This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

StickMotion: Generating 3D Human Motions by Drawing a Stickman

Tao Wang*[1], Zhihua Wu*[2], Qiaozhi He[3], Jiaming Chu[1], Ling Qian[4],
Yu Cheng[5], Junliang Xing[6], Jian Zhao[7,8], Lei Jin†[1]

[1]Beijing University of Posts and Telecommunications, [2]University of Science and Technology of China,
[3]NLP Lab, School of Computer Science and Engineering, Northeastern University, Shenyang, China,
[4]China Mobile (Suzhou) Software Technology Co, Ltd., [5]National University of Singapore,
[6]Tsinghua University, [7]The Institute of AI (TeleAI), China Telecom, [8]Northwestern Polytechnical University

wangtao@bupt.edu.cn, wangzwhu@whu.edu.cn, qiaozhihe2022@outlook.com,
chuiaming886@bupt.edu.cn, caidunbo@cmss.chinamobile.com, e0321276@u.nus.edu,
jlxing@tsinghua.edu.cn, jian_zhao@nwpu.edu.cn, jinlei@bupt.edu.cn

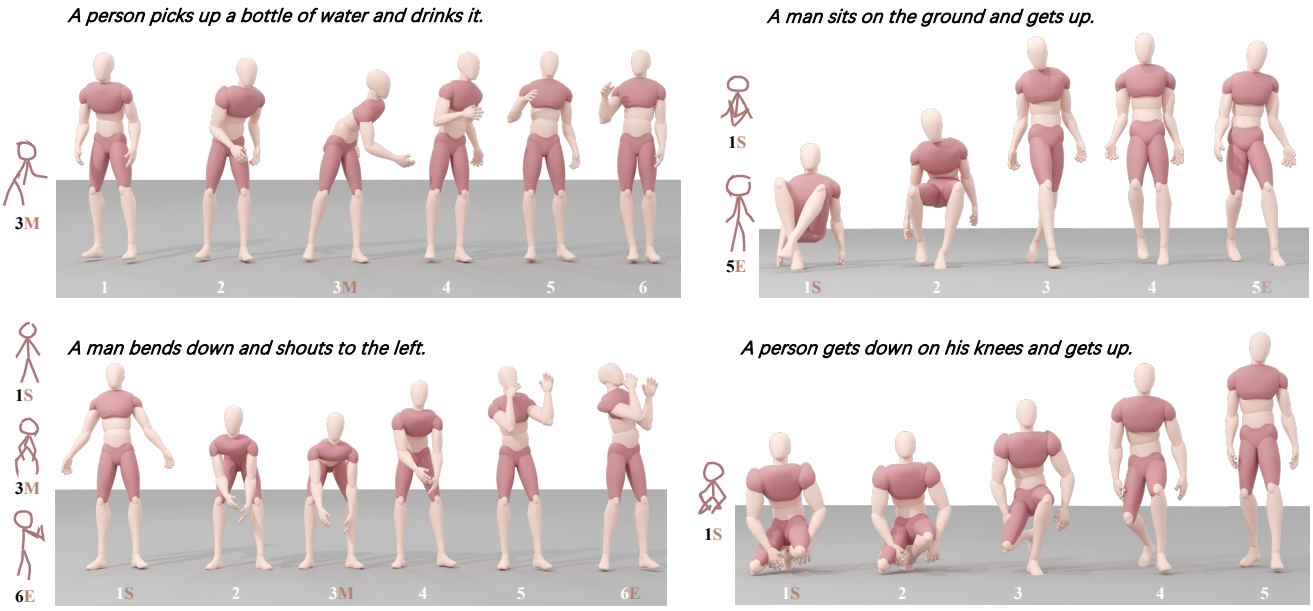


Figure 1. Human motions generated by **StickMotion** under both stickmen and textual description conditions. The black number under the stickman denotes the index of a frame in the generated motion sequences, at which the human pose is generated with regarding the stickman. S, M, and E denote the start, middle, and end of this motion sequence. **These above stickman figures are drawn by users.**

