MotionDiffuse: Text-Driven Human Motion Generation With Diffusion Model

Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu

Abstract—Human motion modeling is important for many modern graphics applications, which typically require professional skills. In order to remove the skill barriers for laymen, recent motion generation methods can directly generate human motions conditioned on natural languages. However, it remains challenging to achieve diverse and fine-grained motion generation with various text inputs. To address this problem, we propose MotionDiffuse, one of the first diffusion model-based text-driven motion generation frameworks, which demonstrates several desired properties over existing methods. 1) Probabilistic Mapping. Instead of a deterministic language-motion mapping, MotionDiffuse generates motions through a series of denoising steps in which variations are injected. 2) Realistic Synthesis. MotionDiffuse excels at modeling complicated data distribution and generating vivid motion sequences. 3) Multi-Level Manipulation. MotionDiffuse responds to fine-grained instructions on body parts, and arbitrary-length motion synthesis with time-varied text prompts. Our experiments show MotionDiffuse outperforms existing SoTA methods by convincing margins on text-driven motion generation and action-conditioned motion generation. A qualitative analysis further demonstrates MotionDiffuse’s controllability for comprehensive motion generation.

Index Terms—Conditional motion generation, diffusion model, motion synthesis, text-driven generation.

I. INTRODUCTION

HUMAN motion modeling is a critical component of animating virtual characters to imitate vivid and rich human movements, which has been a vital topic for many applications, such as film-making, game development, and virtual YouTuber animation. To mimic human motions, virtual characters should be capable of moving around naturally, reacting to environmental stimuli, and meanwhile expressing sophisticated emotions. Despite decades of exciting technological breakthroughs, it requires sophisticated equipment (e.g., expensive motion capture systems) and domain experts to produce lively and authentic body movements. In order to remove skill prerequisites for layman users and potentially scale to the mass audience, it is vital to create a versatile human motion generation model that could produce diverse, easily manipulable motion sequences.

Various condition signals, including pre-defined motion categories [1], [2], [3], music pieces [4], [5], [6], [7], [8], and natural language [9], [10], [11], [12], have been leveraged in previous human motion generation methods. Among them, natural language is arguably the most user-friendly and convenient input format for motion sequence synthesis, and hence we focus on text-driven motion generation in this work. Recently, TEMOS [12] utilizes KIT Motion-Language MoCap dataset [13] to demonstrate accurate human motion synthesis. However, it does not support stylizing the generated motions and, therefore, could not achieve high diversity. MotionCLIP [14] could generate stylized motions, but it is still limited to short text inputs and fails to handle complicated motion descriptions. In addition, they [12], [14] typically only accept a single text prompt, which greatly limits users’ creativity.

To tackle the aforementioned challenges, we propose MotionDiffuse, a versatile and controllable motion generation framework that could generate diverse motions with comprehensive texts. Fig. 1 shows some examples generated by MotionDiffuse, including stylized action, synchronous movement and sequential motion. Inspired by the recent progress of the text-conditioned image generation [15], [16], [17], [18], we propose to incorporate Denoising Diffusion Probabilistic Models (DDPM) [19] into motion generation. Unlike classical DDPM which is only capable of fixed-size generation, we propose a Cross-Modality Linear Transformer to achieve motion synthesis with an arbitrary length depending on the motion duration. Instead of learning a direct mapping between the text space and the motion space [14], we propose to guide the generation pipeline with input texts softly, which could significantly increase the diversity of the generation results. To maintain the uncertainties in the denoising process, we process the noise terms conditioned on the input texts by several transformer decoder layers for each denoising step. In this way, the entire pipeline is no longer a deterministic mapping from text conditions, allowing for the generation of diverse motion sequences under the given conditions.

Manuscript received 11 January 2023; revised 6 October 2023; accepted 1 January 2024. Date of publication 29 January 2024; date of current version 7 May 2024. This work was supported by the Ministry of Education, Singapore, under its MOE AcRF Tier 2 under Grant MOE-T2EP20221-0012, in part by the NTU NAP, and under the RIE2020 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from the industry partner(s). Recommended for acceptance by K. G. Derpanis. (Mingyuan Zhang and Zhongang Cai contributed equally to this work.) (Corresponding author: Ziwei Liu.)

Mingyuan Zhang, Liang Pan, Fangzhou Hong, Xinying Guo, and Ziwei Liu are with S-Lab, Nanyang Technological University, Singapore 639798 (e-mail: mail@mnzhang718@gmail.com; liang.pan@ntu.edu.sg; fangzhouhong820@gmail.com; xgquete12@nus.edu.sg; zwliu.hs@gmail.com).

Zhongang Cai is with S-Lab, Nanyang Technological University, Singapore 639798, and also with SenseTime Research and Shanghai AI Laboratory, Shenzhen 518100, China (e-mail: caizhihong@sensetime.com).

Lei Yang is with SenseTime Research and Shanghai AI Laboratory, Shenzhen 518100, China (e-mail: jerry.yanglei@gmail.com).

Homepage: https://mingyuan-zhang.github.io/projects/MotionDiffuse.html

This article has supplementary downloadable material available at https://doi.org/10.1109/TPAMI.2024.3355414, provided by the authors.

Digital Object Identifier 10.1109/TPAMI.2024.3355414
Furthermore, MotionDiffuse can achieve body part-independent control with fine-grained texts. Specifically, to accommodate the human body structures, MotionDiffuse divides the whole-body motion into several near-independent parts (e.g., upper body and lower body). Based on the fine-grained body parts definition, we propose ‘noise interpolation’ to separately control different body parts while taking their correlations into consideration. Moreover, to synthesize arbitrary-length motion sequences, we propose a new denoising strategy to denoise several overlapped sequences simultaneously. Specifically, MotionDiffuse first gets results from each sequence independently and then mixes them with correction terms. Different from auto-regressive inference schemes that often require many long motion sequences for training, MotionDiffuse is capable of modeling the correlations between continuous actions without introducing additional training costs.

We perform both extensive qualitative experiments on popular benchmarks, and quantitative evaluation on comprehensive motion generation. First, we demonstrate significant improvements on text-driven motion generation over the current art on HumanML3D and KIT-ML. Second, to illustrate the superiority of MotionDiffuse, we transform action labels into textual descriptions and train our proposed text-driven motion generation pipeline on the action-conditioned motion generation task. Our approach outperforms all existing methods on both the HumanAct12 dataset and the UESTC dataset. Furthermore, we also demonstrate more possibilities of MotionDiffuse by conditioning the model on mixed control signals that allow body-part level manipulation and long motion generation.

In summary, our proposed MotionDiffuse has several desired properties over prior arts:

- **Probabilistic Mapping**: Benefiting from our new formulation of motion generation, where the DDPM is incorporated in, MotionDiffuse can be conditioned on text descriptions to generate motions in a probabilistic style, naturally leading to high diversity.

- **Realistic Synthesis**: The careful design of the architecture allows MotionDiffuse to synthesize high-fidelity motion sequences and achieves state-of-the-art on two conditional motion generation tasks.

- **Multi-Level Manipulation**: With the extended design, MotionDiffuse can handle fine-grained text descriptions that mobilize the entire body (e.g., ‘drinking water’ for upper limbs while ‘walking’ for lower limbs) and time-varied signals (e.g., ‘a person is walking’ followed by ‘is running’) without additional training.

