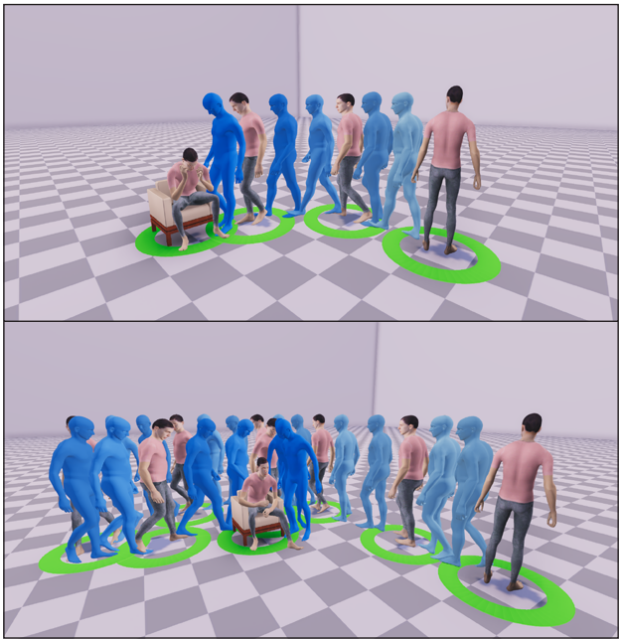

Figure 1. **Generation of human-object interactions.** Given an object, our method first predicts a set of milestones, where the rings indicate the positions and the humans with pink clothes represent the local poses. Then the motions are infilled between milestones. This figure shows that our method can generate diverse milestones and motions with the same object. The flow of time is shown with a color code where darker blue denotes the later frame.

Existing methods for motion synthesis can be roughly characterized into online generation and offline generation. Most online methods [56, 19, 76] focus on real-time control of characters. Given a target object, they generally use auto-regressive models to recurrently generate future motions by feeding their predictions. Although they have been widely used for interactive scenarios like video games, their motion quality is not satisfactory enough for long-term generation [63]. A plausible cause is the error accumulation in the auto-regressive process, where the errors in previous predictions are fed back as the model input, as discussed in [40, 26, 45, 46, 22]. To improve the motion quality,

some recent offline methods [6, 65, 64, 63] employ a multi-stage framework, which first generates the trajectory and then synthesizes motions. TDNS [63] generates paths by combining the cVAE model [33] and deterministic planning methods like A* [18]. Although this strategy can produce reasonable paths, the path diversity is limited, as demonstrated by our experimental results in Sec. 4.4.

In this paper, we propose a novel offline approach for synthesizing long-term and diverse human-object interactions. Our innovation lies in a hierarchical generation strategy, which first predicts a set of milestones and then generates human motions between milestones. Fig. 1 illustrates the basic idea. Specifically, given the starting position and the target object, we design a *milestone generation module* to synthesize a set of milestones along the motion trajectory, each of which encodes the local pose and indicates the transition point during the human movement. Based on these milestones, we employ a *motion generation module* to produce the full motion sequence. Thanks to the milestones, we simplify the long-sequence generation into synthesizing several short motion sequences. Furthermore, the local pose at each milestone is generated by a transformer that considers the global dependency [62], leading to temporally consistent results, which further contribute to coherent motions.

In addition to our hierarchical generation framework, we further exploit diffusion models [53, 23, 54] to synthesize human-object interactions. Previous diffusion models for motion synthesis [31, 59] combine transformer [62] and Denoising Diffusion Probabilistic Model (DDPM) [23]. Directly applying them to our setting is prohibitively compute-intensive due to the long motion sequences and may lead to the GPU memory explosion [50]. Because our hierarchical generation framework converts the long-term generation to the synthesis of several short sequences, the required GPU memory is reduced to the same level of short-term motion generation. Therefore, we can efficiently leverage the transformer DDPM to synthesize long-term motion sequences, which improves the generation quality.

We validate our design choices on the NSM [56], COUCH [76], and SAMP [19] datasets with extensive experiments. On these datasets, our hierarchical framework outperforms previous methods significantly in both motion quality and diversity.

2. Related Work

2.1. Motion synthesis

Motion synthesis is a long-standing problem in computer graphics and vision [68, 51, 42, 9]. With the rapid development of deep learning, recent works have applied neural networks to motion synthesis [14, 40, 26, 72]. Some methods are deterministic [28, 81, 44, 67] while others try to predict stochastic motions by VAE [70, 1, 71, 79, 45] and GAN [4, 34]. To further improve the performance, some works apply GCN [67, 10, 66, 35, 39] or transformers [5] to extract the features from the human skeleton. To handle the ambiguity of human motion, some works [26, 55] propose to employ phase signals to guide the motions. Recent works start to consider the scene context [56, 6]. NSM [56] is the first work that aims at synthesizing human motions with object-level interactions with specific action labels. Based on NSM, SAMP [19] applies cVAE to predict diverse motions. These works [56, 19, 76] mainly focus on the interaction with one or two objects while others [6, 65] generate motions with a full scene (including the information about floors and walls) as the input. [6] predicts future motions of the given motion sequence by first predicting the trajectory and then generating motions based on a 2D image of the scene. Similar to this pipeline, [65] applies GAN [16]. Furthermore, [64] proposes a framework that first places pose on a human-provided path and then synthesizes motions in a 3D scene with a full-scene scan. [63] combines A* and cVAE to generate diverse human motions. To further control human motion, [80] employs the gaze, and COUCH [76] explicitly models the hand contacts to guide the prediction. Some methods [7, 21] also enable physically simulated characters to perform scene interaction tasks including sitting [7, 21] and carrying boxes [21]. Some works also focus on grasp [58, 69], manipulation [73] and the interaction with dynamic objects [57].

Ours vs. others. This work follows the setting of [56, 19] and focuses on object-level interaction. In contrast to [56, 19] which generate motions mainly based on auto-regressive models, we design a hierarchical framework to synthesize motions. Different from [6, 64], our work focuses on much more long-term generation (longer than 10 seconds) while [6] is 2 seconds and [64] is 6 seconds. Instead of planning the path auto-regressively with an extra network to generate diverse trajectories like [63], our method directly predicts a set of milestones to describe the approaching process which is inherently diverse. In addition, most methods [19, 76, 64, 63] rely on cVAE to generate stochastic motions while we exploit DDPM [23] to synthesize trajectories and motions.

2.2. Diffusion models

Diffusion models [53] are a class of likelihood-based methods that generate samples by gradually removing noise from a signal. Then, [23, 54] develop the diffusion models for high-quality image generation. To control the generated results, [12] proposes classifier guidance for trading off diversity for fidelity. Later, the classifier-free model [24] achieves better results [48] in text-conditioned image generation. In addition, diffusion models have been successfully applied to other domains like the generation of videos

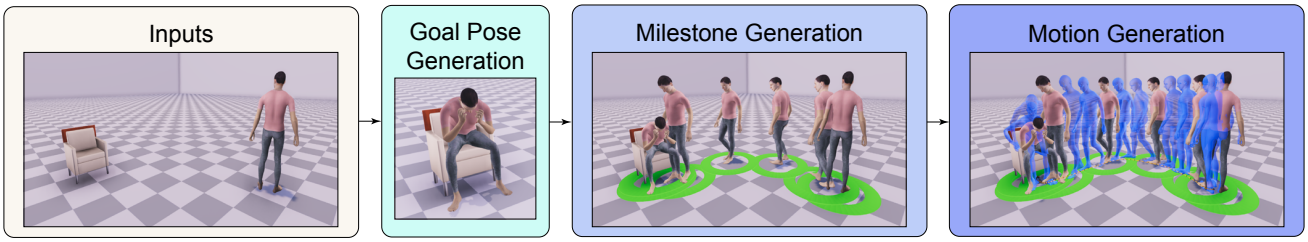

Figure 2. **Overview of our pipeline.** Our pipeline consists of three components: First, the goal pose is synthesized given the object. Then, a number of milestones with local poses are predicted based on the goal pose. Finally, the trajectory and the full motion sequences are infilled between the milestones.