Abstract

*Text-to-motion generation, which translates textual descriptions into human motions, has been challenging in accurately capturing detailed user-imagined motions from simple text inputs. This paper introduces **StickMotion**, an efficient diffusion-based network designed for multi-condition scenarios, which generates desired motions based on traditional text and our proposed stickman conditions for global and local control of these motions, respectively. We address the challenges introduced by the user-friendly stickman from three perspectives: 1) **Data generation.** We develop an algorithm to generate hand-drawn stickmen automatically across different dataset formats. 2) **Multi-condition fusion.** We propose a multi-condition module that integrates into the diffusion process and obtains outputs of all possible condition combinations, reducing computational complexity and enhancing StickMotion's performance compared to conventional approaches with the self-attention module. 3) **Dynamic supervision.** We empower StickMotion to make minor adjustments to the stickman's position*

*Both authors contributed equally. †Lei Jin is corresponding author.

within the output sequences, generating more natural movements through our proposed dynamic supervision strategy. Through quantitative experiments and user studies, sketching stickmen saves users about 51.5% of their time generating motions consistent with their imagination. Our codes, demos, and relevant data will be released in https://github.com/InvertedForest/StickMotion.

1. Introduction

The task of human motion generation [45, 56, 58] has a wide range of applications across diverse fields, including film and television production, virtual reality simulations, the gaming industry, and beyond. Specifically, the popular sub-task of motion generation, text-to-motion task, can generate natural human motion sequences based on language descriptions, freeing 3D animators from manually key-framing 3D character poses.

However, it is evident that a simple description such as "A high kick forward" may not fully satisfy users' detailed imagination of the complex arm gesture shown in Fig. 3. Previous works [29, 55, 57, 58] focus on generating the fine-grained motion with complex textual descriptions. For instance, Flame [29] allows for appending additional textual descriptions to modify the character's motion sequence based on a diffusion model. FineMoGen [57] controls the individual body parts of the 3D character through detailed descriptions. These works enable users to generate motion sequences that closely align with their intentions through enhanced textual descriptions and increased generation iterations. However, the user's demand for more accurate outputs necessitates more detailed textual descriptions. Based on the above, we propose a novel stickman representation to control the details of human motion sequences and mitigate the need for extensive textual descriptions.

Applying stickman condition may compromise the naturalness of the generated motion if stickman figures are directly assigned to the specified frame in the generated motion sequence, as observed in our experiments (details are presented in the supplementary materials). Considering the ease of use, we propose that users only need to specify an approximate position for a stickman (*i.e.*, start, middle, or end of the generated motion sequence). StickMotion dynamically adjusts the position of the stickman slightly based on actual conditions to maintain the naturalness of the generated motion sequence. Consequently, StickMotion should generate motion sequences by considering any combination of four inputs: textual descriptions and three stickman figures positioned at beginning, middle, and end stages of the motion sequence as presented in Fig. 1.

Nevertheless, these above desired functionalities mainly pose three challenges for StickMotion: 1) Data generation. Hand-drawn stickman figures are limited by the drawing style of the annotators and are time-consuming to collect. We propose the Stickman Generation Algorithm (SGA) to randomly generate stickman figures in various styles as shown in Fig. 2. 2) Supervision in Training. The user only needs to specify the stickman's approximate position and the network will adjust the index of the stickman near the specified position. This functionality is achieved by the proposed Dynamic Supervision as shown in Sec. 3.4. 3) Multi-Condition Fusion for the diffusion process. Previous works [7, 56] achieve all possible combinations of two conditions via the mask operation for condition input in self-attention [48, 56] module. Still, it will introduce unnecessary computation when calculating the attention of the masked token. Moreover, the not aligned representation [43] between stickman and text could cause the decrease of performance with the self-attention module as shown in Sec. 4.3. Consequently, we propose an efficient Multi-Condition Module (MCM) to deal with multiple conditions as presented in Sec. 3.3.

The main contributions are summarized as follows:

- To our best knowledge, we are the first to propose a novel stickman representation for motion generation. Furthermore, we design a stickman generation algorithm to generate stickman automatically from existing datasets.
- We propose the Multi-Condition Module for condition fusion in the diffusion process, which reduces computational complexity and enhances the performance of StickMotion compared to the self-attention module.
- We design a Dynamic Supervision strategy to achieve user-friendly interaction, enabling StickMotion to adjust the suitable position of the stickman during generation.
- We evaluate our proposed StickMotion on both the KIT-ML dataset and the HumanML3D dataset, demonstrating its comparable performance to state-of-the-art text-to-motion approaches. In addition, we propose a new metric named StiSim to assess the extent of bias towards the stickman figure in the generated results.