II. RELATED WORK

A. Motion Generative Model

Motion Generation has been studied for decades. Some early works focus on unconditional motion generation [20], [21], [22]. While some other works try to predict future motions given an initial pose or a starter motion sequence [23], [24], [25]. Statistical models such as PCA [26], Motion Graph [27] are applied for these purposes.

With the rapid development of Deep Learning (DL) techniques, more generative architectures occur and flourish. Previous works can be broadly divided into four groups: 1) Variational Auto Encoder (VAE); 2) Generative Adversarial Networks (GAN); 3) Normalization Flow Network; 4) Implicit Neural Representations.

VAE [28] is one of the most commonly used generative models in motion synthesis. Yan et al. [29] and Aliakbarian et al. [30] regard the motion generation task as predicting a small future motion sequence with the given small current motion sequence. They use VAE to encode the pair of current sequence and future sequence and then reconstruct the future motion sequence. With the extended design, MotionDiffuse can be conditioned on text descriptions that allow body-part level manipulation and long motion generation.

In summary, our proposed MotionDiffuse has several desired properties over prior arts:
prior, which is used to justify the prediction quality. Harvey et al. [35] target solving the blurriness of the predicted motion in the Motion In-between task and propose two discriminators for both short-term critic and long-term critic. Wang et al. [36] build up a cyclic pipeline. With the help of a discriminator, the proposed pipeline can generate both class-specific and mixed-class motion sequences.

Normalization Flow Network [37] has a long history and has been studied extensively for image synthesis [38], [39]. This kind of architecture builds up a reversible neural network and will map the input data into a multi-dimensional Gaussian distribution. Hence, we can generate an initially random vector from this distribution and feed them into the reversed network to generate motion samples. Inspired by the success of GLOW [39], MoGlow [40] proposes an auto-regressive normalization network to model motion sequences. History features from an LSTM model [41] serve as the condition of the flow network, which predicts the next pose.

Recently, another generative model has attracted much attention with the considerable success achieved by NeRF [42], [43] in rendering realistic images. Implicit Neural Representations (INR) are a series of neural networks that optimize their parameters to fit one sample instead of the whole distribution. One principal advantage is that this technique has superb generalization ability on spatial or temporal dimensions. For example, Cervantes et al. [3] propose an implicit scheme, which simultaneously models action category and timestamp. Similar to the original NeRF, the timestamp is represented by sinusoidal values. After supervised training, the proposed method can generate a variable-length motion sequence for each action category.

This paper proposes a new motion generation pipeline based on the Denoising Diffusion Probabilistic Model (DDPM) [19]. One of the principal advantages of DDPM is that the formation of the original motion sequence can be retained. It means that we can easily apply more constraints during the denoising process. In the later sections, we will explore more potential of DDPM in different types of conditions. Besides, benefiting from this nature, DDPM can generate more diverse samples. Concurrent with this work, Tevet et al. [44], and Kim et al. [45] also attempted to exploit the potential of diffusion models on motion generation. Compared to these two works, we carefully design the architecture of MotionDiffuse to improve efficiency (with linear attention) and performance (with stylization block).

### B. Conditional Motion Generation

The increasing maturity of various generative models stimulates researchers’ enthusiasm to study conditional motion generation. For example, some works [1], [2], [3] aim at synthesizing motion sequences of several specific categories. Action2Motion [1] builds up a recurrent conditional VAE for motion generation. Given history memory, this model predicts the next pose under the constraints of the action category. ACTOR [2] also uses VAE for random sampling. Unlike Action2Motion, ACTOR embeds the whole motion sequence into the latent space. This design avoids the accumulative error in the recurrent scheme. Besides, ACTOR proposes a Transformer-based motion encoder and decoder architecture. This structure significantly outperforms recurrent methods. Cervantes et al. [3] attempt to model motion sequence with implicit functions, which can generate motion sequences with varied lengths.

Another significant conditional motion generation task is music to dance. This task requires that the generated motion has beat-wise connectivity, is a specific kind of dance, or can express similar content with the music. Many works attempt to embed the music feature and motion feature into a joint space [5], [6], [46], [47]. Unlike direct feature embedding, Bailando [8] proposes a two-stage dance generator. It first learns a quantized codebook of meaningful dance pieces and then attempts to generate the whole sequence with a series of elements from a codebook.

Similar to music-to-dance, text-driven motion generation can be regarded as learning a joint embedding of text feature space and motion feature space. There are two major differences. The first one is that language commands correlate more with the human body. Therefore, we expect to control each body part accurately. The second difference is that text-driven motion generation contains a vast range of motions. Some descriptions are direct commands to a specific body part, such as “touch head”. Some describes arbitrary concepts like “playing the violin”. Such huge complexity of motions brings many difficulties to the architecture design. Recently, many works have proposed text-driven motion generation pipelines. Most of them are deterministic generation [10], [11], [14], which means they can only generate a single result from the given text. TEMOS [12] introduces the VAE architecture into this task. It can generate different motion sequences given one text description. However, these methods attempt to acquire a joint embedding space of motion and natural language. This design significantly compresses the information from text. Therefore, these works can hardly generate correct motion sequences from a detailed description. Guo et al. [48] proposes an auto-regressive pipeline. It first encodes language descriptions into features and then auto-regressively generates motion frames conditioned on the text features. However, this method is hard to capture the global relation due to the auto-regressive scheme. Moreover, the generation quality is inferior. Instead, our proposed MotionDiffuse softly fuses text features into generation and can yield the whole sequence simultaneously. The experiment results prove the superiority of our design.

### C. Motion Datasets

Human motion modeling has been a long-standing problem in computer vision and computer graphics. With the advent of deep learning, data has become increasingly important for training neural networks that perceive, understand, and generate human motions.

A common form of datasets containing videos of human subjects are recorded with annotations such as 2D keypoints [49], [50], 3D keypoints [6], [51], [52], [53], [54] and statistical model parameters [55], [56], [57], [58], [59]. Action labels are also a popular attribute of datasets for human action understanding that contains human-centric actions [60], [61], [62], [63], [64],
interaction [66], [67], [68], fine-grained action understanding [63], [64], [65] and 3D data [69]. For the action-conditioned motion generation task, HumanAct12 [1], UESTC [70], and NTU RGB+D [69] are three commonly used benchmarks. However, the above-mentioned datasets do not provide paired sophisticated semantic labels to the motion sequences.

KIT [13] contains motion capture data annotated with detailed descriptions. Zhang et al. [71] recruit actors and actresses to record body movements expressing emotions. Recently, BABEL [72] and HumanML3D [48] re-annotates AMASS [73], a large scale motion capture dataset, with English language labels.

In this paper, we use the HumanML3D dataset and KIT dataset to evaluate the proposed methods for the text-driven motion generation task. HumanAct12 and UESTC are used to demonstrate the wide applicability of the proposed pipeline. Furthermore, we use the BABEL dataset for additional applications.