[52, 25] and 3D contents [47].

There are some works [74, 59, 31, 38] that apply DDPM to synthesize motions. [3, 8] also explore latent diffusion models [50] for motion generation. However, most works focus on text-conditioned [74, 11], audio-driven [82] and music-driven [60, 82] motion generation while we target human-object interaction. In this work, we apply transformer DDPM [23] to a multi-stage framework, which separately generates trajectories and synthesizes motions.

There are some concurrent works [27] that apply DDPM to synthesize motions in a scene. However, [27] is designed for short-term motion generation (around 2 seconds) while we target long-term human motion generation (longer than 10 seconds). Different from existing work [59, 27], we employ DDPM in a multi-stage framework, where trajectories and motions are separately predicted.

3. Methods

Given the object $\mathbf{I}$ and the starting point $\mathbf{s}$, our goal is to synthesize 3D human motions $\{(\mathbf{r}_i, \boldsymbol{\theta}_i)\}_{i=1}^{N}$ with human-object interactions. $\mathbf{r}_i$ is the root trajectory, and the $\boldsymbol{\theta}_i$ indicates local pose at i-th frame.

We design a hierarchical motion generation framework as shown in Fig. 2. First, we employ GoalNet[19] to predict an interaction goal on the object. Then, we generate the goal pose (Sec. 3.1) to explicitly model the human-object interaction. Next, our milestone generation module (Sec. 3.2) estimates the length of the milestones, produces the trajectory of the milestones from the starting point to the goal, and places milestone poses. Therefore, the long-range motion generation is decomposed into combinations of short-range motion synthesis. Finally, we design a motion generation module (Sec. 3.3) to synthesize the trajectory between the milestones and infill the motions.

3.1. Goal pose generation

We call the pose in which a person interacts with an object and remains stationary a goal pose. To synthesize diverse human motion, we first generate a goal pose interacting with the object following [58, 69, 63]. Most methods [78, 75, 20] generate human poses using the cVAE model.

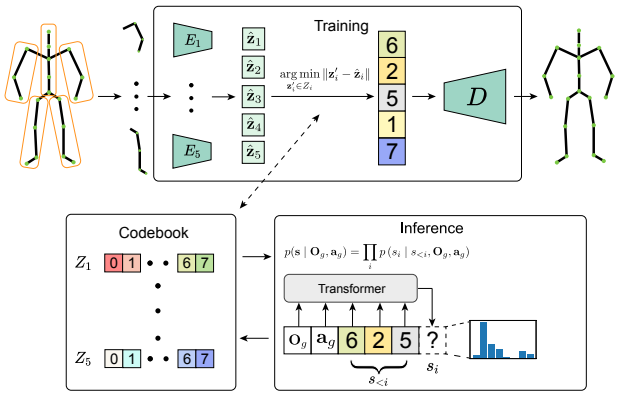

Figure 3. **Part VQ-VAE.** Part VQ-VAE first splits the skeleton into multiple parts and learns the codebooks separately. The composition of different parts is subsequently modeled with the autoregressive prediction model.

They project the poses to the standard normal distribution in the continuous space [33]. Empirically, we find that cVAE models do not perform well in our setting. To overcome this challenge, we introduce VQ-VAE [61, 49] to model the data distribution which exploits a discrete representation to cluster the data in a finite set of points [37]. We hypothesize that limited data of goal pose from the SAMP dataset [19] can always be clustered by VQ-VAE but may not be enough for learning a continuous latent space for VAE [41]. In addition, based on the observation that different human poses may share similar properties [15, 29] (e.g., humans may sit with different hand positions but the same leg positions), we split joints into L ($L = 5$) different non-overlapping groups like MotionPuzzle [29].

Quantization. As shown in Fig. 3, the goal pose $\boldsymbol{\theta}_g$ is split into separate joint groups as $\boldsymbol{\theta}_g = \{\boldsymbol{\theta}_{gi}\}_{i=1}^{L}$. Then a discrete codebook $\mathbf{Z}_i$ with a list of vectors compares the output $\hat{\mathbf{z}}_i$ from the encoder E_i to find the closest vector in Euclidean distance. The L vectors with minimal distances will be concatenated and fed into a shared decoder D to

reconstruct $\boldsymbol{\theta}_g$. The loss function is defined as:

$$\mathcal{L}(\boldsymbol{\theta}_g, D(\mathbf{z})) = \|\boldsymbol{\theta}_g - D(\mathbf{z})\|_2^2 + \sum_{i=1}^{L} \|sg[E_i(\boldsymbol{\theta}_{gi})] - \mathbf{z}_i\|_2^2 + \beta \sum_{i=1}^{L} \|sg[\mathbf{z}_i] - E_i(\boldsymbol{\theta}_{gi})\|_2^2, \tag{1}$$

where

$$\mathbf{z} = [\mathbf{z}_1, ..., \mathbf{z}_L], \tag{2}$$

$$\mathbf{z}_i = \underset{\mathbf{z}_i' \in Z_i}{\arg\min} \|\mathbf{z}_i' - \hat{\mathbf{z}}_i\|, \tag{3}$$

$$\hat{\mathbf{z}}_i = E(\boldsymbol{\theta}_{gi}). \tag{4}$$

Here, the term $sg[\cdot]$ donotes the stop-gradient operator, and $\|sg[\mathbf{z}_i] - E_i(\boldsymbol{\theta}_{gi})\|_2^2$ is the commitment loss [49] controlled by a weighting factor β.

Generation. With E_i and D available, we can represent $\boldsymbol{\theta}_g = \{\boldsymbol{\theta}_{g1}, ..., \boldsymbol{\theta}_{gL}\}$ by a sequence of the part-based codebook indices. To be more specific, we use E_i to extract the feature from $\boldsymbol{\theta}_{gi}$ and find the closest vector $\mathbf{z}_i \in \mathbf{Z}_i$. Then we use $s_i \in \{0, ..., |\mathbf{Z}_i| - 1\}$ to indicate the index of $\mathbf{z}_i$ in $\mathbf{Z}_i$. Therefore, $\boldsymbol{\theta}_g$ can be represented by $\mathbf{s} = \{s_1, ..., s_L\}$.

To generate a natural goal pose, we convert the problem to predict a sequence of indices that can represent $\boldsymbol{\theta}_g$. We formulate the inference as a conditional auto-regressive process and employ a transformer [62] to learn to predict the distribution of possible indices [13]. The condition contains the environment around the goal $\mathbf{O}_g$, and the action $\mathbf{a}_g$. Following NSM [56] and COUCH [76], a cylindrical volume of a pre-defined radius and height is created around the goal. Within this volume, spheres are uniformly sampled and the occupancies corresponding to the object of these spheres are calculated. Then, these occupancies are flattened to form a feature vector denoted as $\mathbf{O}_g$. $\mathbf{a}_g$ is a vector indicating the action type. These variables are fed into the transformer as tokens. Our target is to learn the likelihood of the sequence:

$$p(\mathbf{s} \mid \mathbf{O}_g, \mathbf{a}_g) = \prod_i p\left(s_i \mid s_{<i}, \mathbf{O}_g, \mathbf{a}_g\right). \tag{5}$$

After predicting the indices, we map them back to their corresponding codebook entries to get the quantized features $\mathbf{z} = [\mathbf{z}_1, ..., \mathbf{z}_L]$, which are fed into the decoder D to generate the goal pose $\boldsymbol{\theta}_g$.