2. Related Work

Human Motion Generation. Human motion generation aims to generate natural sequences of human motion based on various forms of control conditions. This task can be categorized into the following types depending on the conditions. Motion prediction task [6, 22, 37, 49, 59] involves using previous human motion sequences as input to predict the subsequent sequences. This task can be applied to autonomous driving and social security analysis. Action-to-motion task [5, 10, 19, 36, 38, 53] generates human motion sequences based on specified action categories, providing a more direct but coarse-grained control over human motion. Sound-to-motion task can be further divided into music-to-dance [14, 27, 31, 47] and speech-to-gesture [2, 15, 30, 52] tasks, which generate corresponding human motions or ges-

tures in response to audio stimuli simultaneously. Text-to-motion task [1, 9, 16, 20, 21, 45, 56, 58] generates human motion sequences from natural language descriptions like "walk fast and turn right" or "squat down then jump up". However, users may face challenges in precisely controlling the position of each limb with limited textual description alone. Additionally, there are interaction-to-motion tasks that consider interactions between humans and scenes [23, 28, 33, 35, 50] / objects [12, 13, 34, 51] / humans [3, 8, 17, 32, 44], while incorporating generated human motions as reactions from digital lives.

Diffusion Models. In recent years, significant progress has been made in applying deep learning-based generative models, particularly in diffusion models. The proposed denoising diffusion probabilistic model (DDPM) [26, 42] aims to learn the process of restoring original data that has been corrupted by noise, progressively eliminating the noise during inference and resulting in final outputs that closely approximate the distribution of the original data. ADM [11] firstly achieves superior sample quality compared to Generative Adversarial Networks (GAN) [18] with their proposed Denoising Diffusion Implicit Model (DDIM). ADM also incorporates classifier guidance inspired by GANs to control the categories of generated content. Jonathan Ho and Tim Salimans [24] propose a classifier-free guidance technique for reducing sample diversity in diffusion models without relying on a classifier. Currently, diffusion models [4] are employed for generating various data types such as images, videos, text, sound, time series data, *etc*.

3. Method

Overview. StickMotion employs the stickman and the textual descriptions as input modalities. Specifically, users can provide any combination of the four support inputs, including three stickmen positioned near the start, middle, and end of their imaged motion sequence and an accompanying textual description. For instance, users can offer a textual description alongside a middle stickman or opt for a start and end stickman without additional textual context. Furthermore, StickMotion also allows controlling the bias towards generating motions based on either text or stickman conditions through the implementation of the diffusion process.

3.1. Stickman Representation

Using stickman figures obviates the necessity for extensive textual descriptions in generating desired human motion sequences, as shown in Fig. 1. However, it is also crucial to address the challenges associated with the generation, encoding, and application of the stickman figures.

Stickman Generation Algorithm. Due to the lack of hand-draw stickman in existing datasets, we propose a Stickman Generation Algorithm (SGA) based on the 3D coordinates of human joints from existing datasets to generate the hand-draw stickman automatically. Considering the characteristics of human hand-drawing, we have taken into account the following aspects: 1) Stroke smoothness: The smoothness of strokes is influenced by force and individual preferences. Moreover, drawing trajectories may exhibit variations in smoothness across different devices. For instance, strokes made with a mouse tend to be more jittery than those created on an iPad. 2) Misplacement: Inevitably, mistakes in pen placement can lead to global position deviations in these body parts. 3) Scaling: Hand drawings focus on local details while disregarding global information, resulting in size discrepancies among different body parts. The stickmen generated from different datasets are shown in Fig. 2. Moreover, the stickman may appear similar when observing different poses from various angles, so we stipulate that the stickman should be obtained by observing the human pose from the front, *i.e.*, where the line of sight is approximately perpendicular to the pelvic plane of the pose.

