Shape Conditioned Human Motion Generation with Diffusion Model

Kebing Xue, Hyewon Seo

ICube laboratory, CNRS–University of Strasbourg, France

Abstract

Human motion synthesis is an important task in computer graphics and computer vision. While focusing on various conditioning signals such as text, action class, or audio to guide the generation process, most existing methods utilize skeleton-based pose representation, requiring additional skinning to produce renderable meshes. Given that human motion is a complex interplay of bones, joints, and muscles, considering solely the skeleton for generation may neglect their inherent interdependency, which can limit the variability and precision of the generated results. To address this issue, we propose a Shape-conditioned Motion Diffusion model (SMD), which enables the generation of motion sequences directly in mesh format, conditioned on a specified target mesh. In SMD, the input meshes are transformed into spectral coefficients using graph Laplacian, to efficiently represent meshes. Subsequently, we propose a Spectral-Temporal Autoencoder (STAE) to leverage cross-temporal dependencies within the spectral domain. Extensive experimental evaluations show that SMD not only produces vivid and realistic motions but also achieves competitive performance in text-to-motion and action-to-motion tasks when compared to state-of-the-art methods.

1 Introduction

Human motion generation aims to generate realistic and lifelike human movements, which is essential for applications involving virtual characters, such as film production or virtual reality experiences. However, this task poses significant challenges. Firstly, the array of possible motions and body dynamics is extensive and diverse. Secondly, the availability of datasets is limited due to the necessity of professional equipment for motion capture. Compounding this challenge is the inherent variability in human motion, even within a single motion class, such as walking or running, the execution can vary substantially from person to person due to subtle anatomical variation and shape differences. Recently, many studies have been concentrating on this objective, among which the diffusion model[11, 24] has been prominently deployed as a backbone. These models can perform motion generation conditioned by various types of signals, such as motion classes, text descriptions[33, 39], music[4], and trajectories[14]. All these methods utilize skeleton-based representation as the primary means of depicting body pose and characteristics. This representation condenses human motion into a sequence of joint positions, rotations, and velocities across various body parts [8], which provides a simplified and efficient way to represent complex human motion. However, it lacks the fine details captured by mesh-based representations, such as muscle deformations. Since human motion is driven by both bones and muscles, generating human motion without considering body shape disregards the natural interdependency between these two elements. This limitation may hinder the model's generalization ability and restrict the variation and precision of the results. Moreover, to generate motion for a 3D character, these methods typically require an additional skinning process to attach a renderable skin to an underlying articulated skeleton, often followed by postprocessing steps such as retargeting. Specifically, the body shape parameters of SMPL model[21] are regressed to the generated skeleton, then combined with the skeleton pose to produce a mesh sequence. Setting aside the issue of additional computational cost, the parameter-based regression process makes it difficult to precisely control the desired body shape. These limitations consequently constrain the applicability of these models in 3D animation.

To address the aforementioned challenges, we introduce SMD, a shape-conditioned human motion diffusion framework. SMD is designed to generate realistic human motion directly in renderable mesh format, covering varied motion classes (Figure 1). Notably, the generated motion is conditioned by a given body mesh, ensuring consistency not only in identity shape throughout the motion but also in the inherent characteristics of the motion itself. Inspired by SAE[18] for effective information encoding from meshes into deep learning models, we also employ the Laplacian representation to compress human body meshes in the spectral domain. This approach circumvents the great computational

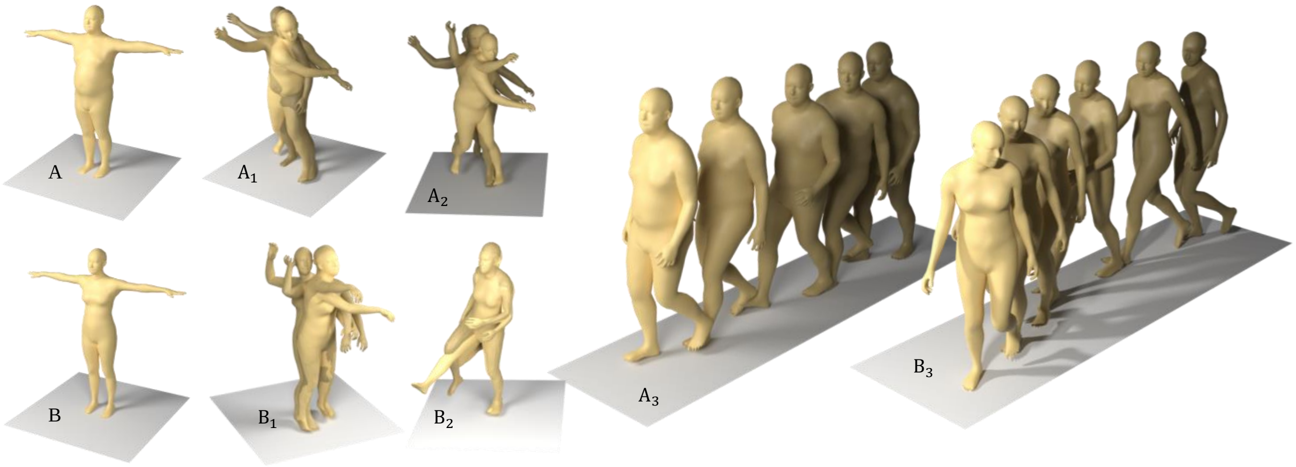

Figure 1: Shape conditioned generation results of SMD. A_1, A_2, and A_3 are based on A. B_1, B_2, and B_3 are generated when B is the target mesh.