### III. Methodology

We present a diffusion model-based framework, Motion-Diffuse, for high-fidelity and controllable text-driven motion generation. We first give the problem definition, settings of the original text-driven motion generation in Section III-A. After that, we provide an overall illustration of the proposed Motion-Diffuse in Section III-B, followed by introducing the diffusion model in Section III-C and the transformer-based architecture in Section III-D. Finally, the inference strategy is illustrated for the fine-grained generation scenarios in Section III-E.

#### A. Preliminaries

The motion sequence is denoted as \( \Theta \in \mathbb{R}^{F \times D} \), where \( F \) is the number of frames and \( D \) the dimensionality of each pose state. The representation of each pose state is distinct in different datasets. It generally contains joint rotation, joint position, joint velocity, and foot contact conditions. Our proposed MotionDiffuse is robust to the various motion representations. Therefore, we do not specify the components in this section, and leave the details in Section IV.

For standard Text-driven Motion Generation, the training datasets consist of text-motion pairs. During inference, given a set of descriptions, we are requested to generate the corresponding motion sequence for each query. This task can also be regarded as a text-to-motion translation (T2M) task. We will use this abbreviation below.

An related task is Action-conditioned Motion Generation. Given a pre-defined action category set, models are supposed to fit the data distribution and synthesize motion sequences of each category. Annotated data in this task is a set of text-category pairs. In this paper, we replace the category ID by its semantic description so that we can use the same pipeline as in the T2M task.

#### B. Pipeline Overview

Following the literature on the diffusion model in the image synthesis field [19], we first build up a text-conditioned motion generation pipeline using a denoising diffusion probabilistic model (DDPM). This model is the basis of our proposed Motion-Diffuse. For the denoising process, we propose a Cross-Modality Linear Transformer to process input sequences conditioned on the given text prompts. Beyond the direct application of text-driven motion generation, we take one step further to explore methods that are conditioned on motion representation during denoising. Specifically, we experiment with two types of additional signals: part-aware text controlling and time-varied controlling. The former assigns different text conditions to different body parts so that we can accurately control each part of the body and generate more complicated motion sequences. The latter divides the whole sequence into several parts and assigns independent text conditions for each interval. Therefore, we can synthesize arbitrary-length motion sequences that incorporate several actions. These two kinds of conditions significantly expand the capability of MotionDiffuse. The overall pipeline is shown in Fig. 2. We introduce each part of this architecture in the following subsections.

#### C. Diffusion Model for Motion Generation

Generative Adversarial Networks (GANs) involve a discriminator to improve the generation quality in an adversarial manner. GANs are typically challenging to train [74], [75], especially for conditional motion generation tasks. Implicit Functions use Multi-Layer Perceptron (MLP) to fit motion sequences. This neat architecture is easily trained on a small number of data but tends to be less generalizable when it is subjected to complicated conditions. Auto-Encoder (AE) and Variational Auto-Encoder (VAE) are the most widely used approaches in text-driven motion generation [11], [12]. Previous works learn a joint embedding of motion sequences and languages that explicitly apply the text condition in the deterministic language-motion mapping. However, high-level text features typically contain insufficient fine-grained details to guide the generation of subtly different motion. Hence, directly linking text embedding to motion embedding results in the limited diversity of the generated motions.

To tackle the problem, we build our text-driven motion generation pipeline based on diffusion models. Diffusion Models [15], [16], [17], [19] are a new class of generative models. A probabilistic model is learned to gradually denoise a Gaussian noise to generate a target output, such as a 2D image or 3D point cloud. Formally, diffusion models are formulated as

\[
p_0(x_0) := \int p_0(x_{0:T}) \, d\mathbf{x}_{1:T},
\]

where \( x_0 \sim q(x_0) \) is the real data, and \( x_1, \ldots, x_T \) are the latent data. They generally have a diffusion process and a reverse process. To approximate posterior \( q(x_{1:T}|x_0) \), the diffusion process follows a Markov chain to gradually add Gaussian noise to the data until its distribution is close to the latent distribution \( \mathcal{N}(0, I) \), according to variance schedules given by \( \beta_t \):

\[
q(x_{1:T}|x_0) := \prod_{t=1}^{T} q(x_t|x_{t-1}),
\]

\[
q(x_t|x_{t-1}) := \mathcal{N} \left( x_t; \sqrt{1 - \beta_t}x_{t-1}, \beta_tI \right). \tag{1}
\]

The reverse process \( p_0(x_{0:T}) \) is also a Markov chain that predicts and eliminates the noise with learned Gaussian transitions.
starting at \( p(x_T) = N(x_T; 0, I) \):

\[
p_\theta(x_{0:T}) := p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t),
\]

\[
p_\theta(x_{t-1}|x_t) := N(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)). \tag{2}
\]

To accomplish the reverse process of the diffusion model, we need to construct and optimize a neural network. During training, first we uniformly sample steps \( t \) for each ground truth motion \( x_0 \) and then generate a sample from \( q(x_t|x_0) \). Instead of repeatedly adding noises on \( x_0 \), [19] formulate the diffusion process as

\[
q(x_t|x_0) = \sqrt{\alpha_t} x_0 + \epsilon \sqrt{1-\alpha_t}, \epsilon \sim N(0, I), \tag{3}
\]

where \( \alpha_t = 1 - \beta_t, \beta_t = \prod_{\tau=1}^{t} \beta_\tau \). Hence, we can simply sample a noise \( \epsilon \) and then directly generate \( x_t \) by this formulation. The dimensionality of \( \epsilon \) is \( \mathbb{R}^{F \times D} \), which is the same as that of the input motion sequence \( x_0 \). Instead of predicting \( x_{t-1} \), here we follow GLIDE [17] and predict the noise term \( \epsilon \). Specifically, we construct a network to fit \( \epsilon_\theta(x_t, t, \text{text}) \). We optimize the model parameters to decrease a mean squared error as

\[
\mathcal{L} = E_{t \in [1,T], x_0 \sim q(x_0), \epsilon \sim N(0, I)} \mathbb{E} \left[ \| \epsilon - \epsilon_\theta(x_t, t, \text{text}) \| \right]. \tag{4}
\]

This is the only loss we used in model training. To generate samples from the given text description, we denoise the sequence from \( p(x_T) = N(x_T; 0, I) \). Equation (2) shows that we need to estimate \( \mu_\theta(x_t, t) \) and \( \Sigma_\theta(x_t, t) \). To simplify the problem, we follow the literature [15] to set \( \Sigma_\theta(x_t, t) \) as the constant value \( \beta_t \), which is the same one that we used in \( \epsilon\text{-}

Equation (1). \( \mu_\theta(x_t, t, \text{text}) \) is then estimated as below [15]:

\[
\mu_\theta(x_t, t, \text{text}) = \frac{1}{\sqrt{\alpha_t}} \left( x_t - \frac{1 - \alpha_t}{\sqrt{1 - \beta_t}} \epsilon_\theta(x_t, t, \text{text}) \right). \tag{5}
\]

Hence, for each step, the motion sequence is initially denoised to \( x_0 \) and subsequently denoised to \( x_{t-1} \). By iteratively following this process, we ultimately obtain a clean motion sequence that is conditioned on the provided text.

**D. Cross-Modality Linear Transformer**

In Section III-C, we illustrate diffusion models as motion generators and a neural network \( \epsilon_\theta(x_t, t, \text{text}) \) is used for denoising steps. In this section, we will introduce the design of \( \epsilon_\theta(x_t, t, \text{text}) \) in our proposed MotionDiffuse.