3.2. Milestone generation

Based on the starting pose and the goal pose, we can generate the milestone trajectory and synthesize the local poses at the milestones. Following [74, 59], we build a transformer DDPM [23] and apply it to generate the milestones for better quality. Because the length of motion data is unknown and can be arbitrary (e.g., the human could quickly walk towards the chair and sit down or sit after walking around the chair slowly), we predict the length of milestones, denoted by N. Then we synthesize N milestone points and place the local poses on these points.

Transformer DDPM. Here we first briefly introduce DDPM [23], which is learned to reverse a diffusion process. Formally, the diffusion model [53] is defined as a latent variable model of the form $p_\theta(\mathbf{x}_0) := \int p_\theta(\mathbf{x}_{0:T})\, d\mathbf{x}_{1:T}$, where $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ is the data and $\mathbf{x}_1, \ldots, \mathbf{x}_T$ are the latents. $p_\theta(\mathbf{x}_{0:T})$ is formulated as a Markov chain as:

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t), \tag{6}$$

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)). \tag{7}$$

Diffusion models approximate posterior $q(\mathbf{x}_{1:T} \mid \mathbf{x}_0)$ as a Markov chain that gradually adds Gaussian noise to the data with variance schedules given by β_t:

$$q(\mathbf{x}_{1:T} \mid \mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t \mid \mathbf{x}_{t-1}), \tag{8}$$

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) := \mathcal{N}\left(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I}\right). \tag{9}$$

In contrast to adding noises on $\mathbf{x}_0$ sequentially, DDPM formulates the diffusion process as:

$$q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}\right), \tag{10}$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$. Hence, we can generate $\mathbf{x}_t$ by sampling a noise ϵ as the training data. DDPM employs a neural network to model $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$ and the inference is to gradually denoise $\mathbf{x}_t$ from $t = T$ to $t = 1$ where $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

Like existing works [74, 31] that apply DDPM in the motion domain, we employ a transformer decoder [62] as our architecture of DDPM. The transformer takes the noise $\mathbf{x}_t$ and the condition $\mathbf{C}$ as input. The condition $\mathbf{C}$ means variables related to the generation, which will be described in detail in each subsection. The diffusion time-step t is in the sinusoidal position embeddings form [62] and is injected into each block in the transformer. Different from existing works [59, 74] that assume the length of the motion sequences is already given, we insert a parallel branch to estimate the length of milestones by taking the length token $\mathbf{H}^{tok}$ and the condition $\mathbf{C}$ as the input, as shown in Fig. 4. The length prediction head is an MLP and predicts a multinomial distribution over discrete length indices $\{1, 2, ..., N_{max}\}$ like [17], where N_{max} represents the

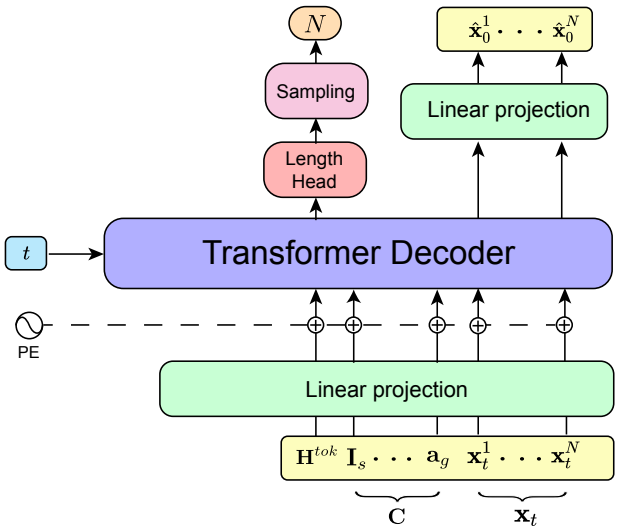

Figure 4. **Overview of transformer DDPM for milestone generation.** The model first takes the length token $\mathbf{H}^{tok}$ and the condition $\mathbf{C}$ as the input to predict the data length. Then it constructs the noise sequence $\mathbf{x}_T^{1:N}$ with length N. In the diffusion process, it is fed with $\mathbf{C}$ and the sequence $\mathbf{x}_t^{1:N}$ at time-step t to predict the target $\hat{\mathbf{x}}_0^{1:N}$. For other sub-modules, we remove the length prediction head.

length of the longest sequences for training. We use cross-entropy loss as the loss function. At inference time, we sample the length N from the estimated distribution. Next, we construct N milestones as the input to the transformer. We predict the denoised data $\hat{\mathbf{x}}_0$ based on the condition $\mathbf{C}$ by the DDPM f. This is formulated as $\hat{\mathbf{x}}_0 = f(\mathbf{x}_t, t, \mathbf{C})$, and the training loss is defined as:

$$\mathcal{L} = \mathrm{E}_{t\in[1,T],\mathbf{x}_0\sim q(\mathbf{x}_0),\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\left[\|\mathbf{x}_0 - \hat{\mathbf{x}}_0\|\right]. \quad (11)$$

In our setting, we use a two-step strategy like [64, 63], where two transformer DDPMs are applied to first generate the milestone points and then synthesize the local pose at each milestone.

Generation of milestone points. The milestone points are conditioned on the object, the information of the starting point, and the goal. We then define the condition as:

$$\mathbf{C}_\mathbf{m} = \{\mathbf{I}_s, \mathbf{I}_g, \mathbf{O}_s, \mathbf{O}_g, \mathbf{g}, \mathbf{s}\}, \quad (12)$$

where $\mathbf{I}_s$ and $\mathbf{I}_g$ are the object representation relative to the starting point and the goal. Similar to other methods [56, 19, 76], the object representation is modeled by an $8\times8\times8$ grid with their positions and occupancies. Following NSM [56] and COUCH [76], we also explicitly model the occupancies around the starting point and the goal with $\mathbf{O}_s$ and $\mathbf{O}_g$ (the same form as the occupancy feature described in Sec. 3.1).

The information of goal $\mathbf{g}$ is defined as $\{\mathbf{r}_g, \mathbf{a}_g, \boldsymbol{\theta}_g\}$, where $\mathbf{r}_g$ means the goal position and orientation in the starting point coordinate system, $\mathbf{a}_g$ is the target action label at the goal, and $\boldsymbol{\theta}_g$ is the goal pose. $\mathbf{s} = \{\mathbf{a}_s, \boldsymbol{\theta}_s\}$ represents the action labels $\mathbf{a}_s$ and the pose $\boldsymbol{\theta}_s$ at the starting point.

The target is to predict the milestone $\{\mathbf{m}_1, ..., \mathbf{m}_N\}$ with length N. Following NSM [56], a bi-directional scheme is employed for the milestone points generation where we predict the roots of milestones in both the starting point coordinate system and the goal coordinate system. The final predicted roots are blended from these two kinds of outputs. The milestone point $\mathbf{m}_i$ is defined as:

$$\mathbf{m}_i = \{\mathbf{r}_i^b, \mathbf{c}_i, \mathbf{w}_i\}. \quad (13)$$

The representation is similar to previous work [19, 56, 76], where $\mathbf{r}_i^b$ indicates the root position and forward directions relative to the starting point and the goal, $\mathbf{c}_i$ is a label vector indicating the contact between the environment and the body, and we use a high-dimensional feature vector $\mathbf{w}_i$ to encode the character state at the milestone following [56, 19, 76]. The details of these variables are provided in the supplementary material. We use transformer DDPM $f_\mathbf{m}$ to predict the length and synthesize milestone points.