and storage costs associated with directly generating the triangle meshes using deep learning models. To effectively capture temporal dependencies between frames in a motion, we employ a transformer backbone[34]. This architecture can also aid in integrating information extracted from the conditioning signal into motion features. Furthermore, SMD enables the bi-modal conditioning of the generation process using both natural language and body shapes. We adopt a classifier-free approach[13], which facilitates a balance between variability and fidelity. This approach eliminates the need for training auxiliary classifiers by sampling both conditional and unconditional instances from the same model. Additionally, to make the shape conditioning process more user-friendly, SMD accepts meshes in any pose as the target mesh, with the help of a shape encoder pre-trained with a contrastive loss[10], which is capable of extracting shape identity features from meshes in arbitrary poses. Integrating it into our framework allows for precise control of the generated body shape.

Finally, to validate our model, we perform extensive qualitative and quantitative experiments on BABEL[28] and HumanML3d[8], both of which are the annotations of AMASS[23] dataset. The results show that our proposed SMD achieves competitive performance compared with state-of-the-art works on several tasks (text-to-motion and action-to-motion). We also show that our approach reliably generates high-quality and vivid human motion while maintaining the body shape consistent with a given target mesh.

We summarize our contributions as follows:

1. We introduce a novel framework capable of directly generating human motion in the form of mesh sequences;
2. We propose to leverage human body mesh as a conditioning signal, and generate realistic human motion based on it;
3. We evaluate our approach extensively across multiple tasks, demonstrating competitive performance in each scenario.

2 Related Work

Human motion synthesis has become a long-standing research topic[2]. By applying deep learning methods to this field, researchers first perform unconstrained generation, whose target is to generate vivid and realistic human motion sequences without any constraints[6, 19, 25]. With the advancement of deep learning methods and increasing demands, some works tried to guide motion generation with different action classes. Action2Motion[9] uses conditional temporal VAE to capture frame-level features. ACTOR[26] adopts conditional transformer VAE to better analyze temporal dependency between frames. ActFormer[36] uses a GAN-based transformer to realize multi-person motion generation. With the development of pre-trained large language models such as CLIP[30] and RoBERTa[20], they provided a bridge between human intention and machine understanding and showed new possibilities to control generative deep learning models. Text-conditioned motion generation thus begins to dominate research frontiers[33, 15, 39, 4].

Diffusion models have achieved significant success in various tasks such as image denoising, inpainting, super-resolution, and image generation. The work of Sohl-Dickstein *et al.*[31] first introduce this theory from non-equilibrium statistical physics into the deep learning field. Based on it, Ho *et al.* [12] design DDPM for high-quality image synthesis, Others try to augment the efficacy and fidelity of diffusion model[32, 24]. Considering the strong generalize ability of the diffusion model, Dhariwal and Nichol[5] further apply it in class-conditioned image generation tasks, by designing classifier guidance, the diffusion model first surpasses GAN[7] on this task. To avoid training auxiliary classifiers, Ho and Salimans proposed a classifier-free guidance[13]. The significant success in applying the diffusion model in the field of image generation demonstrates its remarkable generalization ability and encourages researchers to apply it in other tasks such as audio synthesis[27, 16] and point cloud generation[22].

Diffusion-based motion generation has thus drawn considerable interest from researchers. MDM[33], Flame[15], MotionDiffuse[39] first try to apply the diffusion model in human motion generation tasks, they use pre-trained large language models to encode commands described in natural language, then guide the generation process based on these feature of text to generate motions follow description. Some of them have also tried motion inpainting. Mofusion[4] realizes a music-conditioned generation. Make-An-Animation[1] extracts a large-scale pseudo-pose dataset from image-text datasets, enhancing the stability of the generation process through pre-training the model with this dataset. MLD[3], MultiAct[17], EMS[29] and Fg-T2M[35] focus on generating a longer motion sequence with finer-grained text descriptions. AttT2M[40] propose to use a multi-perspective attention mechanism to extract dependency between body parts and different frames to augment the generation quality. Interdiff[37] uses the diffusion model to simulate human-object interactions. GMD[14] realize a motion generation along given trajectories or avoid certain obstacles. PhysDiff[38] proposes to apply motion correction based on a physic simulator iteratively during denoising steps of reverse diffusion to generate physically plausible motions.

3 Method

We introduce SMD, a diffusion-based framework for shape-conditioned human motion generation.

3.1 Method Overview

An overview of our framework is shown in Figure 2, given a conditioning target body mesh c_{shape} in an arbitrary pose and a conditioning signal $c_{dynamic}$ in the form of text or action class, our objective is to generate a motion sequence $M^{1:F}$ of length F that corresponds to the description and under the same body shape as the target mesh. In our work, we use body meshes that are compatible with SMPL[21]. For training, starting from a motion sequence $M^{1:F} \in \mathbb{R}^{F\times N\times 3}$ consisting of F triangle meshes each with N vertices, we first separate the global position $P^{1:F} \in \mathbb{R}^{F\times 3}$ and rotation $R^{1:F} \in \mathbb{R}^{F\times 3}$ of the root joint for each frame. This will ensure that the meshes are centered at the origin point and facing in the same direction. Next, the centered meshes are transformed into spectral coefficients $C^{1:F} \in \mathbb{R}^{F\times k\times 3}$ by applying the graph Fourier transform, where k is the number of coefficients. These coefficients are further normalized by subtracting the mean and dividing by the standard deviation, computed across the entirety of the training dataset. The separated translation and rotation of the root joint are concatenated with the normalized spectral coefficients to form $x_0^{1:F} \in \mathbb{R}^{F\times(k+2)\times 3}$, which is used to train the diffusion model, along with features extracted from the condition signals.