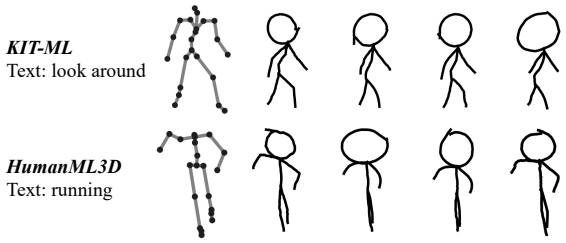


Figure 2. Stickmen generated by Stickman Generation Algorithm on the KIT-ML [39] and HumanML3D [20] dataset.

Information Encoding. A trade-off exists between user convenience and efficient computing in handling stickman information. To accurately reconstruct the user's drawing, it is recommended to collect a minimum of 200 (estimated based on visualization) 2D coordinate points and their connectivity information when users can draw freely. However, this approach necessitates significant memory and computational resources due to the involvement of at least 200^2 interactions among these data points through the network. To mitigate resource consumption, we propose a guideline where users should depict six one-stroke lines representing the head, torso, and four limbs in any order. Each line is individually encoded and subsequently interacted with each other using a simple transformer encoder [48] to obtain an embedding for a stickman figure. This method effectively reduces computational requirements while improving the accuracy of stickman recognition.

3.2. Diffusion-based Motion Generation

Diffusion-based works have demonstrated excellent performance in the field of human motion generation. We selected it as the base model for StickMotion since it can control the bias towards generating motions based on either textual description or stickman conditions. Diffusion models aim