Generation of milestone poses. Previous works [64, 63] separately place the poses on the sparse points along the path and infill the motions between, which may lead to unnatural transitions between these poses. On the contrary, we generate the local pose at each milestone with transformer DDPM to build the temporal dependency. With the help of the generated milestone points, we can access the accurate spatial relationship $\mathbf{I}_i$ between the object and the character and the ego-centric environment occupancies $\mathbf{O}_i$ [76] at the i-th milestone. The milestone poses also depend on the milestone state including $\{\mathbf{c}_i, \mathbf{w}_i\}$ at the i-th milestone. Consequently, we define the condition $\mathbf{C}_\mathbf{k}$ as a combination with starting pose $\boldsymbol{\theta}_s$, goal pose $\boldsymbol{\theta}_g$, and frame-wise condition $\boldsymbol{\gamma}_i$ at the i-th milestone as:

$$\mathbf{C}_\mathbf{k} = \{\boldsymbol{\theta}_s, \boldsymbol{\theta}_g, \boldsymbol{\gamma}_1, ..., \boldsymbol{\gamma}_N\}, \quad (14)$$

$$\text{where} \quad \boldsymbol{\gamma}_i = \{\mathbf{I}_i, \mathbf{O}_i, \mathbf{c}_i, \mathbf{w}_i\}. \quad (15)$$

The local poses at the milestones are predicted by the transformer DDPM $f_\mathbf{k}$ without the length prediction head.

3.3. Motion generation

Instead of predicting motions frame-by-frame [56, 19, 76], our approach hierarchically synthesizes the full sequence based on the generated milestones. We follow [6, 64, 63] to first generate the trajectory and then synthesize the motions. Specifically, within two consecutive milestones, we first complete the trajectory. Then, the motions are infilled with the guidance of successive milestone

poses. The two steps are accomplished using two transformer DDPMs (described in Sec. 3.2), respectively. For each step, we carefully design the condition of DDPM to generate the target output.

Trajectory completion. For the trajectory completion between the milestones $\mathbf{m}_i$ and $\mathbf{m}_{i+1}$, we assume that it is only conditioned on the milestones and the object. Thus we define the condition as:

$$\mathbf{C_r} = \{\mathbf{I}_i, \mathbf{I}_{i+1}, \mathbf{O}_i, \mathbf{O}_{i+1}, \mathbf{m}_i, \mathbf{m}_{i+1}, \mathbf{t}_i^{i+1}\}, \quad (16)$$

where $\mathbf{I}_i$ indicates the object representation relative to the milestone i and $\mathbf{O}_i$ denotes the ego-centric occupancies (the same form as the occupancy feature described in Sec. 3.1) around the i-th milestone like NSM [56] and COUCH [76]. $\mathbf{m}_i$ has been shown in Eq. (13). $\mathbf{t}_i^{i+1}$ represents the position and orientation of the $(i+1)$-th milestone in the i-th milestone's coordinate system. Between two consecutive milestones, we generate the trajectory with a length of 2 seconds, which is 61 frames. Similar hyperparameters can be found in previous methods [64, 63].

The target output is the trajectory between two consecutive milestones. Similar to the milestone point, we synthesize the trajectory in the j-th frame with a bi-directional scheme. The trajectory is composed of a set of points that have the same representation as the milestone in Eq. (13). We generate the trajectory with a transformer DDPM $f_{\mathbf{r}}$ similar to the one in Sec. 3.2. Since we assume the trajectory length between two milestones is fixed, the DDPM $f_{\mathbf{r}}$ does not have the length prediction head.

Motion infilling. To synthesize the long-range motion, we convert a long sequence into several fixed-length short sequences with the help of milestone points and milestone poses. For a sub-sequence between the consecutive milestone poses, our goal is to generate the missing local poses over the trajectory. The generated motion has to satisfy the trajectory and naturally transits from a milestone to the next milestone. Like milestone pose generation, we use the same representation of frame-wise condition in Eq. (15). The condition is defined as:

$$\mathbf{C_p} = \{\boldsymbol{\theta}^1, \boldsymbol{\theta}^{61}, \boldsymbol{\gamma}^1, ..., \boldsymbol{\gamma}^{61}\}, \quad (17)$$

where $\boldsymbol{\theta}^1$ and $\boldsymbol{\theta}^{61}$ are the local poses of two consecutive milestones. By taking these inputs, we generate smooth motions using another transformer DDPM $f_{\mathbf{p}}$ without the length prediction head.

4. Experiments

4.1. Implementation details

We train the part VQ-VAE and transformer DDPM models with the Adam optimizer [32]. All the models are trained with a fixed learning rate of 0.0001 with batch size 256. The remaining details are in the supplementary material.

4.2. Datasets and evaluation metrics

Test setting. Our experiments are conducted on the SAMP [19], COUCH [76], and NSM [56] datasets. Provided with a starting point, a starting pose, an object, and an endpoint, the virtual human is asked to approach the object, interact with it, and leave to reach the endpoint. The ablation studies are conducted on the SAMP dataset.

Metrics. Following the previous method [19], we calculate the Fréchet distance (FD) between the generated and ground-truth motions to measure the motion quality. We also conduct user studies and each sequence is evaluated by at least 3 users with scores ranging from 1 to 5. In addition, we calculate the penetration ratio [64, 75, 78] and foot sliding [72, 36] to show the physical plausibility between the 3D object and the synthesized motions. We compute the Average Pairwise Distance (APD) [71, 77] to evaluate the diversity. Specifically, we calculate the APD of synthesized motion, the character's pose during object interactions, and trajectories. Following previous work [56, 19], we calculate PE (positional errors) and RE (rotational errors) to indicate the precision of object interactions. For each test object, we generate multiple sequences. More details are included in the supplementary material.

4.3. Comparison with other methods

Results on the SAMP dataset. On the SAMP dataset [19], we compare our method with online methods SAMP [19] and MoE [72]. As our method is offline, we also implement and modify offline methods SLT [64] and TDNS [63] to our setting. For SLT [64], we employ A* [18] to plan a path and select points along the path as subgoals to form the input for SLT. More details about the implementation of [64, 63] are in the supplementary material. Since MoE often fails to finish the action, we do not calculate the penetration ratio for it. As shown in Tab. 1, our approach outperforms other methods in terms of lower FD, higher user study scores, and higher APD. Furthermore, our method achieves much higher trajectory diversity than SAMP [19]. Although TNDS [63] proposes Neural Mapper (NM) which combines A* [18] and cVAE, the diversity of the generated trajectory is inferior to our method, as indicated by APD$_T$.

Results on the COUCH dataset. As our target is to synthesize diverse motions instead of controlling the characters, we only evaluate the motion quality on the COUCH dataset [76]. Tab. 2 shows that our method outperforms all baselines. Our approach achieves much higher APD$_T$ than other methods. We observe that APD$_T$ of TDNS [63] is

Method	FD ↓	User study ↑	APD_M ↑	APD_P ↑	APD_T ↑	PE↓	RE↓	Penetration↓	Sliding↓
MoE [72]	74.33	2.76	3.50	2.63	52.46	∞	∞	-	1.68
SAMP [19]	57.34	2.86	3.63	3.05	63.18	3.44	4.12	6.98	1.02
SLT* [64]	68.83	2.30	3.08	2.66	40.13	1.77	1.60	4.28	1.71
TDNS* [63]	46.60	2.90	3.68	3.40	66.04	0.45	0.39	5.14	0.94
Ours	**22.34**	**3.62**	**4.06**	**4.52**	**91.38**	**0.39**	**0.32**	**4.00**	**0.50**

Table 1. **Quantitative results on the SAMP dataset.** SLT* and TDNS* mean we modify and implement them on the SAMP dataset. The subscript "M", "P" and "T" stand for "Motion", "Pose" and "Trajectory". PE and RE represent the positional error and rotation error. Sliding denotes foot sliding. ∞ means the method failed to reach the goal.