The conditioning signals consist of two kinds: one is utilized to constrain the body shape, while the other guides the dynamics of the generated motion. Specifically, given a target mesh c_{shape} and an action class or textual description $c_{dynamic}$, our SMD generates a deforming body mesh exhibiting the desired motion and that maintains the same body identity as the target. While our current model is designed to support meshes in SMPL topology, any arbitrary topology can be made SMPL-compatible through the use of surface correspondence or registration techniques.

During inference, we sample $x_T^{1:F}$ from a normal distribution $N(0, I)$ and iteratively denoise it into $\hat{x}_0^{1:F}$ guided by $c_{dynamic}$ and c_{shape} with the trained STAE. Then a reverse graph Fourier transform is applied to the generated coefficients to cover them back into meshes. Finally, spatial position and rotation are injected into them to build a motion in 3D.

3.2 Mesh Spectral Representation

We use graph Laplacian to transform meshes into spectral coefficients and use them as the model's input for training[18]. A human body triangle mesh M can be considered as a graph with N nodes. The position of vertices can be expressed as $f(i) = (x_i, y_i, z_i), i \in [1, N]$ where x, y, z correspond to the 3D coordinates of vertices. The adjacency matrix A of this graph is a square matrix whose elements indicate whether pairs of vertices are adjacent or not and its degree matrix D is a diagonal matrix whose elements correspond to the number of edges attached to each vertex. The Graph Laplacian L is defined as $L = D - A$. Since all meshes we use share the same topology introduced in SMPL[21], their graph Laplacians are also the same.

Let $\{\lambda_l\}_{l=0,1,\ldots,F-1}$ and $\{\mathbf{u}_l\}_{l=0,1,\ldots,F-1}$ be the eigenvalues and eigenvectors of L satisfying $L\mathbf{u}_l = \lambda_l \mathbf{u}_l$, similar to the classical Fourier transform, we can represent M in spectral coefficients by applying the graph Fourier transform $\mathscr{F}$:

$$\mathscr{F}(\lambda_l) = \langle \mathbf{f}, \mathbf{u}_l \rangle = \sum_{i=1}^{N} f(i) u_l^*(i), \tag{1}$$

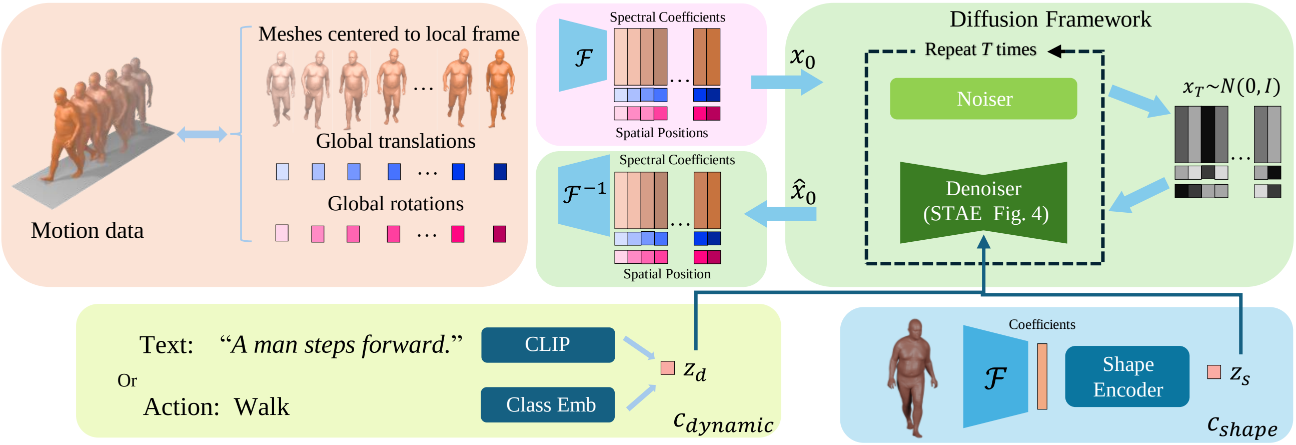

Figure 2: Method overview: 1) From each mesh in the motion data, the vertex coordinates in the local frame are transformed into spectral coefficients by using a graph Fourier transformation; 2) The coefficients together with the rotations and translations are used to train the diffusion model; 3) After training, Spectral-Temporal Autoencoder (STAE) generates motion from a random noise, conditioned on a conditioning signal z_d and a target mesh embedding z_s, by denoising it iteratively.

and the reverse graph Fourier transform can be expressed as:

$$f(i) = \sum_{l=0}^{N-1} \mathscr{F}(\lambda_l) u_l(i). \tag{2}$$

Since the graph Laplacian L is a real symmetric matrix, its eigenvalues are all non-negative, we sort them in a decreasing order. Eigenvectors associated with lower eigenvalues vary slowly across the graph, while eigenvectors associated with higher eigenvalues are more likely to have different values on adjacent nodes. And the most important part of information about the human body is contained in the low frequencies.