Previous works [15], [16], [17], [19] mainly utilize UNet-like structure as the denoising model. However, the target motion sequences are variable-length in the motion generation task, making convolution-based networks unsuitable. Therefore, we propose a Cross-Modality Linear Transformer, as shown in Fig. 2. Similar to the machine translation task, our proposed model includes a text encoder and a motion decoder. To meet the requirement of the diffusion models, we further customize each layer of the motion decoder.

**Text Encoder:** Here we utilize a network architecture similar to that of a Transformer Encoder [31] to extract text features. Specifically, the input data first passes through an embedding layer to get the embedding feature from raw text and then is further processed by a series of transformer blocks. Each block contains two components: a multi-head attention module (MHA) and a feed-forward network (FFN). Suppose the input feature is...
Architecture

ψ

by

·

from other blocks, the Stylization block will process (Y by positional embedding) to the generation process. This block is applied after each \[17\] = \[e\] of our Stylization Block.

\[O_\psi, \quad \psi = \text{softmax}(\text{calculation by the text feature). Other formulations are the}\]

\[W = \text{softmax}(8)\]

Y

K

K

d
layers between them.

\[\in \text{softmax}\]

Y

W

V

R

W

∈

is the output of MHA modules,

\[\in \otimes \in \in \epsilon\]

is the number of the elements in the sequences, \(k\) is the number of heads of self-attention.

Another advantage of efficient attention for the diffusion model is that feature map \(F_g\) explicitly aggregates global information while classical self-attention focus more on pair-wise relation. Global information gives more clues about the semantic meaning of the motion sequence than pair-wise one. The experiment results also prove this conclusion.

Linear Cross-Attention: Cross-attention replaces \(X\) in \(K\) and \(V\) calculation by the text feature. Other formulations are the same as Linear Self-attention. Text features are injected into motion sequences in this process to generate motion conditioned on the given text.

Stylization Block: In each denoising step, the output is conditioned on the given text and the timestamp \(t\). Linear Cross-attention fuses the text features into motion sequences. We utilize another Stylization Block component to bring timestamp \(t\) to the generation process. This block is applied after each Linear Self-attention block, Linear Cross-attention block, and FFN block.

The structure of our Stylization Block is shown in Fig. 4. Similar to GLIDE [17], we first get a text embedding \(e_p\) by a linear transformation on the text features and a timestamp embedding \(e_t\) by positional embedding [31]. These two terms are summed together into one single vector \(e\). Given the original output \(Y\) from other blocks, the Stylization block will process the feature as:

\[B = \psi_b(\phi(e)), \quad W = \psi_w(\phi(e)), \quad Y' = Y \cdot W + B,\]

where (\cdot) denotes Hadamard product, \(Y'\) is the output of stylization blocks. \(\psi_b, \psi_w, \phi\) denote three different linear projections. In classical transformers, the output from each block is added to the original input as a residual connection, as shown in Fig. 2. In MotionDiffuse, these outputs pass through stylization blocks and are added to the information. This modification enables our proposed method to know the timestamp \(t\).
E. Fine-Grained Controlling

To enrich the capability of MotionDiffuse, we explore the properties of both the motion representation and the denoising process of DDP. Unlike VAE, the generated motion sequence is in its explicit form instead of being compressed in the latent space. This characteristic of DDP-based motion generation allows more operations to be applied to this motion sequence to increase the manipulability, without additional training strategy or data formation. Here we show two examples: Body Part-independent Controlling and Time-varied Controlling.

Body Part-Independent Controlling: Due to the lack of diversity in text descriptions, we cannot achieve accurate motion control for each body part from text descriptions only. For example, the prompt ‘a person is jumping and raising arms’ is highly challenging to the model because the expected motion sequence is significantly far from the training distribution. Even if we manually split the original description into two independent ones: ‘a person is jumping’ for lower limbs, and ‘a person is raising arms’ for upper limbs, it is still difficult for the model to generate correct motions. An intuitive solution for this situation is to separately generate two motion sequences and combine the upper-limb motion of the first sequence and the lower-limb motion of the second sequence. This simple solution mitigates the problem to some extent. However, it ignores the correlation between these two parts. Specifically for ‘jumping and raising arms’, the arms should reach their highest point during the upward phase of the person’s jump. If they extend to the highest point during the descending phase of the jump, it may appear less natural. To better solve this problem, we propose a body part-independent controlling scheme.

Recall that, during the denoising process, our diffusion model predicts the noise term \( \epsilon_i(x_{ij}, t, \text{text}) \in \mathbb{R}^{F \times D} \), where \( F \) represents the number of frames, \( D \) denotes the dimension of each pose state, which includes translation and rotations of body joints. This noise term determines the denoising direction of the whole body.

Inspired by the application of the latent code interpolation, here we propose ‘noise interpolation’ to separately control different parts of the human body. Suppose we have \( n \) text descriptions \( \{\text{text}_i\} \) for different body parts \( \{s_i\} \). We want to calculate the noise term \( \epsilon = \{\epsilon_{ij}^{\text{joint}}\}, i \in [1, m] \), where \( \epsilon_{ij}^{\text{joint}} \) represents the noise term for the \( i \)-th body part, \( m \) denotes the number of partition. We first estimate the noise \( \epsilon_{ij} = \epsilon_i(x_{ij}, t, \text{text}_i), \epsilon^{\text{part}}_{ij} \in \mathbb{R}^{F \times D} \). An intuitive method for combining these terms is \( \epsilon^{\text{part}} = \sum_{i=1}^{m} \epsilon_{ij}^{\text{part}} \cdot M_i \), where \( M_i \in \{0, 1\}^D \) is a binary vector to show which body part should we focus. \((\cdot)\) denotes the Hadamard product, and here we ignore the broadcast in computation for simplicity. Although this method succeeds to some extent, the direct ignoring of some parts in \( \epsilon^{\text{part}} \) will increase the combination difficulty and lead to low-quality generation results. Therefore, we add a correction item for smoothing interpolation:

\[
\epsilon^{\text{part}} = \sum_{i=1}^{m} \epsilon_{ij}^{\text{part}} \cdot M_i + \lambda_1 \cdot \nabla \left( \sum_{1 \leq i, j \leq m} \| \epsilon_{ij}^{\text{part}} - \epsilon_j^{\text{part}} \| \right), \quad (10)
\]

where \( \nabla \) denotes gradient calculation, \( \lambda_1 \) is a hyper-parameter to balance these two items. Because different parts of the predictions are likely to have a significant overlap, for instance, when generating an action where the upper body is punching and the lower body is kicking, the predictions for the former include estimates for the lower body, and vice versa for the latter. In this context, the expectation for the correlation item is that results from different predictions in the same region should be as similar as possible. For example, in the previous example, we introduce this item to make the estimates for the upper and lower body as similar as possible from two prompts, resulting in smoother combinations. As L1 and L2 loss would largely serve the same purpose with proper lambda values, the choice of loss function was not the key consideration of our method.