Method	FD↓	User study ↑	APD_M ↑	APD_P ↑	APD_T ↑	Penetration↓	Sliding↓
NSM [56]	118.98	2.99	0	0	0	8.20	0.59
SAMP [19]	160.12	1.94	0.89	0.18	4.69	4.94	0.72
COUCH [76]	127.19	3.05	1.41	0.66	23.48	7.43	**0.37**
SLT* [64]	93.46	3.01	1.68	1.17	2.86	3.90	1.85
TDNS* [63]	71.72	3.29	2.70	2.19	16.98	5.12	1.17
Ours	**56.35**	**4.27**	**3.22**	**2.30**	**64.96**	**3.54**	0.55

Table 2. **Quantitative results on the COUCH dataset.** SLT* and TDNS* mean we modify and implement them.

Method	FD↓	User study↑	APD_M ↑	APD_P ↑	APD_T ↑	PE↓	RE↓	Penetration↓	Sliding↓
NSM [56]	90.39	3.95	0	0	0	1.72	0.40	6.43	**0.69**
SAMP [19]	68.86	3.77	1.21	0.11	20.01	4.77	4.84	7.83	1.38
Ours	**57.02**	**4.04**	**2.62**	**0.93**	**62.23**	**1.01**	**0.28**	**4.85**	0.80

Table 3. **Quantitative results on the NSM dataset.**

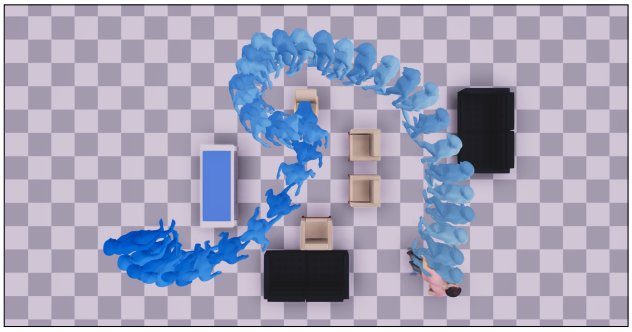

Figure 5. **Results in a cluttered scene.** Our method can generate motions that avoid obstacles in a cluttered scene.

higher than the approaches [19, 64] that employ deterministic A* [18], but much lower than our method. Although COUCH [76] exhibits lower foot sliding than our method, it may sometimes get stuck, resulting in lower foot sliding since the character does not move.

Results on the NSM dataset. On the NSM dataset [56], we compare our approach with SAMP [19] and NSM [56]. Tab. 3 shows that our method outperforms baselines. Compared with the deterministic method NSM, our approach could generate stochastic motion and diverse trajectories.

Variants	FD↓	APD_M ↑	APD_P ↑	APD_T ↑	Penetration↓	Sliding↓
w/o GP	31.07	3.21	2.46	85.21	4.10	0.49
w/o MT	27.87	3.62	3.01	**127.94**	5.30	0.72
w/o MP	25.14	4.01	3.78	85.69	4.58	**0.43**
w/o TC	36.77	2.76	2.42	83.26	4.34	0.63
Ours	**22.34**	**4.06**	4.52	91.38	**4.00**	0.50

Table 4. **Ablation study of the impact of sub-modules.** Although the variant without MT generates more diverse trajectories as shown in APD_T, the motion quality is much worse as indicated by the higher FD, penetration, and sliding. w/o: without. GP: goal pose generation. MT: milestone point generation. MP: milestone pose generation. TC: trajectory completion.

Results in a cluttered scene. We show our generated results in a cluttered scene in Fig. 5. The percentage of frames with penetration is 3.8% for our method and 4.9% for SAMP. More details are in the supplementary material.

Qualitative results. As demonstrated in Fig. 6, our approach achieves better results than baselines [19, 63] on the SAMP dataset. Fig. 7 compares our method with COUCH and TDNS on the COUCH dataset. More qualitative results are in the supplementary material.

4.4. Ablation study

Impact of each sub-module. To show the effectiveness of our hierarchical design, we evaluate our method against four variants where we remove one sub-module for each variant. Tab. 4 indicates that each component improves performance. Generating a whole trajectory leads to more diverse trajectories but the motion quality is worse and has more foot sliding. This validates the necessity of the separate generation of trajectories and motions.

Goal pose generation. We evaluate our goal pose generator with several variants, including cVAE, DDPM, and standard VQ-VAE. For this evaluation, we only replace the goal pose generation module and keep the others the same.

Tab. 5 shows that part VQ-VAE generates more diverse poses than continuous latent space models. The comparison

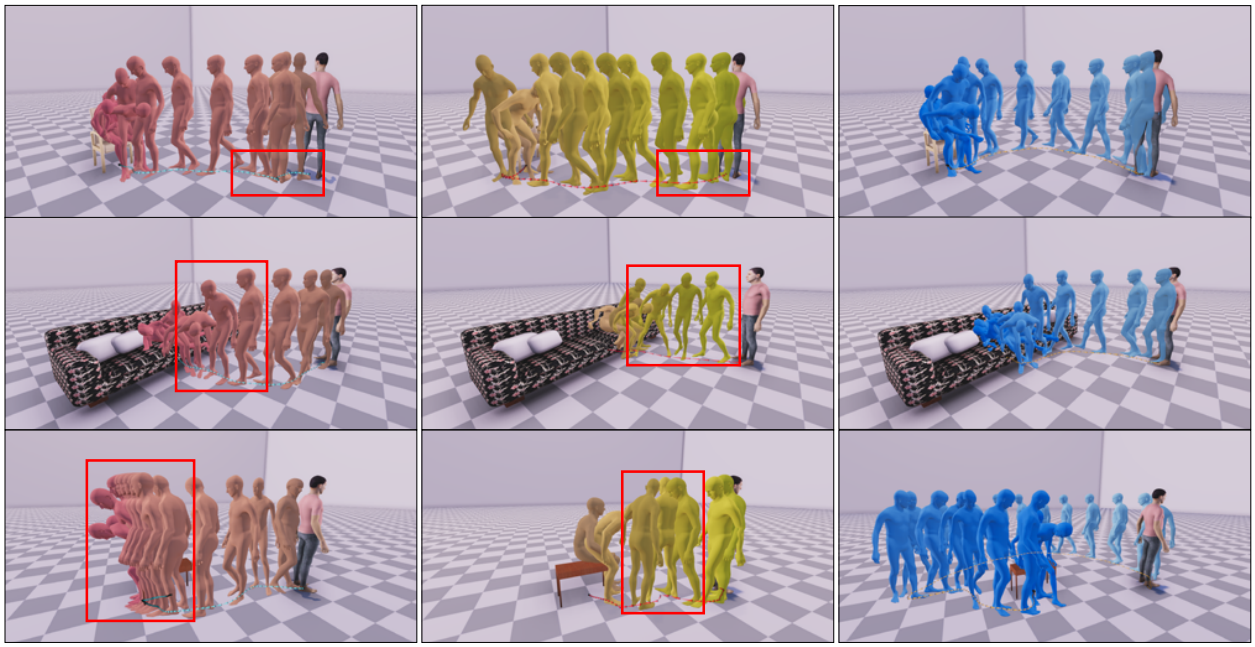

Figure 6. **Qualitative results on the SAMP dataset.** We compare our method with the baselines SAMP [19] and TDNS [63]. The failure cases are pointed out by the red rectangles. Specifically, the first row indicates that SAMP and TDNS tend to walk backward with more foot sliding. The second row shows that the baselines start to sit and lie down before reaching the object while our method synthesizes more natural results. The third row demonstrates that our framework has the capability to generate a long trajectory and walk naturally along the trajectory while SAMP gets stuck near the object and TDNS synthesizes unnatural motions. Lines on the floor indicate trajectories. The human with pink clothes indicates the start position. Darker color denotes later frames in the sequence.