We choose k eigenvectors where $k \ll n$ corresponds to k lowest eigenvalues to calculate the coefficients, this can reduce the size of input training data, and further reduce the computational complexity of our method. Of course, this process will lose some information, but as shown in Figure 3, by choosing a proper k, we achieve a high level of representation accuracy while minimizing information loss to a great extent.

Compared to using graph convolutional networks which also exploit the information of graph in spectral domain by convolving along the graph topology, this approach avoids downsampling/upsampling, which can potentially hinder the exploitation of spectral information to its fullest extent[18].

3.3 Motion diffusion model

The overview of our diffusion model is shown in Figure 2. It is based on the diffusion framework, in which we design a Spatial-Temporal Autoencoder as the denoiser for

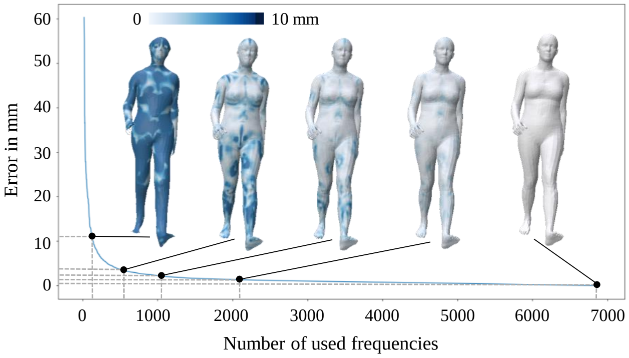

Figure 3: Reconstruction error as a function of the number of used eigenvectors. SMPL meshes are compressed by applying graph Fourier transform/inverse transform using a certain number of eigenvectors, from left to right we try 128,512,1024,2048 and 6890 eigenvectors, the color corresponds to the distance between the compressed mesh and original mesh.

Algorithm 1 Sampling

```
Input: conditions $C$, well trained $STAE$
$x_t \sim N(\mathbf{0}, \mathbf{I})$
for t=T,...,1 do
    $\mathbf{z} \sim N(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = 0$
    $x_{t-1} = \tilde{\mu}_t(x_t, STAE(x_t, t, C)) + \delta_t \mathbf{z}$
end for
return$x_0$
```

the reverse process.

Diffusion model is a deep generative model that consists of two stages, a forward diffusion stage which acts as a Markov noising process, and a backward denoising process based on a deep learning model. In the forward process, given an uncorrupted training sample $x^{1:F}$, the noise version $x_1^{1:F}$, $x_2^{1:F}$...,$x_t^{1:F}$ at diffusion step t can be driving from the following approximate posterior:

$$q(x_T^{1:F}) = \prod_{t=1}^{T} q(x_t^{1:F}|x_{t-1}^{1:F}), \forall t \in 1, ..., T, \qquad (3)$$

$$q(x_t^{1:F}|x_{t-1}^{1:F}) = N(x_t^{1:F}; \sqrt{1-\beta_t} x_{t-1}^{1:F}, \beta_t \mathbf{I}), \qquad (4)$$

where T corresponds to the predefined maximum diffusion step and $\beta_1, ...\beta_t \in [0, 1)$ are the predefined noise schedule. For simplicity, we use x_t to denote the full sequence $x_t^{1:F}$ at diffusion step t and use x_0 for the uncorrupted input. An important property of this process is that a sample at any step t is driven directly from the input $x^{1:N}$ with this formula[11]:

$$q(x_t|x_0) = N(x_t; \sqrt{\hat{\alpha}_t} x_0, (1-\hat{\alpha}_t)\mathbf{I}), \qquad (5)$$

where $\hat{\alpha}_t = \prod_{i=1}^{t} \alpha_t$ and $\alpha_t = 1 - \beta_t$. This allows us to easily train a denoising model for arbitrary steps by defining a noise schedule $\{\beta_t\}_{t=1,...,t}$. Following the setting in this paper [24], we adopt the cosine noise schedule in terms of $\hat{\alpha}_t$ as:

$$\hat{\alpha_t} = \frac{f(t)}{f(0)}, f(t) = cos(\frac{t/T+s}{1+s} \cdot \frac{\pi}{2})^2. \qquad (6)$$

When t is closer to T, $\hat{\beta}_t$ will be close to 1, in which case we can approximate $p(x_T) \sim N(0, I)$.

The goal of the conditioned motion diffusion model is to generate a sequence of human motion under certain constraints, which is further abstracted in this formula $p(x_0|C)$ in which C corresponds to condition signals. This is realized by using a deep learning model whose parameter is represented as θ to denoise an initial state x_T sampled from a normal distribution into an uncorrupted sample by sampling from these distributions iteratively:

$$p_\theta(x_0|C) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t, C). \qquad (7)$$

Some works applied the diffusion model to the image generation task[11, 5], their model is trained to predict the noise during training. In our context, predicting the input x_0 during training will allow us to apply geometric losses more easily, making the generated motion more stable. Our training objective for the diffusion model can be written as the following expected value:

$$L_{diff} = E_{x_0 \sim q(x_0|C), t \sim [1,T]}[||x_0 - STAE(x_t, t, C)||^2]. \qquad (8)$$

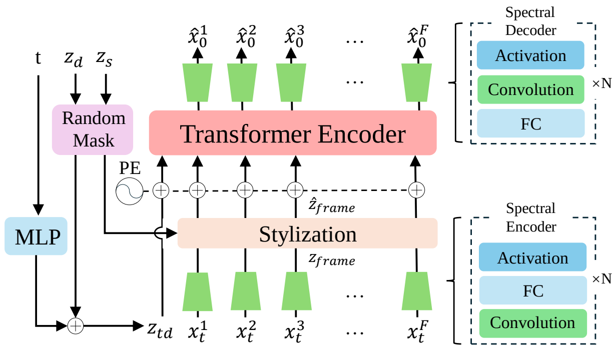

Figure 4: Overview of our Spectral-Temporal Autoencoder (STAE).