Time-Varied Controlling: Long-term motion generation plays a vital role in real-world applications. Previous works mainly focus on motion generation with a single type of motion. Autoregressive methods [40], [48] have solved this problem with satisfactory performance. However, none of them are capable of synthesizing different actions in a continuous manner. Benefiting from the nature of DDP, here we propose another sampling method to meet this requirement.

Recall that we are given an array \( \{\text{text}_{ij}, [l_{ij}, r_{ij}]\}, i \in [1, m] \), where \( m \) is the number of intervals. Similar to the method we proposed in the previous paragraph, we first estimate the noise term \( \epsilon_{ij}^{\text{time}} \) for \( i \)-th interval independently. Suppose the overall length of the target sequence is \( F' \). By padding zeros, we extend each noise term into the same dimension \( F' \times D \). Then we interpolate these noises with a correcting term:

\[
\epsilon_{ij}^{\text{time}} = \sum_{i=1}^{m} \epsilon_{ij}^{\text{time}} + \lambda_2 \cdot \nabla \left( \sum_{1 \leq i, j \leq m} \| \epsilon_{ij}^{\text{time}} - \epsilon_j^{\text{time}} \| \right), \quad (11)
\]

where \( \epsilon_{ij}^{\text{time}} \) is the padded term from \( \epsilon_{ij}^{\text{time}} \), \( \lambda_2 \) is a hyper-parameter.

IV. EXPERIMENTS

We evaluate MotionDiffuse with three categories of experiments: text-driven motion generation (Section IV-A), action-conditioned motion generation (Section IV-B), and motion manipulation (Section IV-C). In all the evaluated benchmarks, MotionDiffuse could significantly outperform previous SoTA methods.

A. Text-Driven Motion Generation

Datasets: KIT Motion Language datset [13] provides 3911 motion sequences and 6353 sequence-level natural language descriptions. HumanML3D [48] re-annotates the AMASS dataset [73] and the HumanAct12 dataset [1]. It provides 44970 annotations on 14616 motion sequences. KIT and HumanML3D are two important benchmarks for text-driven motion generation tasks. Following Guo et al. [48], we utilize the pretrained text-motion contrastive model.

Evaluation Metrics: We evaluate all methods with five different metrics. 1) Frechet Inception Distance (FID). Features are extracted from both the generated results and ground truth motion sequences by the pretrained motion encoder. FID is
We compare our proposed MotionDiffuse with five baseline models: Language2Pose [10], Text2Gesture [80], MoCoGAN [81], Dance2Music [46], and Guo et al. [48]. All baselines’ performances are quoted from Guo et al. [48]. Tables I and II show the quantitative comparison on the HumanML3D dataset and the KIT-ML dataset. We also train TEMOS [12] on both datasets from scratch based on the official implementation, with the same motion representation as our method for a fair comparison. Our proposed MotionDiffuse outperforms all existing works with a remarkable margin in aspects of precision, FID, MultiDodal Distance, and Diversity. The precision of MotionDiffuse is even close to that of real motions, which suggests that our generated motion sequences are satisfyingly high-fidelity and realistic.

Guo et al. [48] states that the results on the MultiModality metric should be larger whenever possible. However, the literature in action-conditioned motion generation task [1], [2], [3] argue that this metric should be close to the real motion. In the T2M task, it is difficult to calculate this metric of real motions. Therefore, we only report these results without comparison.

**Qualitative Results:** To further understand the function of CLIP initialization, efficient attention and stylization block, we report ablation results in Table III. The models without pretrained CLIP suffer from severe performance drops, which indicates the necessity of a pretrained language model for the T2M task. As for efficient attention, it is significantly beneficial when we use CLIP simultaneously. However, this module also limits the model’s performance when without CLIP. A possible explanation for this phenomenon is that the global relation in efficient attention is misleading when the semantic information from given text is insufficient. The stylization blocks are crucial for achieving better performance due to their effective incorporation of diffusion time steps. Additionally, we devised a simpler baseline that employs eight transformer encoder layers to process text features from scratch and eight transformer decoder layers to translate from language sequences into motion sequences. The quantitative results indicate that such a straightforward architecture is not suitable for complex text-driven motion generation tasks. We also provide information on the average inference time, and from this column, it is evident that the efficient attention mechanism significantly enhances efficiency.
TABLE II
QUANTITATIVE RESULTS ON THE KIT-ML TEST SET

| Methods | Top 1 | Top 2 | Top 3 | FID | MultiModal Dist | Diversity | MultiModality |
|---------|------|------|------|-----|-----------------|-----------|---------------|
| Real motions | 0.42±0.046 | 0.649±0.006 | 0.759±0.006 | 0.031±0.004 | 2.788±0.012 | 11.08±0.067 | - |
| Language2Pose | 0.221±0.005 | 0.373±0.004 | 0.483±0.005 | 6.546±0.072 | 5.147±0.060 | 9.073±0.100 | - |
| Text2Gesture | 0.156±0.004 | 0.255±0.004 | 0.338±0.005 | 12.12±183 | 6.964±0.029 | 9.334±0.079 | - |
| MoCoGAN | 0.022±0.002 | 0.042±0.003 | 0.063±0.003 | 82.69±242 | 10.47±0.012 | 3.091±0.043 | 0.250±0.009 |
| Dance2Music | 0.031±0.002 | 0.058±0.002 | 0.086±0.003 | 115.4±260 | 10.40±0.016 | 0.241±0.004 | 0.062±0.002 |
| Guo et al. | 0.370±0.005 | 0.569±0.007 | 0.693±0.007 | 2.776±0.109 | 3.401±0.008 | 10.91±0.119 | 1.482±0.065 |
| TEMOS | 0.367±0.005 | 0.569±0.004 | 0.694±0.008 | 2.655±0.052 | 3.553±0.016 | 9.83±0.081 | 0.328±0.015 |
| Ours | 0.417±0.004 | 0.621±0.004 | 0.739±0.004 | 1.954±0.062 | 2.958±0.005 | 11.10±0.148 | 0.730±0.014 |

All methods use the real motion length from the ground truth.

TABLE III
ABLATION OF THE PRETRAINED CLIP, THE EFFICIENT ATTENTION TECHNIQUE (EFF) AND THE STYLIZATION BLOCK (STY)

| CLIP | EFF | STY | Top 1 | Top 2 | Top 3 | Time(s) |
|------|-----|-----|------|------|------|---------|
| -    | -   | -   | 0.109±0.003 | 0.198±0.003 | 0.254±0.003 | 0.002 |
| N N Y | Y Y Y | Y Y Y | 0.283±0.004 | 0.440±0.004 | 0.539±0.004 | 0.317 |
| N Y Y | Y Y Y | Y Y Y | 0.136±0.003 | 0.233±0.003 | 0.309±0.003 | 0.129 |
| Y N Y | Y Y Y | Y Y Y | 0.357±0.004 | 0.555±0.004 | 0.679±0.005 | 0.317 |
| Y Y N | Y Y Y | Y Y Y | 0.152±0.003 | 0.247±0.003 | 0.329±0.003 | 0.117 |
| Y Y Y | Y Y Y | Y Y Y | 0.417±0.004 | 0.621±0.004 | 0.739±0.004 | 0.129 |

All results are reported on the KIT-ML test set. The absence of “EFF” signifies the use of traditional self-attention and cross-attention mechanisms. In the first row, a simple baseline is presented, utilizing a Transformer to directly convert token sequences into motion sequences. Furthermore, we provide information on the average processing time for each generation query.