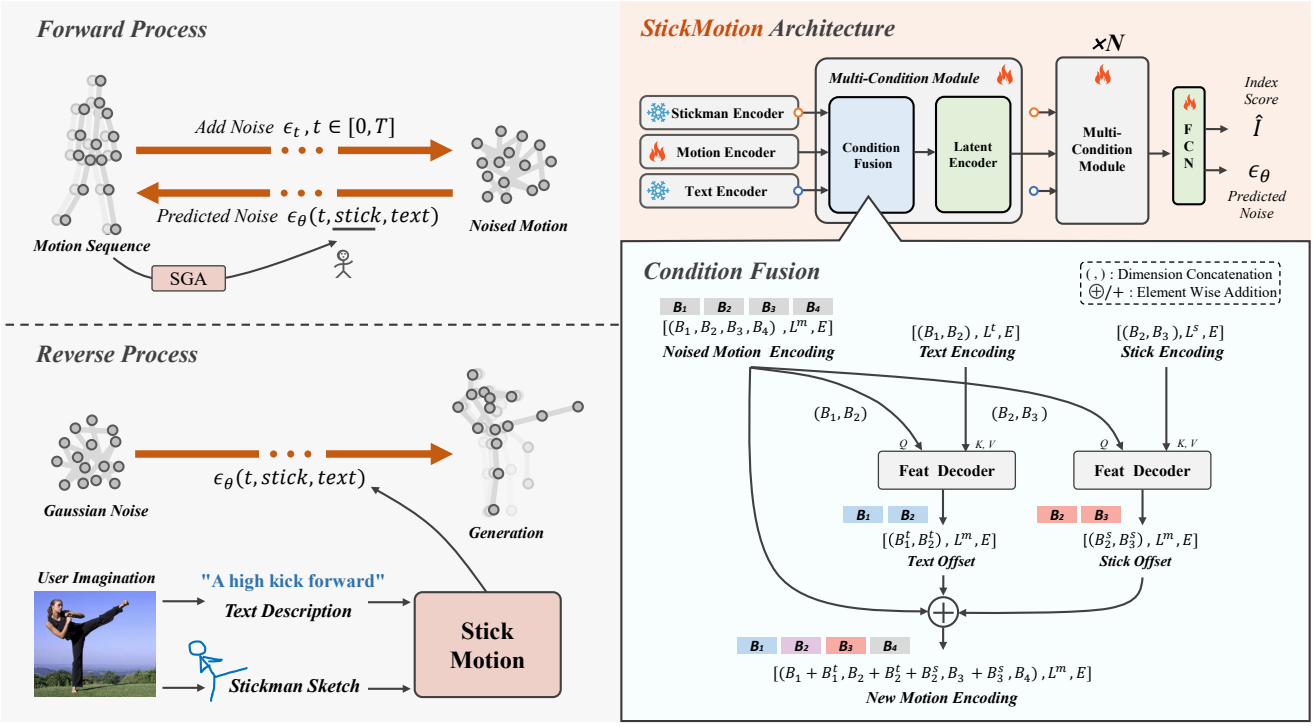


Figure 3. The StickMotion framework consists of the diffusion process on the left and the network structure on the right. 1) The diffusion process is divided into two components: the forward process and the reverse process. In the forward process, original motions are artificially augmented with Gaussian noise and fed into StickMotion to facilitate its prediction of the added noise based on text from the dataset and stickman generated by actual motion through the Stickman Generation Algorithm (SGA). In the reverse process, the user's textual descriptions and stickman figures are inputted into StickMotion, enabling the gradual generation of motion sequences with its predicted noise. 2) Regarding the structure of StickMotion, both the stickman encoder and text encoder remain frozen while other components participate in training. After encoding the input data, it undergoes multiple Multi-Condition Modules (MCM) to obtain predictions for noise, which are then utilized in generating motion sequences during the reverse process.

to estimate the model distribution $q_\theta(x_0)$ based on samples drawn from the motion distribution $q(x_0)$. Here, θ represents the learnable parameters in StickMotion.

As shown in Fig. 3, the diffusion models decompose this task into a forward process and a reverse process, which involves adding noise to actual motion and restoring the motion from noise, respectively. In the forward process, Gaussian noise $\epsilon_t \sim \mathcal{N}(0, \mathbf{I})$ is gradually added into the actual motion x_0 over a time interval from $t = 0$ to $t = T$, with an increasing sequence β_t serving as its weight, as shown in Equ. 1.

$$\begin{aligned} q(\mathbf{x}_{1:T}|\mathbf{x}_0) &= \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}), \\ q(\mathbf{x}_t|\mathbf{x}_{t-1}) &= \mathcal{N}(\mathbf{x}_t; \sqrt{\alpha_t}\mathbf{x}_{t-1}, (1-\alpha_t)\mathbf{I}). \end{aligned} \tag{1}$$

This formula is equivalent to $\mathbf{x} = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t$, where $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$. In this way, we can directly get x_t from x_0 without the intermediate variables. The forward process above is used for the training of StickMotion. To achieve a balance between stickman and text conditions for the final generation during inference, we employ the classifier-free diffusion guidance [25] for the diffusion process, and the supervision formula is set as

$$\mathbb{E}_{\epsilon_t, t, x_0}[||\epsilon_t - \epsilon_\theta(\mathbf{x}_t, t, L, C(stick), C(text))||^2], \tag{2}$$

where L represents the length of motion sequences. The stickman condition considered in training is denoted as $C(stick)$ with a probability of p^c_{stick}, while the text condition is represented as $C(text)$. $\epsilon_\theta(\cdot)$ denotes the predicted noise from StickMotion. We set $p^c_{stick} = p^c_{text} = 0.7$ during the training process.

In the reverse process, the well-trained StickMotion gradually removes the noise from X_T to get the real motion sequence. According to DDPM [26], x_{t-1} can be sampled from $p_\theta(x_{t-1}|x_t)$ by the formula

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta) + \sigma_t, \tag{3}$$

where ϵ_θ is the predicted noise from StickMotion, and σ_t is

a weighted Gaussian noise. In the reverse process from T to 0, we can get outputs that are biased to different combinations of conditions as follows.

$$\hat{\epsilon_\theta} = w_1 \cdot \epsilon_\theta(stick, text) + w_2 \cdot \epsilon_\theta(stick,) + w_3 \cdot \epsilon_\theta(text,) + w_4 \cdot \epsilon_\theta(,). \quad (4)$$