The probability distribution for the denoise process in Equation 7 can be simulated by calculating its mean and variance by the deep learning model:

$$p_\theta(x_{t-1}|x_t, C) = N(x_t - 1; \mu_\theta(x_t, t, C), \Sigma_\theta(x_t, t, C)) \qquad (9)$$

Following the setting in DDPM [11], we fix the variance term $\Sigma_\theta(x_t, t, C)$ to $\delta_t^2 \mathbf{I}$ where $\delta_t^2 = \hat{\beta}_t = \frac{1-\hat{\alpha}_{t-1}}{1-\hat{\alpha}_t}$ for a more stable training. To transform the predicted $\hat{x}_0$ to $\mu_\theta(x_t, t, C)$, we need to consider the forward process posteriors which are tractable given x_0 [11]:

$$q(x_{t-1}|x_t, x_0) = N(x_{t-1}; \tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t \mathbf{I}), \qquad (10)$$

$$\tilde{\mu}_t(x_t, x_0) = \frac{\sqrt{\hat{\alpha}_{t-1}}\beta_t}{1-\hat{\alpha}_t} x_0 + \frac{\sqrt{\alpha_t}(1-\hat{\alpha}_{t-1})}{1-\hat{\alpha}_t} x_t. \qquad (11)$$

We thus replace the x_0 with the predicted $\hat{x}_0 = STAE(x_t, t, C)$ in $\tilde{\mu}_t(x_t, x_0)$ to approximate the μ_θ in Eq. 9 . Sampling $x_{t-1} \sim p_\theta(x_{t-1}|x_t, C)$ is thus performed via a parametrization trick by calculating $x_{t-1} = \tilde{\mu}_t(x_t, \hat{x}_0) + \delta_t \mathbf{z}$, where $\mathbf{z} \sim N(\mathbf{0}, \mathbf{I})$, as shown in Algorithm 1.

Spectral-Temporal Autoencoder consists of a spectral encoder, a spectral decoder, and a Transformer encoder, as illustrated in Figure 4. Given that the conditioning signals come from different modalities, with the dynamic conditions being either an action class or a text description, and the shape condition being represented by a triangle mesh, we adopt distinct strategies to integrate them. Note that this is different from other approaches that sum different conditioning signals into one before sending it to the model. The embedding of diffusion step t and features extracted from dynamic conditions z_d are summed together to z_{td}. It is concatenated to the first place of the features extracted from spectral coefficients with the spectral encoder. The subsequent transformer encoder allows it to have different influences on different temporal parts of motion. Different from z_d, the feature z_s extracted from target meshes is expected to have a consistent influence on all frames, so we adopt a stylization block[39] to inject them into motion features. The stylization block consists of three dense layers ψ_b, ψ_w, and ϕ, they map the original feature z_{frame} to $\hat{z}_{frame}$ with $\hat{z} = z \circ \psi_w(\phi(z_s)) + \psi_b(\phi(z_s))$.

A transformer encoder is also used to further exploit the temporal dependency between the features of different frames, its self-attention mechanism permits the calculation of features in each frame based on all other frames, which implicitly ensures the continuity and consistency of the whole motion. The output will be decoded using a spectral decoder to match the dimensions of the input. The spectral encoder and decoder are composed of a sequence of convolutional blocks. Each block comprises a convolutional layer, a fully connected layer, and an activation layer, as shown in Figure 4.

3.4 Conditioned generation

Dynamic conditions can be an action class or a text description. We adopt CLIP[30] as a text encoder to embed the text prompt and use a trainable tensor as embeddings for different action classes.

Target mesh is embedded by a shape encoder capable of embedding the input mesh into features that solely include the identity shape of the body mesh independently of poses. This encoder is trained within a *shape autoencoder* using the paradigm of contrastive learning. It shares the same architecture as the STAE, except that the Transformer Encoder is omitted. Given meshes of different characters in arbitrary poses, it is trained to predict the mesh in canonical T-pose. The contrastive learning loss is adopted to better build a uniform representation space. T-pose meshes $M_i^{t\text{-}in}$ and meshes in arbitrary poses $M_i^{a\text{-}in}$ where $i \in [1, N]$ corresponds to N different chosen characters are sent to the model. The loss function is:

$$L_{ShapeEmb} = L_{mesh} + L_{contrast}, \tag{12}$$

$$\begin{aligned} L_{mesh} = \frac{1}{2N}(&\sum_{i=1}^{N} ||M_i^{t\text{-}out} - M_i^{t\text{-}in}||^2 \\ &+ \sum_{i=1}^{N} ||M_i^{a\text{-}out} - M_i^{t\text{-}in}||^2), \end{aligned} \tag{13}$$

$$L_{contrast} = -\sum_{i=1}^{N} \frac{1}{N} log \frac{exp(sim(z_i^t, z_i^a)/\tau)}{\sum_{k=1,k\neq i}^{M} exp(sim(z_i^t, z_k)/\tau)}, \tag{14}$$

where $M_i^{t\text{-}out}$,$M_i^{a\text{-}out}$ are predicted meshes and z_i^t,z_i^a are latent features. Minimizing $L_{contrast}$ requires maximizing the similarity between z_i^t and z_i^a which means attracting the features associated with the same character close to each other. At the same time, it pushes the features extracted from different characters away by minimizing the similarity between z_i^t and z_k. The L_{mesh} encourages the model to focus on shape information when facing meshes in different poses. These two loss terms will ensure we build a pose-independent representation space for identity shapes. Note that this shape encoder is also used to embed the target mesh c_{shape} into z_s.