TABLE IV
ABLATION OF THE LATENT DIMENSION AND THE NUMBER OF TRANSFORMER LAYERS

| #layers | Dim | Top 1 | Top 2 | Top 3 | R Precision† |
|---------|-----|------|------|------|-------------|
| 4      | 128  | 0.034±0.002 | 0.066±0.005 | 0.097±0.003 | 0.000±0.000 |
| 4      | 256  | 0.095±0.002 | 0.166±0.003 | 0.227±0.003 | 0.000±0.000 |
| 4      | 512  | 0.405±0.005 | 0.620±0.005 | 0.743±0.004 | 0.000±0.000 |
| 8      | 128  | 0.025±0.002 | 0.053±0.002 | 0.086±0.002 | 0.000±0.000 |
| 8      | 256  | 0.198±0.003 | 0.334±0.004 | 0.441±0.004 | 0.000±0.000 |
| 8      | 512  | 0.417±0.004 | 0.621±0.004 | 0.739±0.004 | 0.000±0.000 |
| 12     | 128  | 0.031±0.002 | 0.063±0.003 | 0.091±0.002 | 0.000±0.000 |
| 12     | 256  | 0.209±0.003 | 0.348±0.004 | 0.452±0.004 | 0.000±0.000 |
| 12     | 512  | 0.412±0.006 | 0.616±0.004 | 0.741±0.004 | 0.000±0.000 |

All results are reported on the KIT-ML test set.

We explore how the size of architecture influences the performance. Table IV suggests that the latent dimension plays a more important role. The models with 512 latent dimension significantly outperform the models with 256 latent dimension. On the contrary, the increase of the number of layers improves the performance when the latent dimension is either 128 or 256, but has little effect when the dimension is 512.

Fig. 5 shows a comparison between our method and Guo et al. [48] as a baseline. We highlight that MotionDiffuse achieves a balance between diversity and realness. For example, for prompt ‘A person swings a golf club and hits the ball’, our generated motions portrait the described motion more faithfully. In contrast, the baseline method has high multi-modality at the expenses of accuracy. In addition, given a complicated prompt such as “A person is pushed hard to the left and then recovers into a standing position”, MotionDiffuse is able to generate high-quality motions that reflects the detailed description whereas the baseline method fails to produce any meaningful movement.

We also qualitatively compare our method with TEMOS on the KIT-ML dataset as shown in Fig. 6. The KIT-ML dataset is smaller than HumanML3D, thus leading to lower motion diversity. However, MotionDiffuse can still generate diverse motion sequences consistent with the given prompts. As for “A human performs punches and kicks”, TEMOS generates two similar motion sequences, which only contain “punches”. It also only generates arm transitions for the prompt “dancing freely”. These examples show that TEMOS has difficulties generating complicated motion sequences, which contain multiple actions or are rare in the training set.

User Study: We randomly selected 25 samples from the HumanML3D test set to conduct a comparative evaluation involving Guo et al. TEMOS, and MotionDiffuse. A total of 42 responses were collected, and the outcomes are presented in Fig. 8. It is evident that in the majority of cases, evaluators found the motions generated by our method to exhibit greater consistency with the provided text descriptions, resulting in more natural animations. This validates the superior quality of our proposed motion generation framework.

B. Action-Conditioned Motion Generation

Datasets: HumanAct12 dataset [1] provides 12 kinds of motion sequences. This dataset is adapted from PHSPD dataset [82], which contains 1191 videos. HumanAct12 further arranges these videos into trimmed motion clips. UESTC dataset [70] is also a significant benchmark for action-conditioned motion generation tasks, which includes 25 K motion sequences across 40 different action categories. ACTOR [2] further uses pre-trained VIBE [83] to extract SMPL [84] sequences from the UESTC dataset and provides pretrained action recognition model for evaluation.

Evaluation Metrics: Four evaluation metrics are applied for this task: FID, Accuracy, Diversity, and Multimodality. The pretrained action recognition module can directly calculate the average accuracy for all action categories without arranging mini-batches. This metric has a similar function to R Precision. The other three metrics have been introduced in Section IV-A. HumanAct12 has no official split, and we report the FID on the whole dataset. UESTC has a test split, so we report the FID on it.
which is more representative than the train split. In this section, FID and Accuracy are two principal metrics. Our conclusion are mainly based on them.

Implementation Details: All the setting are the same to those for text-driven motion generation tasks except for the learning rate, the number of iterations and the motion representation. In this series of experiments, we train 100 K iterations for the HumanAct12 dataset and 500 K for the UESTC dataset, both with a 0.0001 learning rate.

Motion representation in this task is slightly different from the T2M task. As for the HumanAct12 dataset, each pose state can be represented as $(j^x, j^y, j^z)$, where $j^x, j^y, j^z \in \mathbb{R}^{24 \times 3}$ are the coordinates of 24 joints. We use $(r^x, r^y, r^z, j')$ as the pose representation for the UESTC dataset, where $r^x, r^y, r^z \in \mathbb{R}$ are
Figure 7. Qualitative results on the BABEL dataset. MotionDiffuse is able to generate dynamic sequences according to fine-grained prompt that involves multiple body parts or actions.

Table V

| Methods  | HumanAct12 | UESTC |
|----------|-------------|-------|
|          | FID↓ | Accuracy↑ | Diversity→ | MM→ | FID↓ | Accuracy↑ | Diversity→ | MM→ |
| Real motions | 0.020 ± 0.001 | 0.997 ± 0.001 | 6.850 ± 0.001 | 2.450 ± 0.001 | 2.79 ± 0.29 | 0.988 ± 0.001 | 33.34 ± 3.20 | 14.16 ± 8.06 |
| Action2Motion | 0.338 ± 0.019 | 0.917 ± 0.003 | 6.879 ± 0.066 | 2.511 ± 0.023 | - | - | - | - |
| ACTOR | 0.12 ± 0.00 | 0.955 ± 0.008 | 6.84 ± 0.03 | 2.53 ± 0.02 | 23.43 ± 2.20 | 0.911 ± 0.003 | 31.96 ± 3.33 | 14.52 ± 0.09 |
| INR | 0.088 ± 0.004 | 0.973 ± 0.001 | 6.881 ± 0.04 | 2.569 ± 0.040 | 15.00 ± 0.09 | 0.941 ± 0.001 | 31.59 ± 1.90 | 14.66 ± 0.07 |
| Ours | 0.07 ± 0.009 | 0.992 ± 0.13 | 6.85 ± 0.02 | 2.46 ± 0.04 | 9.10 ± 2.35 | 0.950 ± 0.000 | 32.42 ± 2.14 | 14.74 ± 0.07 |

As for UESTC dataset, we report FID on the test split. MM: MultiModality.

Figure 8. Result of our user study. The results from our user study demonstrate a significant superiority of our approach in terms of both text-motion consistency and motion quality when compared to other methods.