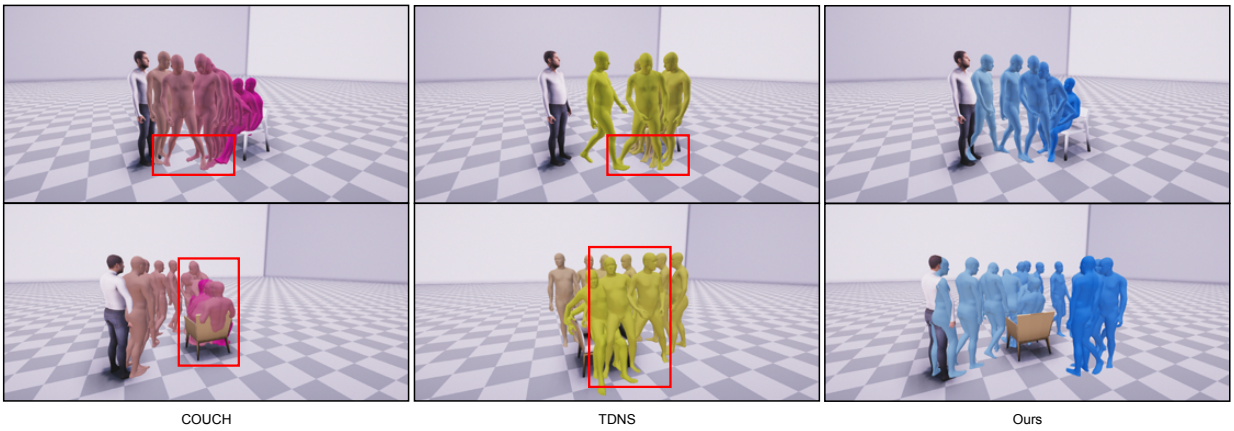

Figure 7. **Qualitative results on the COUCH dataset.** We compare our method with the baseline COUCH [76] and TDNS [63]. The failure cases are pointed out by the red rectangles. Specifically, the first row indicates that COUCH and TDNS tend to sit when the character is far from the object. The second row shows that COUCH may fail to stand up and TDNS may stand up unnaturally. To better visualize the results of TDNS in the second row, we only keep the frames in which the human stands up and goes to the endpoint.

with standard VQ-VAE shows the necessity of our part design. We also try part VQ-VAE for motion infilling, but the results in Tab. 6 show that its performance is worse.

Milestone generation. To further investigate the impact of milestones, we compare our approach with the variants that employ path-planning methods and select points along the path as milestones. For this ablation, we implement A* and NM [63] proposed by TDNS. As demonstrated in Tab. 7, the diversity of generated trajectories of A* [18] is much worse, and the motion quality drops significantly, as indicated by the lower APD$_T$ and higher FD. The trajectory

Variants	FD ↓	APD$_M$ ↑	APD$_P$ ↑	APD$_T$ ↑	Penetration↓	Sliding↓
cVAE	27.06	3.36	3.27	90.52	4.36	**0.47**
DDPM	34.22	3.75	3.37	84.76	4.31	0.49
VQ-VAE	24.77	3.78	3.87	89.19	4.27	0.48
Part VQ-VAE	**22.34**	**4.06**	**4.52**	**91.38**	**4.00**	0.50

Table 5. **Ablation study of goal pose generation.** We implement the goal pose module with different architectures.

Variants	FD↓	APD$_M$ ↑	APD$_P$ ↑	APD$_T$ ↑	Penetration↓	Sliding↓
Part VQ-VAE	28.71	3.85	3.62	83.51	4.32	0.89
DDPM	**22.34**	**4.06**	**4.52**	**91.38**	**4.00**	**0.50**

Table 6. **Ablation study of part VQ-VAE for motion infilling.** We replace the DDPM as part VQ-VAE to predict motions.

Variants	FD↓	APD$_M$ ↑	APD$_P$ ↑	APD$_T$ ↑	Penetration↓	Sliding↓
A*	40.59	4.07	4.42	42.98	4.24	0.88
NM [63]	40.54	**4.15**	4.31	52.74	4.15	0.96
MT	**22.34**	4.06	**4.52**	**91.38**	**4.00**	**0.50**

Table 7. **Ablation study of milestone generation.** We compare our method with the variant based on the path generated by A* path planning [18]. NM: Neural Mapper [63]. MT: milestone point generation.

Method	FD↓	APD$_M$ ↑	APD$_P$ ↑	Penetration↓	Sliding↓
ConvAE [30]	26.50	3.95	4.33	4.82	0.89
SLT [64]	27.95	3.76	4.05	4.13	0.78
Ours	**22.34**	**4.06**	**4.52**	**4.00**	**0.50**

Table 8. **Ablation study of motion infilling module.** We compare our method with other motion infilling methods.

diversity of NM [63] is better than A*, but still worse than our method. The reason why these variants perform poorly might be the low diversity of trajectory that affects the distribution of generated motions for calculating FD.

Motion infilling. To validate our motion infilling module, we compare it with ConvAE [30] and SLT [64]. We only replace the motion infilling module and keep the others the same. The comparison of motion quality and diversity is shown in Tab. 8 and our method outperforms ConvAE [30] and SLT [64] with lower FD.

DDPM vs. VAE. To show the effectiveness of DDPM, we implement a cVAE variant, where we simply replace our transformer DDPM with transformer cVAE [45, 63]. As shown in Tab. 9, although cVAE models could generate more diverse trajectories, their motion quality is far from satisfactory, indicated by the much higher value of FD.

Comparison with other diffusion models. Our approach stands out from architectures in MDM [59] and FLAME [31] by incorporating frame-wise conditions. Tab. 10 demonstrates the significance of the frame-wise conditions.

Arch	FD↓	R-APD$_M$ ↑	R-APD$_P$ ↑	R-APD$_T$ ↑	Penetration↓	Sliding↓
cVAE	29.32	3.65	3.58	**109.55**	5.06	1.12
DDPM	**22.34**	**4.06**	**4.52**	91.38	**4.00**	**0.50**

Table 9. **Evaluation of the architecture for our generation framework.** Although cVAE based variant generates more diverse trajectories, the motion quality drops significantly as indicated by the much higher FD. Arch stands for the architecture type.

Variants	FD↓	APD$_M$ ↑	APD$_P$ ↑	APD$_T$ ↑	Penetration↓	Sliding↓
MDM [59]	38.69	4.01	3.77	72.67	5.19	0.70
MDM [59] + C	23.97	4.02	4.13	89.80	4.87	**0.39**
FLAME [31]	28.93	4.02	4.46	65.46	5.78	0.89
Ours	**22.34**	**4.06**	**4.52**	**91.38**	**4.00**	0.50

Table 10. **Comparison with MDM and FLAME.** C denotes the frame-wise conditions.

More analyses and ablation studies. More detailed analyses and ablation studies of our design choices are provided in the supplementary material.

4.5. Limitations

Although our method can generate diverse and natural motions, there are still some limitations. Our method is offline and cannot be applied to interactive scenarios. We assume that the objects are static and cannot handle moving objects. The diffusion models require a long inference time. It takes 7.13 seconds on average for a 720-frame sequence on a TITAN Xp GPU. The slow speed might be solved by methods that could accelerate diffusion models [43, 2].

5. Conclusion

In this work, we propose a novel hierarchical pipeline for motion synthesis of human-object interactions. Our approach first generates the goal pose and then predicts a set of milestones. Next, we synthesize motions with the guide of milestones. Furthermore, we apply DDPM in our hierarchical pipeline. We also show that our framework could generate more diverse and natural human-object interaction motions than other methods.