Classifier free guidance is an approach to achieve highly precise conditioned sampling for a diffusion model without the need for training auxiliary models. Specifically, as shown in Figure 4, we randomly mask a certain proportion of the condition signals to $\emptyset$ to simulate the unconditioned generation. Then, during inference, we can manipulate a ratio s between the conditionally generated result and the unconditionally generated result to trade off diversity and fidelity, formulate as:

$$\begin{aligned} STAE_s(x_t, t, C) = STAE(x, t, \emptyset) + \\ s_d \cdot (STAE(x_t, t, c_{dynamic}, \emptyset_{shape}) \\ - STAE(x_t, t, \emptyset)) + \\ s_s \cdot (STAE(x_t, t, \emptyset_{dynamic}, c_{shape}) \\ - STAE(x_t, t, \emptyset)). \end{aligned} \tag{15}$$

3.5 Losses

In our work, for each denoise step, we predict the uncorrupted signal $\hat{x}_0$, which is composed of rotations $R^{1:F}$, translations $P^{1:N}$, and coefficients $C^{1:F}$. As discussed in Section 3.2, coefficients associated with lower frequencies inherently encapsulate richer information, often characterized by larger values. To amplify our model's focus on these coefficients we multiply the coefficient loss by their respective variances. The coefficient loss is expressed as:

$$L_{coef} = \frac{1}{F} Var(\tilde{C}) \sum_{i=1}^{F} ||C_0^i - \hat{C}_0^i||^2, \tag{16}$$

where $Var(\tilde{C}) \in \mathbb{R}^{k\times 3}$ is the variance of coefficients calculated on the whole training set. Since the coefficients can be further reversed into 3D mesh with pre-calculated eigenvectors, our losses will also be applied to spatial domains:

$$L_{mesh} = \frac{1}{F} \sum_{i=1}^{F} ||M_0^i - \hat{M}_0^i||^2; \tag{17}$$

$$L_{pos} = \frac{1}{F}(\sum_{i=1}^{F} ||R_0^i - \hat{R}_0^i||^2 + \sum_{i=1}^{F} ||P_0^i - \hat{P}_0^i||^2). \quad (18)$$

These three terms keep the coefficients, mesh, and position consistent with the input. To ensure the smoothness of the moving trajectory, we further apply a residual loss to control the translation:

$$L_{res\_P} = \frac{1}{F-1}\sum_{i=2}^{F} ||(P_0^i - P_0^{i-1}) - (\hat{P}_0^i - \hat{P}_0^{i-1})||^2. \quad (19)$$

Overall, our training loss is:

$$L = \lambda_d \cdot L_{diff} + \lambda_c \cdot L_{coef} + \lambda_m \cdot L_{mesh} + \lambda_p \cdot L_{pos} + \lambda_r \cdot L_{res\_P}. \quad (20)$$

4 Experiments

To evaluate the effectiveness of our method, we utilize SMD in two typical settings: text-to-motion and action-to-motion generation, each combined with a conditioning target mesh. Furthermore, since our model supports shape-conditioned generation, we evaluate the shape consistency between generated motions and the given target mesh.

4.1 Settings

Datasets. We train our model by using HumanML3D[8] dataset, which contains 32,357 per-motion text annotations on 10,524 motion sequences from the AMASS dataset[23]. AMASS is chosen because it provides mesh data for each motion, which is required by our model. We also augment the data by mirroring all sequences about the sagittal plane, following the settings in HumanML3D [8]. Their test set for the text-to-motion generation is used for the evaluation. For the action-to-motion generation, we use the BABEL[28] dataset, which provides per-motion action labels for the AMASS[23] dataset. We chose 8 action classes, resulting in 10,688 motions.

Evaluation metrics. Following the common settings for text-to-motion generation, we evaluate our SMD on text-to-motion tasks using Fréchet Inception Distance(FID), R-Precision, and diversity. FID quantifies the realism and diversity of generated motions by comparing their distribution in latent space with the ground truth distributions using a pre-trained motion encoder. R-precision measures the relevance of the generated motions to the input prompts, and diversity quantifies the variability of the generated motions. While our model generates motions in the form of mesh sequences, for comparisons with other works that are mostly skeleton-based, we convert our model's generated mesh sequences into skeletons. This is achieved by using a regression matrix introduced in SMPL[21] to regress joints from the mesh. The aforementioned metrics can then be computed on the regressed skeleton-based motions. Furthermore, SMD is trained with mesh geometry loss, which implicitly empowers it to better account for the mesh-environment interaction, and generate physically plausible motions. To evaluate this ability, we adopt three additional metrics: *Penetration* measures ground penetration by the body, *floating* measures the extent of mesh floating above the floor, and *skating* measures foot sliding effect, all in millimeters (*mm*). For the action-to-motion generation, *accuracy* replaces the R-Precision by calculating the classification accuracy of a pre-trained classifier on the generated motion. To measure the performance of shape-conditioned generation, we measure the *consistency* of identity shape between the generated motion and the conditioning target mesh by computing the positional error between corresponding vertices of their respective pose-normalized meshes.