Table VI

| #layers | Dm | FID↓ | Accuracy↑ |
|---------|----|------|------------|
| 4       | 128 | 0.29 ± 0.00 | 0.892 ± 1.37 |
| 4       | 256 | 0.14 ± 0.00 | 0.958 ± 0.51 |
| 4       | 512 | 0.09 ± 0.00 | 0.984 ± 0.21 |
| 8       | 128 | 0.22 ± 0.00 | 0.929 ± 1.04 |
| 8       | 256 | 0.09 ± 0.00 | 0.983 ± 0.23 |
| 8       | 512 | 0.07 ± 0.00 | 0.992 ± 0.13 |
| 12      | 128 | 0.11 ± 0.00 | 0.954 ± 0.67 |
| 12      | 256 | 0.10 ± 0.00 | 0.988 ± 0.21 |
| 12      | 512 | 0.08 ± 0.00 | 0.996 ± 0.08 |

All results are reported on the HumanAct12 dataset.

Quantitative Results: Following Cervantes et al. [3], three baseline models are selected: Action2Motion [1], ACTOR [2], INR [3]. Table V shows the quantitative results on the HumanAct12 dataset and the UESTC datasets. Our proposed MotionDiffuse achieves the best performance in aspects of FID and Accuracy when compared to other existing works. We want to highlight that our results of the HumanAct12 dataset are notably close to real motions on all four metrics.

Table VI

Ablation of the Latent Dimension and the Number of Transformer Layers

C. Motion Manipulation

To better evaluate the capability of text-driven motion generation models, we design two task variants. First, Spatially-diverse
T2M task (T2M-S). T2M-S requires the generated motion sequence to contain multiple actions on different body parts (e.g., ‘a person is running and drinking water simultaneously’). Specifically, i-th test sample in T2M-S task can be represented by a set of text-mask pairs \( \{ (\text{text}_{i,j}, \text{M}_{i,j}) \} \), where \( \text{M}_{i,j} \in \{0, 1\}^D \) is a D-dimension binary vector. It indicates which body part we should focus on when given the text description \( \text{text}_{i,j} \). Second, Temporally-diverse T2M task (T2M-T). T2M-T expects models to generate a long motion sequence, which includes multiple actions in a specific order spanning over different time intervals (e.g., ‘a person is walking and then running’). The i-th test sample is an array of text-duration pairs \( \{ (\text{text}_{i,j}, (l_{i,j}, r_{i,j})) \} \), where \( l_{i,j} < r_{i,j} \). It means that the motion clip from \( l_{i,j} \)th frame to \( r_{i,j} \)th frame is supposed to contain the action \( \text{text}_{i,j} \).

Implementation Details: We train our proposed MotionDiffuse on the BABEL dataset [72] with 50 K iterations. Each pose state is represented by \( r^x, r^y, r^z, J \), which is same to the setting for the UESTC dataset. Other settings remain unchanged. \( \lambda_1 = \lambda_2 = 0.01 \) are used for the visualization.

Quantitative Results: To better evaluate our proposed method, we use the similar metrics to compare our method and TEACH [85] on the BABEL test set, as shown in Table VII. It should be mentioned that our method is only trained on single actions and generate long sequences in a zero-shot manner. The results show that our proposed method outperforms TEACH with a remarkable margin. Additionally, the correction item we proposed also contributes significantly to the final performance.

Qualitative Results: As shown in Fig. 7, MotionDiffuse has the capability to handle highly comprehensive prompts that assign motions to multiple body parts (such as ‘Kicking and punching’ and ‘Jumping and raising arms’ that require coordination of the upper and lower body). Moreover, MotionDiffuse is able to generate long sequences according to a complex instruction that includes multiple actions (such as ‘Tying the shoe, standing up and then walking forward’ that includes a series of vastly different motions).

V. CONCLUSION, LIMITATIONS AND FUTURE WORK

We propose MotionDiffuse, one of the first diffusion model-based methods for text-driven motion generation. MotionDiffuse demonstrates three major strengths: Probabilistic Mapping that enhances diversity, Realistic Synthesis that ensures plausibility of motion sequences, and Multi-Level Manipulation that allows for per-part manipulation and long sequence generation. Both quantitative and qualitative evaluations show that MotionDiffuse outperforms existing arts on various tasks such as text-driven motion generation and action-conditioned motion generation, and demonstrates remarkable motion manipulation capabilities.

Although MotionDiffuse has pushed forward the performance boundary of motion generation tasks, there still exist some problems. First, diffusion models require a large amount of diffusion steps during inference and it is challenging to generate motion sequences in real-time. Second, current pipeline only accepts a single form of motion representation. A more generalized pipeline that adapts concurrently to all datasets would be more versatile for various scenarios.

**TABLE VII**

| Methods | R Precision | FD2L | Diversity | Multimodality |
|---------|-------------|------|-----------|---------------|
| Real motions | 0.62 | 0.004 | 8.51 | 3.37 |
| TEACH | 0.46 | 1.12 | 8.28 | 7.14 |
| Ours (w/o correction) | 0.42 | 1.58 | 8.21 | 7.62 |
| Ours (w/ correction) | 0.50 | 0.97 | 8.59 | 8.03 |

The R Precision shown in this table is under the Top 3 configuration.

**REFERENCES**

[1] C. Guo et al., “Action2motion: Conditioned generation of 3D human motions,” in Proc. 28th ACM Int. Conf. Multimedia, 2020, pp. 2021–2029.

[2] M. Petrovich, M. J. Black, and G. Varol, “Action-conditioned 3D human motion synthesis with transformer VAE,” in Proc. IEEE/ICCV Int. Conf. Comput. Vis., 2021, pp. 10 985–10 995.

[3] P. Cervantes, Y. Sekikawa, I. Sato, and K. Shinoda, “Implicit neural representations for variable length human motion generation,” in Proc. Eur. Conf. Comput. Vis., 2022, pp. 356–372.

[4] R. Huang, H. Hu, W. Wu, K. Sawada, M. Zhang, and D. Jiang, “Dance revolution: Long-term dance generation with music via curriculum learning,” 2020, arXiv:2006.06119.

[5] J. Li et al., “Learning to generate diverse dance motions with transformer,” 2020, arXiv:2008.06171.

[6] R. Li, S. Yang, D. A. Ross, and A. Kanazawa, “AI choreographer: Music conditioned 3D dance generation with AIST,” in Proc. IEEE/ICCV Int. Conf. Comput. Vis., 2021, pp. 13 401–13 412.

[7] W. Zhuang, C. Wang, J. Chai, Y. Wang, M. Shao, and S. Xia, “Music2dance: DanceNet for music-driven dance generation,” ACM Trans. Multimedia Comput. Commun. Appl., vol. 18, no. 2, pp. 1–21, 2022.

[8] L. Siyao et al., “Bailando: 3D dance generation by actor-critic GPT with choreographic memory,” in Proc. IEEE/ICCV Conf. Comput. Vis. Pattern Recognit., 2022, pp. 11050–11059.

[9] A. S. Lin, L. Wu, R. Corona, K. Tai, Q. Huang, and R. J. Mooney, “Generating animated videos of human activities from natural language descriptions,” in Proc. Visually Grounded Interaction Lang. Workshop Neural Inf. Process. Syst. Workshop, 2018. [Online]. Available: https://nips2018vigil.github.io/

[10] C. Abuja and L.-P. Morency, “Language2Pose: Natural language grounded pose forecasting,” in Proc. Int. Conf. 3D Vis., 2019, pp. 719–728.