Acknowledgements

We thank Jintao Lu, Zhi Cen, Zizhang Li, and Kechun Xu for the valuable discussions. This work was partially supported by the Key Research Project of Zhejiang Lab (No. K2022PG1BB01), NSFC (No. 62172364), and Information Technology Center and State Key Lab of CAD&CG, Zhejiang University.

References

[1] Sadegh Aliakbarian, Fatemeh Sadat Saleh, Mathieu Salzmann, Lars Petersson, and Stephen Gould. A stochastic conditioning scheme for diverse human motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[2] Fan Bao, Chongxuan Li, Jun Zhu, and Bo Zhang. Analytic-DPM: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. In *International Conference on Learning Representations*, 2022. 9

[3] German Barquero, Sergio Escalera, and Cristina Palmero. BeLFusion: Latent Diffusion for Behavior-Driven Human Motion Prediction. *arXiv e-prints*, page arXiv:2211.14304, Nov. 2022. 3

[4] Emad Barsoum, John R. Kender, and Zicheng Liu. Hp-gan: Probabilistic 3d human motion prediction via gan. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1499–149909, 2018. 2

[5] Yujun Cai, Lin Huang, Yiwei Wang, Tat-Jen Cham, Jianfei Cai, Junsong Yuan, Jun Liu, Xu Yang, Yiheng Zhu, Xiaohui Shen, Ding Liu, Jing Liu, and Nadia Magnenat Thalmann. Learning progressive joint propagation for human motion prediction. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII*, page 226–242, Berlin, Heidelberg, 2020. Springer-Verlag. 2

[6] Zhe Cao, Hang Gao, Karttikeya Mangalam, Qi-Zhi Cai, Minh Vo, and Jitendra Malik. Long-term human motion prediction with scene context. In *Computer Vision ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I*, page 387–404, Berlin, Heidelberg, 2020. Springer-Verlag. 2, 5

[7] Yu-Wei Chao, Jimei Yang, Weifeng Chen, and Jia Deng. Learning to sit: Synthesizing human-chair interactions via hierarchical control. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 5887–5895, 2021. 2

[8] Xin Chen, Biao Jiang, Wen Liu, Zilong Huang, Bin Fu, Tao Chen, Jingyi Yu, and Gang Yu. Executing your Commands via Motion Diffusion in Latent Space. *arXiv e-prints*, page arXiv:2212.04048, Dec. 2022. 3

[9] Simon Clavet. Motion matching and the road to next-gen animation. In *GDC*, 2016. 2

[10] Qiongjie Cui, Huaijiang Sun, and Fei Yang. Learning dynamic relationships for 3d human motion prediction. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6518–6526, 2020. 2

[11] Rishabh Dabral, Muhammad Hamza Mughal, Vladislav Golyanik, and Christian Theobalt. MoFusion: A Framework for Denoising-Diffusion-based Motion Synthesis. *arXiv e-prints*, page arXiv:2212.04495, Dec. 2022. 3

[12] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 8780–8794. Curran Associates, Inc., 2021. 2

[13] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12873–12883, June 2021. 4

[14] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. Recurrent network models for human dynamics. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015. 2

[15] Anindita Ghosh, Noshaba Cheema, Cennet Oguz, Christian Theobalt, and Philipp Slusallek. Synthesis of compositional animations from textual descriptions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1396–1406, 2021. 3

[16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Commun. ACM*, 63(11):139–144, oct 2020. 2

[17] Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5152–5161, June 2022. 4

[18] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968. 2, 6, 7, 8, 9

[19] Mohamed Hassan, Duygu Ceylan, Ruben Villegas, Jun Saito, Jimei Yang, Yi Zhou, and Michael J. Black. Stochastic scene-aware motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11374–11384, October 2021. 1, 2, 3, 5, 6, 7, 8

[20] Mohamed Hassan, Partha Ghosh, Joachim Tesch, Dimitrios Tzionas, and Michael J. Black. Populating 3d scenes by learning human-scene interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14708–14718, June 2021. 3

[21] Mohamed Hassan, Yunrong Guo, Tingwu Wang, Michael Black, Sanja Fidler, and Xue Bin Peng. Synthesizing physical character-scene interactions. In *ACM SIGGRAPH 2023 Conference Proceedings*, SIGGRAPH '23, New York, NY, USA, 2023. Association for Computing Machinery. 2

[22] Chengan He, Jun Saito, James Zachary, Holly Rushmeier, and Yi Zhou. Nemf: Neural motion fields for kinematic animation. In *NeurIPS*, 2022. 1

[23] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020. 2, 3, 4

[24] Jonathan Ho and Tim Salimans. Classifier-Free Diffusion Guidance. *arXiv e-prints*, page arXiv:2207.12598, July 2022. 2

[25] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video Diffusion Models. *arXiv e-prints*, page arXiv:2204.03458, Apr. 2022. 3

[26] Daniel Holden, Taku Komura, and Jun Saito. Phase-functioned neural networks for character control. *ACM Trans. Graph.*, 36(4), jul 2017. 1, 2

[27] Siyuan Huang, Zan Wang, Puhao Li, Baoxiong Jia, Tengyu Liu, Yixin Zhu, Wei Liang, and Song-Chun Zhu. Diffusion-based Generation, Optimization, and Planning in 3D Scenes. *arXiv e-prints*, page arXiv:2301.06015, Jan. 2023. 3

[28] Ashesh Jain, Amir R. Zamir, Silvio Savarese, and Ashutosh Saxena. Structural-rnn: Deep learning on spatio-temporal graphs. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5308–5317, 2016. 2

[29] Deok-Kyeong Jang, Soomin Park, and Sung-Hee Lee. Motion puzzle: Arbitrary motion style transfer by body part. *ACM Trans. Graph.*, 41(3), jun 2022. 3

[30] Manuel Kaufmann, Emre Aksan, Jie Song, Fabrizio Pece, Remo Ziegler, and Otmar Hilliges. Convolutional autoencoders for human motion infilling. In *2020 International Conference on 3D Vision (3DV)*, pages 918–927, 2020. 9

[31] Jihoon Kim, Jiseob Kim, and Sungjoon Choi. FLAME: Free-form Language-based Motion Synthesis & Editing. *arXiv e-prints*, page arXiv:2209.00349, Sept. 2022. 2, 3, 4, 9

[32] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv e-prints*, page arXiv:1412.6980, Dec. 2014. 6

[33] Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. *arXiv e-prints*, page arXiv:1312.6114, Dec. 2013. 2, 3

[34] Jogendra Nath Kundu, Maharshi Gor, and R. Venkatesh Babu. Bihmp-gan: Bidirectional 3d human motion prediction gan. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019. 2

[35] Maosen Li, Siheng Chen, Yangheng Zhao, Ya Zhang, Yanfeng Wang, and Qi Tian. Dynamic multiscale graph neural networks for 3d skeleton based human motion prediction. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[36] Hung Yu Ling, Fabio Zinno, George Cheng, and Michiel Van De Panne. Character controllers using motion vaes. *ACM Trans. Graph.*, 39(4), aug 2020. 6

[37] Thomas Lucas, Fabien Baradel, Philippe Weinzaepfel, and Grégory Rogez. Posegpt: Quantization-based 3d human motion generation and forecasting. In *European Conference on Computer Vision (ECCV)*, 2022. 3

[38] Jianxin Ma, Shuai Bai, and Chang Zhou. Pretrained Diffusion Models for Unified Human Motion Synthesis. *arXiv e-prints*, page arXiv:2212.02837, Dec. 2022. 3

[39] Wei Mao, Miaomiao Liu, Mathieu Salzmann, and Hongdong Li. Multi-level motion attention for human motion prediction. *International Journal of Computer Vision*, 129(9):2513–2535, 2021. 2