Table 1: Hyperparameters.

Configuration	Value
k	1024
Optimizer	Adam
Learning rate	1e-4
Number of multi-head attention	8
Latent dimension	256
Dropout	0.2
$\lambda_d, \lambda_c, \lambda_m$	1, 1, 1
λ_p	50
λ_r	1e4
k: Number of frequencies used	1024
Batchsize	32
s_d	0.85
s_s	0.7

Implementation details. The main hyperparameter values of the SMD are shown in Table 1. It took less than three days of training on a single NVIDIA GeForce RTX 3080 GPU.

Table 2: Action-to-motion results on BABEL[28] dataset. ↑ means the larger values are better, and ↓ indicates the smaller values are better.

Method	FID↓	Accuracy↑
Real(Skeleton)	0.001	0.984
Real(Mesh)	0.001	0.997
MDM[33]	0.173	0.979
SMD(Skeleton)	0.251	0.974
SMD(Mesh)	**0.161**	**0.991**

Table 3: Text-to-motion results on HumanML3D[8] dataset. → means the results are better if the value is closer to the real distribution.

Method	FID↓	R-Precision↑	Diversity→	Penetrate↓	Float↓	Skate↓
Real	0.002	0.797	9.50	5.965	2.354	0.929
T2M[8]	1.067	0.740	9.188	11.897	7.779	2.908
MDM[33]	0.489	0.707	9.45	11.291	18.876	**1.406**
MotionDiffuse[39]	0.630	0.782	9.410	20.278	6.450	3.925
Fg-T2M[35]	0.243	**0.783**	9.278	-	-	-
SMD(Ours)	**0.214**	0.737	**9.472**	**8.741**	**5.854**	2.576

4.2 Shape conditioned generation

Our model can generate motion conditioned on a target mesh. We assess the performance of this task by measuring the shape consistency within different frames of the same generated motion and between the generated and target meshes. As introduced in Section 3.4, our shape embedder is trained to transform a mesh in an arbitrary pose into the canonical T-pose. Utilizing this model, we convert the meshes in the generated motion into T-poses and calculate the shape error based on the Euclidean distance between corresponding vertices in these meshes. We randomly select 20 characters, with each character represented by five meshes in various poses, and use these to generate 1,000 random sequences. The average inner-sequence identity shape error measures at 1.19 ± 0.68 *mm*, while the discrepancy between the target mesh and the generated motion stands at 2.34 ± 1.3 *mm*. In comparison, the compression error arising from the limited number of eigenvectors used for the graph Fourier transform is 0.75 ± 0.58 *mm*. The error distribution is shown in Figure 5: The inter-sequence identity shape errors concentrate on foot and hand while the errors with respect to the target follow a similar distribution to the reconstruction error shown in Figure 3. These quantitative results show our model's proficiency in generating motion aligned with target meshes and maintaining shape consistency across frames. We provide a video demo for the visual inspection.

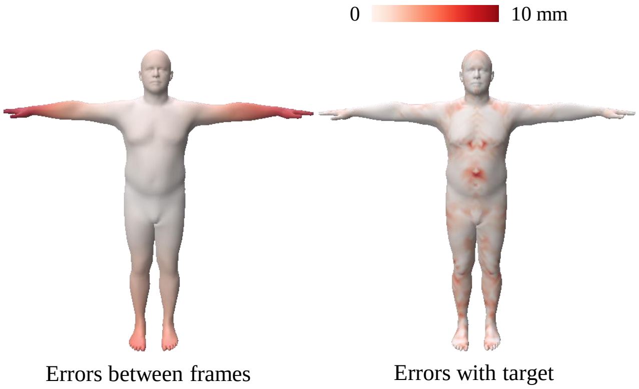

Figure 5: Average errors in shape consistency within the motion (left) and in comparison with the target (right).

4.3 Text-to-motion

We compared our method with several state-of-the-art models that can generate motions given text descriptions, including T2M [8], MDM[33], MotionDiffuse[39], and Fg-T2M[35].

As shown in Table 3, SMD outperforms other diffusion-based works in terms of FID and diversity which are indicators of motion quality, while maintaining a similar R-Precision score. Note that since our model takes the mesh as input, it can implicitly account for the mesh-ground interaction, even without explicit constraints to address this aspect. This is evidenced by its improved performance compared to other skeleton-based methods on physics-based metrics such as penetration and float. Examples of our generated motions are provided in the demo video.

4.4 Action-to-motion

We conduct this experiment for both mesh-based and skeleton-based motions, employing the settings described above. The results, summarized in Table 2, reveal that the mesh-based method exhibits a higher accuracy upper bound, indicating its superior inherent expressive capability. This is perhaps one reason why our model outperforms the state-of-the-art diffusion-based human motion generation model. Please refer to our demo video for examples of our generated motions.