Adhering to the principle $w_1 + w_2 + w_3 + w_4 = 1$ [24] that preserves output statistics, we propose an efficient condition mixture by considering the characteristics of stickman and text conditions. Initially, the approximate motion sequence is determined during the beginning stage of time interval $t \in [T, T/10]$, the condition mixture follows the formula $(w_1 = w, w_2 = \hat{w}, w_3 = w - \hat{w}, w_4 = 1 - 2 \cdot w)$. Here, 1) constant $w > 1$ [24] adjusts the condition sampling strength; 2) $w_1 = w$ ensures that the fusion of stickman and text is harmonious. 3) $p(\hat{w} = w) + p(\hat{w} = 0) = 1$, with $p(\hat{w} = w)$ controlling the preference of generated motion for the stickman condition. 4) $w_4 = 1 - 2 \cdot w$ controls the constant distribution of the output. In the final stage $(t \in [T/10, 0])$, we set $(w_1 = 1, w_{2,3,4} = 0)$ to use all conditions to further refine the preliminary result from the beginning stage. Finally, a motion sequence corresponding to the stickman condition and text condition is generated in the reverse process. More details can be found in Sec. 4.3 and the supplementary materials.

3.3. Architecture of StickMotion

The diffusion model offers StickMotion a straightforward training and controllable generation process. However, designing a network architecture that efficiently handles both stickman and text conditions for the diffusion process is equally crucial. As depicted in the right section of Fig. 3, StickMotion comprises three input encoders and Multi-Condition Modules (MCM) to generate the motion sequence's final predicted noise and stickman index score. The input encoders transform the noisy motion, text, and stickman into vectors, which are subsequently fed into MCMs to produce the final outputs under multiple condition combinations, *i.e.*, $(text), (text, stick), (stick)$, and $()$.

Input. The input data consists of noisy motion sequences, stickman figures, and text, which are encoded to the shape of $[L^m, E], [L^s, E]$ and $[L^t, E]$, respectively. Here, L denotes the length of the input encoding, and E represents the dimension of each token in the encoding. We employ a simple linear layer to encode the motion sequences. We utilize CLIP ViT-B/32 [40], which consists of 154M parameters for text encoding. Regarding the stickman figures, we leverage a standard transformer encoder with six encoder layers. However, based on experimental findings, it has been observed that pre-training and freezing the stickman encoder significantly enhances StickMotion's performance. Consequently, we train a pair of auto-encoder models for the stickman condition: a stickman encoder and a feature-to-pose decoder. The stickman encoder encodes the stickman into a feature embedding. At the same time, the feature-to-pose decoder reconstructs the original pose from this feature vector to ensure the preservation of pose information.

Multi-Condition Module. Condition combinations are essential for the diffusion process to fuse these conditions. These combinations are implemented through the Multi-Condition Module (MCM) in StickMotion. Within each MCM, a Condition Fusion module is utilized to incorporate stickman and text conditions into the motion encoding in the latent space. Subsequently, the modified motion encoding undergoes re-encoding by the Latent Encoder for further fusion. Specifically, we partition all data along the batch dimension into four segments (B_1, B_2, B_3, B_4), representing four combinations of text and stickman conditions, *i.e.*, $(text), (text, stick), (stick)$, and $()$. As shown in Fig. 3, in the Condition Diffusion, two standard transformer decoder layers named Feat Decoder consider solely the text input and stickman input for batches (B_1, B_2) and (B_2, B_3), respectively. By summing up these predicted offsets with their corresponding motion encoding along the batch dimension, we obtain new motion encoding for three condition combinations with only two Feat Decoders as shown in Fig. 3. Sequently, the new motion encoding undergoes re-encoding in the Latent Encoder to facilitate further information fusion. Compared to conventional approaches with self-attention module that accomplish all condition combinations by the partial mask operation for condition inputs, MCM reduces computational complexity and enhances StickMotion's performance as demonstrated in Sec. 4.3. Additionally, we employ efficient attention [41, 56] for both feat decoder and latent encoder to further reduce computational complexity.

Output. StickMotion outputs predicted noise for noisy motions and an index score for the position of input stickmen. The index scores are crucial for user interaction and training supervision of stickmen: 1) The index score can explicitly supervise and minimize the distance between the dynamically assigned pose and stickman during training, thereby making the results closer to the stickman as presented in Sec. 3.4. 2) The index score indicates the position of the stickman in the generated motion sequences, enabling users to decide whether to adjust the generation towards the stickman figures or textual descriptions freely.

3.4. Dynamic Supervision

Different from previous works, StickMotion faces an additional challenge in determining the stickman indexes of the generated motion sequence. Allowing users to specify the index of each stickman manually would introduce an extra burden on them to consider suitable indexes, potentially resulting in less naturally generated results due to improper positioning of multiple stickman figures. To address this is-

sue, we have designated three positions, start, middle, and end, from which users can choose. The network will dynamically adjust the indexes of stickmen near these specified positions to optimize the generation's naturalness and adherence to the textual description.