[11] A. Ghosh, N. Cheema, C. Oguz, C. Theobalt, and P. Slusallek, “Synthesis of compositional animations from textual descriptions,” in Proc. IEEE/ICCV Int. Conf. Comput. Vis., 2021, pp. 1396–1406.

[12] M. Petrovich, M. J. Black, and G. Varol, “TEMOS: Generating diverse human motions from textual descriptions,” in Proc. Eur. Conf. Comput. Vis., 2022, pp. 480–497.

[13] M. Plappert, C. Mandery, and T. Asfour, “The kit motion-language interface,” Big data, vol. 4, no. 4, pp. 236–252, 2016.

[14] G. Tevet, B. Gordon, A. Hertz, A. H. Bermano, and D. Cohen-Or, “Motion2clip: Exposing human motion generation to clip space,” in Proc. Eur. Conf. Comput. Vis., 2022, pp. 358–374.

[15] P. Dharwal and A. Nichol, “Diffusion models beat GANs on image synthesis,” in Proc. Adv. Neural Inf. Process. Syst., 2021.

[16] A. Q. Nichol and P. Dharwal, “Improved denoising diffusion probabilistic models,” in Proc. Int. Conf. Mach. Learn., 2021, pp. 8162–8171.

[17] A. Q. Nichol et al., “GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models,” in Proc. Int. Conf. Mach. Learn., 2022, pp. 16 784–16 804.

[18] A. Ramesh, P. Dharwal, A. Nichol, C. Chu, and M. Chen, “Hierarchical text-conditional image generation with clip latents,” 2022, arXiv:2204.06125.

[19] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” in Proc. Adv. Neural Inf. Process. Syst., 2020, pp. 6840–6851.

[20] C. Rose, M. F. Cohen, and B. Bodenheimer, “Verbs and adverbs: Multi-dimensional motion interpolation,” IEEE Comput. Graph. Appl., vol. 18, no. 5, pp. 32–40, Sep./Oct. 1998.
T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, “Training generative adversarial networks with limited data,” in Procs. Adv. Neural Inf. Process. Syst., 2020, pp. 12 104–12 114.

D. Saxena and J. Cao, “Generative adversarial networks (GANs) challenges, solutions, and future directions,” ACM Comput. Surv., vol. 54, no. 3, pp. 1–42, 2021.

D. Hendrycks and K. Gimpel, “Gaussian error linear units (GELUs),” 2016, arXiv:1606.08415.

A. Radford et al., “Learning transferable visual models from natural language supervision,” in Procs. Int. Conf. Mach. Learn., 2021, pp. 8748–8763.

Z. Shen, M. Zhang, H. Zhao, S. Yi, and H. Li, “Efficient attention: Attention with linear complexities,” in Procs. IEEE/CVF winter Conf. Appl. Comput. Vis., 2021, pp. 3531–3539.

Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, “On the continuity of rotation representations in neural networks,” in Procs. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 5745–5753.

U. Bhattacharya, N. Rewkowski, A. Banerjee, P. Guhan, A. Bera, and D. Manocha, “Text2gestures: A Transformer-Based Network for Generating Emotive Body Gestures for Virtual Agents,” in Procs. IEEE Virtual Reality 3D User Interfaces, 2021, pp. 1–10.

S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz, “MoCoGAN: Decomposing motion and content for video generation,” in Procs. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 1526–1535.

S. Zou et al., “3D human shape reconstruction from a polarization image,” in Procs. Eur. Conf. Comput. Vis., 2020, pp. 351–368.

M. Kocabas, N. Athanasiou, and M. J. Black, “Vibe: Video inference for human body pose and shape estimation,” in Procs. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 5253–5263.

M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, “Smpl: A skinned multi-person linear model,” ACM Trans. Graph., vol. 34, no. 6, pp. 1–16, 2015.

N. Athanasiou, M. Petrovich, M. J. Black, and G. Varol, “Teach: Temporal action composition for 3D humans,” in Procs. Int. Conf. 3D Vis., 2022, pp. 414–425.

Mingyuan Zhang received the BS degree in computer science and engineering from Beihang University, China. He is currently working toward the PhD degree with MMLab@NTU, advised by Prof. Ziwei Liu. His research interests include computer vision including motion vision, 3D pose estimation, and scene understanding. He has published papers on top-tier conferences, including CVPR, ECCV, ICCV, ICLR, and AAAI.

Zhongang Cai received the bachelor’s degree from Nanyang Technological University and was awarded the Lee Kuan Yew Gold Medal as a top student. He is currently working toward the PhD degree with Nanyang Technological University (NTU) supervised by Prof. Liu Ziwei and Prof Chen Change Loy, and a senior algorithm researcher with SenseTime International Pte. Ltd. His research interest includes point clouds and virtual humans. To date, he has published more than 10 papers on top venues, such as ICLR, CVPR, ICCV, and ECCV. He also serves as a reviewer for top machine learning and computer vision conferences, such as NeurIPS, ICLR, and ICCV.

Liang Pan received the PhD degree in mechanical engineering from the National University of Singapore, in 2019. He is currently a research fellow with S-Lab, Nanyang Technological University, Singapore. Previously, he is a research fellow with the Advanced Robotics Centre, National University of Singapore. His research interests include computer vision and 3D point cloud, with focuses on shape analysis, deep learning, and 3D human. He also serves as a reviewer for top computer vision and robotics conferences, such as CVPR, ICCV, ECCV, ICLR, NeurIPS, IROS, and ICRA.

Fangzhou Hong received the BEng degree in software engineering from Tsinghua University, China, in 2020. He is currently working toward the PhD degree with the School of Computer Science and Engineering, Nanyang Technological University. His research interests include the computer vision and deep learning. Particularly, he is interested in 3D representation learning.

Xinying Guo is currently working toward the graduate degree in information engineering and media with Nanyang Technological University. She is currently working on her final year project with MM-Lab@NTU, under the supervision of Prof. Ziwei Liu. Her research focuses on motion synthesis.

Lei Yang received the BE degree from Tsinghua University, in 2015, and the PhD degree from the Chinese University of Hong Kong, in 2020, advised by Prof. Dahua Lin and Prof. Xiaou Tang. He is currently a research director with SenseTime Group Inc., under the supervision of Prof. Xiaogang Wang. He has published more than 10 papers on top conferences in relevant fields, including CVPR, ICCV, ECCV, AAAI, SIGGRAPH, and RSS.

Ziwei Liu received the PhD degree from the Chinese University of Hong Kong. He is currently an assistant professor with Nanyang Technological University, Singapore. Previously, he was a senior research fellow with the Chinese University of Hong Kong and a postdoctoral researcher with the University of California, Berkeley. His research interests include revolves around computer vision, machine learning, and computer graphics. He has published extensively on top-tier conferences and journals in relevant fields, including CVPR, ICCV, ECCV, NeurIPS, ICLR, ICML, TPAMI, TOG and Nature - Machine Intelligence. He is the recipient of Microsoft Young Fellowship, Hong Kong PhD Fellowship, ICCV Young Researcher Award and HKSTP Best Paper Award. He also serves as an Area Chair of ICCV, NeurIPS, and ICLR.