5 Conclusion

We proposed SMD, a diffusion-based human motion generation model that can generate motion in the format of triangle meshes conditioned on the given text prompts and a given target mesh. We chose the spectral domain to fully exploit the meshes with fewer resource costs. Based on a Spectral-Temporal Autoencoder, our model shows great expressive ability and stability when performing the shape-conditioned generation. We expect that our method can streamline the character animation pipeline and provide new possibilities to use synthetic data to expand the human motion dataset.

References

[1] S. Azadi, A. Shah, T. Hayes, D. Parikh, and S. Gupta. Make-an-animation: Large-scale text-conditional 3d human motion generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15039–15048, 2023.

[2] R. Bowden. Learning statistical models of human motion. In *IEEE Workshop on Human Modeling, Analysis and Synthesis, CVPR*, volume 2000, 2000.

[3] X. Chen, B. Jiang, W. Liu, Z. Huang, B. Fu, T. Chen, and G. Yu. Executing your commands via motion diffusion in latent space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18000–18010, 2023.

[4] R. Dabral, M. H. Mughal, V. Golyanik, and C. Theobalt. Mofusion: A framework for denoising-diffusion-based motion synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9760–9770, 2023.

[5] P. Dhariwal and A. Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[6] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik. Recurrent network models for human dynamics. In *Proceedings of the IEEE international conference on computer vision*, pages 4346–4354, 2015.

[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[8] C. Guo, S. Zou, X. Zuo, S. Wang, W. Ji, X. Li, and L. Cheng. Generating diverse and natural 3d human motions from text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5152–5161, 2022.

[9] C. Guo, X. Zuo, S. Wang, S. Zou, Q. Sun, A. Deng, M. Gong, and L. Cheng. Action2motion: Conditioned generation of 3d human motions. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2021–2029, 2020.

[10] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[11] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[12] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[13] J. Ho and T. Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[14] K. Karunratanakul, K. Preechakul, S. Suwajanakorn, and S. Tang. Guided motion diffusion for controllable human motion synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2151–2162, 2023.

[15] J. Kim, J. Kim, and S. Choi. Flame: Free-form language-based motion synthesis & editing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 8255–8263, 2023.

[16] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020.

[17] T. Lee, G. Moon, and K. M. Lee. Multiact: Long-term 3d human motion generation from multiple action labels. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 1231–1239, 2023.

[18] C. Lemeunier, F. Denis, G. Lavoué, and F. Dupont. Representation learning of 3d meshes using an autoencoder in the spectral domain. *Computers & Graphics*, 107:131–143, 2022.

[19] Z. Li, Y. Zhou, S. Xiao, C. He, Z. Huang, and H. Li. Auto-conditioned recurrent networks for extended complex human motion synthesis. *arXiv preprint arXiv:1707.05363*, 2017.

[20] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[21] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, Oct. 2015.

[22] S. Luo and W. Hu. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2837–2845, 2021.

[23] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black. AMASS: Archive of motion capture as surface shapes. In *International Conference on Computer Vision*, pages 5442–5451, Oct. 2019.

[24] A. Q. Nichol and P. Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR, 2021.

[25] D. Pavllo, D. Grangier, and M. Auli. Quaternet: A quaternion-based recurrent model for human motion. *arXiv preprint arXiv:1805.06485*, 2018.

[26] M. Petrovich, M. J. Black, and G. Varol. Action-conditioned 3d human motion synthesis with transformer vae. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10985–10995, 2021.

[27] V. Popov, I. Vovk, V. Gogoryan, T. Sadekova, and M. Kudinov. Grad-tts: A diffusion probabilistic model for text-to-speech. In *International Conference on Machine Learning*, pages 8599–8608. PMLR, 2021.

[28] A. R. Punnakkal, A. Chandrasekaran, N. Athanasiou, A. Quiros-Ramirez, and M. J. Black. BABEL: Bodies, action and behavior with english labels. In *Proceedings IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 722–731, June 2021.

[29] Y. Qian, J. Urbanek, A. G. Hauptmann, and J. Won. Breaking the limits of text-conditioned 3d motion synthesis with elaborative descriptions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2306–2316, 2023.

[30] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[31] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.

[32] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[33] G. Tevet, S. Raab, B. Gordon, Y. Shafir, D. Cohen-or, and A. H. Bermano. Human motion diffusion model. In *The Eleventh International Conference on Learning Representations*, 2023.

[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[35] Y. Wang, Z. Leng, F. W. Li, S.-C. Wu, and X. Liang. Fg-t2m: Fine-grained text-driven human motion generation via diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22035–22044, 2023.

[36] L. Xu, Z. Song, D. Wang, J. Su, Z. Fang, C. Ding, W. Gan, Y. Yan, X. Jin, X. Yang, et al. Actformer: A gan-based transformer towards general action-conditioned 3d human motion generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2228–2238, 2023.

[37] S. Xu, Z. Li, Y.-X. Wang, and L.-Y. Gui. Interdiff: Generating 3d human-object interactions with physics-informed diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14928–14940, 2023.

[38] Y. Yuan, J. Song, U. Iqbal, A. Vahdat, and J. Kautz. Physdiff: Physics-guided human motion diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16010–16021, 2023.

[39] M. Zhang, Z. Cai, L. Pan, F. Hong, X. Guo, L. Yang, and Z. Liu. Motiondiffuse: Text-driven human motion generation with diffusion model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[40] C. Zhong, L. Hu, Z. Zhang, and S. Xia. Attt2m: Text-driven human motion generation with multi-perspective attention mechanism. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 509–519, 2023.