[40] Julieta Martinez, Michael J. Black, and Javier Romero. On human motion prediction using recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1, 2

[41] Paritosh Mittal, Yen-Chi Cheng, Maneesh Singh, and Shubham Tulsiani. Autosdf: Shape priors for 3d completion, reconstruction and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 306–315, June 2022. 3

[42] Tomohiko Mukai and Shigeru Kuriyama. Geostatistical motion interpolation. *ACM Trans. Graph.*, 24(3):1062–1070, jul 2005. 2

[43] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8162–8171. PMLR, 18–24 Jul 2021. 9

[44] Dario Pavllo, David Grangier, and Michael Auli. Quaternet: A quaternion-based recurrent model for human motion. In *British Machine Vision Conference (BMVC)*, 2018. 2

[45] Mathis Petrovich, Michael J. Black, and Gül Varol. Action-conditioned 3D human motion synthesis with transformer VAE. In *International Conference on Computer Vision (ICCV)*, 2021. 1, 2, 9

[46] Mathis Petrovich, Michael J. Black, and Gül Varol. TEMOS: Generating diverse human motions from textual descriptions. In *European Conference on Computer Vision (ECCV)*, 2022. 1

[47] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. DreamFusion: Text-to-3D using 2D Diffusion. *arXiv e-prints*, page arXiv:2209.14988, Sept. 2022. 3

[48] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical Text-Conditional Image Generation with CLIP Latents. *arXiv e-prints*, page arXiv:2204.06125, Apr. 2022. 2

[49] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019. 3, 4

[50] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022. 2, 3

[51] C. Rose, M.F. Cohen, and B. Bodenheimer. Verbs and adverbs: multidimensional motion interpolation. *IEEE Computer Graphics and Applications*, 18(5):32–40, 1998. 2

[52] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-A-Video: Text-to-Video Generation without Text-Video Data. *arXiv e-prints*, page arXiv:2209.14792, Sept. 2022. 3

[53] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David

Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265, Lille, France, 07–09 Jul 2015. PMLR. 2, 4

[54] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. 2

[55] Sebastian Starke, Ian Mason, and Taku Komura. Deepphase: Periodic autoencoders for learning motion phase manifolds. *ACM Trans. Graph.*, 41(4), jul 2022. 2

[56] Sebastian Starke, He Zhang, Taku Komura, and Jun Saito. Neural state machine for character-scene interactions. *ACM Trans. Graph.*, 38(6), nov 2019. 1, 2, 4, 5, 6, 7

[57] Sebastian Starke, Yiwei Zhao, Taku Komura, and Kazi Zaman. Local motion phases for learning multi-contact character movements. *ACM Trans. Graph.*, 39(4), aug 2020. 2

[58] Omid Taheri, Vasileios Choutas, Michael J. Black, and Dimitrios Tzionas. Goal: Generating 4d whole-body motion for hand-object grasping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13263–13273, June 2022. 2, 3

[59] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Amit H Bermano, and Daniel Cohen-Or. Human motion diffusion model. *arXiv preprint arXiv:2209.14916*, 2022. 2, 3, 4, 9

[60] Jonathan Tseng, Rodrigo Castellon, and C. Karen Liu. EDGE: Editable Dance Generation From Music. *arXiv e-prints*, page arXiv:2211.10658, Nov. 2022. 3

[61] Aaron van den Oord, Oriol Vinyals, and koray kavukcuoglu. Neural discrete representation learning. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 3

[62] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 2, 4

[63] Jingbo Wang, Yu Rong, Jingyuan Liu, Sijie Yan, Dahua Lin, and Bo Dai. Towards diverse and natural scene-aware 3d human motion synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20460–20469, June 2022. 1, 2, 3, 5, 6, 7, 8, 9

[64] Jiashun Wang, Huazhe Xu, Jingwei Xu, Sifei Liu, and Xiaolong Wang. Synthesizing long-term 3d human motion and interaction in 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9401–9411, June 2021. 2, 5, 6, 7, 9

[65] J. Wang, S. Yan, B. Dai, and D. Lin. Scene-aware generative network for human motion synthesis. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12201–12210, Los Alamitos, CA, USA, jun 2021. IEEE Computer Society. 2

[66] Mao Wei, Liu Miaomiao, and Salzemann Mathieu. History repeats itself: Human motion prediction via motion attention. In *ECCV*, 2020. 2

[67] Mao Wei, Liu Miaomiao, Salzemann Mathieu, and Li Hongdong. Learning trajectory dependencies for human motion prediction. In *ICCV*, 2019. 2

[68] D.J. Wiley and J.K. Hahn. Interpolation synthesis of articulated figure motion. *IEEE Computer Graphics and Applications*, 17(6):39–45, 1997. 2

[69] Yan Wu, Jiahao Wang, Yan Zhang, Siwei Zhang, Otmar Hilliges, Fisher Yu, and Siyu Tang. Saga: Stochastic whole-body grasping with contact. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 2, 3

[70] Xinchen Yan, Akash Rastogi, Ruben Villegas, Kalyan Sunkavalli, Eli Shechtman, Sunil Hadap, Ersin Yumer, and Honglak Lee. Mt-vae: Learning motion transformations to generate multimodal human dynamics. In *European Conference on Computer Vision*, pages 276–293. Springer, 2018. 2

[71] Ye Yuan and Kris Kitani. Dlow: Diversifying latent flows for diverse human motion prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 2, 6

[72] He Zhang, Sebastian Starke, Taku Komura, and Jun Saito. Mode-adaptive neural networks for quadruped motion control. *ACM Trans. Graph.*, 37(4), jul 2018. 2, 6, 7

[73] He Zhang, Yuting Ye, Takaaki Shiratori, and Taku Komura. Manipnet: Neural manipulation synthesis with a hand-object spatial representation. *ACM Trans. Graph.*, 40(4), jul 2021. 2

[74] Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. Motiondiffuse: Text-driven human motion generation with diffusion model. *arXiv preprint arXiv:2208.15001*, 2022. 3, 4

[75] S. Zhang, Y. Zhang, Q. Ma, M. J. Black, and S. Tang. Place: Proximity learning of articulation and contact in 3d environments. In *2020 International Conference on 3D Vision (3DV)*, pages 642–651, Los Alamitos, CA, USA, nov 2020. IEEE Computer Society. 3, 6

[76] Xiaohan Zhang, Bharat Lal Bhatnagar, Sebastian Starke, Vladimir Guzov, and Gerard Pons-Moll. Couch: Towards controllable human-chair interactions. In *European Conference on Computer Vision (ECCV)*. Springer, October 2022. 1, 2, 4, 5, 6, 7, 8

[77] Yan Zhang, Michael J Black, and Siyu Tang. We are more than our joints: Predicting how 3d bodies move. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3372–3382, 2021. 6

[78] Yan Zhang, Mohamed Hassan, Heiko Neumann, Michael J. Black, and Siyu Tang. Generating 3d people in scenes without people. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 3, 6

[79] Yan Zhang and Siyu Tang. The wanderings of odysseus in 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20481–20491, 2022. 2

[80] Yang Zheng, Yanchao Yang, Kaichun Mo, Jiaman Li, Tao Yu, Yebin Liu, Karen Liu, and Leonidas J Guibas. Gimo: Gaze-informed human motion prediction in context. *arXiv preprint arXiv:2204.09443*, 2022. 2

[81] Yi Zhou, Zimo Li, Shuangjiu Xiao, Chong He, Zeng Huang, and Hao Li. Auto-conditioned recurrent networks for extended complex human motion synthesis. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. 2

[82] Zixiang Zhou and Baoyuan Wang. UDE: A Unified Driving Engine for Human Motion Generation. *arXiv e-prints*, page arXiv:2211.16016, Nov. 2022. 3