Specifically, we first randomly sample human poses at the beginning, middle, and end positions within the ranges $[0, 1/4 \cdot L]$, $[3/8 \cdot L, 5/8 \cdot L]$, and $[3/4 \cdot L, L]$ (L represents the length of motion sequence), respectively. Then, we randomly create a mask M for each stickman to mimic users' random input. The overall loss function is divided into two parts as follows.

$$\begin{aligned} \mathcal{L}_{\text{index}} &= M \cdot \sum_{l=0}^{L} softmax(\hat{I}_l) \cdot ||\hat{x}_l(stick, *) - x_i||^2, \\ \mathcal{L}_{\text{motion}} &= \sum_{l=0}^{L} ||\hat{x}_l(*, *) - x_l||^2, \\ \mathcal{L}_{\text{total}} &= \mathcal{L}_{\text{index}}^{start} + \mathcal{L}_{\text{index}}^{middle} + \mathcal{L}_{\text{index}}^{end} + \mathcal{L}_{\text{motion}}, \end{aligned} \tag{5}$$

The predicted index score for the stickman at frame l in the generated motion sequence is denoted as $\hat{I}_l$. The highest score at frame l indicates that StickMotion considers the input stickman to belong to that specific frame. $\hat{x}(stick, *)$ refers to the generated motion sequence under condition combinations of $(stick, text)$ and $(stick,)$. Predicted poses at frame l with all condition combinations are denoted as $\hat{x}_t(*, *)$. Here, x_l represents the pose of the real motion sequence at frame l, while a stickman is generated by pose x_i from the ground-truth motion sequence through the Stickman Generation Algorithm. In this way, $\mathcal{L}_{\text{index}}$ ensures that the assigned pose $\hat{x}_l(stick, *)$, which has the highest score in the output sequence, is supervised by ground-truth pose x_i. The supervisions for start, middle, and end stickmen are denoted as $\mathcal{L}_{\text{index}}^{start}$, $\mathcal{L}_{\text{index}}^{middle}$, and $\mathcal{L}_{\text{index}}^{end}$ respectively. The supervision for diffusion process in StickMotion is represented by $\mathcal{L}_{\text{motion}}$ as shown in Equ. 2.

4. Experiments

4.1. Experiment Settings

Dataset and Metrics. We conduct experiments on two prominent datasets of human motion generation, namely the KIT-ML dataset [39] and the HumanML3D dataset [20]. The same evaluation of Guo et al. [20] is adopted, so we can comprehensively compare with existing text-to-motion methods. This evaluation involves encoding the input text and generated motion sequence into embeddings through pre-trained contrastive quantitative assessment models, and then participate in the calculating the following metrics. *R Precision.* Given a prediction motion sequence, its text and 31 other irrelevant texts from the test set are combined into a set, and the Top-k accuracy between the motion and text set is calculated after passing through the pre-trained motion-text contrastive models. *Frechet Inception Distance (FID).* Motion embeddings are generated by ground-truth and generated motion sequences through contrastive models. Then, the difference between the two batches of embedding distribution is calculated. FID is related to generation quality but is limited by the performance of contrastive models. *Multimodal Distance (MM Dist).* The Euclidean distance between the motion embedding and its text embedding. *Diversity.* The variability and richness of the generated motion sequences. *Multimodality.* The variance of generated motion sequences given a specified text.

Implementation Details. StickMotion is trained with 4 A800 GPUs with a batch size of 1024, while 80 dataloader workers are adopted to generate stickmen through SGA. For the diffusion process, we set the noise steps $T = 1000$. And $\alpha_t, t \in [0, T]$ ranges from 0.9999 to 0.9800. The trainable StickMotion model comprises four MCMs for the KIT-ML dataset and five MCMs for the HumanML3D dataset, with parameter counts of 43M and 62M respectively.

4.2. Quantitative Analysis

Comparison with SOTA Methods. We follow the same evaluation of text-to-motion methods to demonstrate the performance of StickMotion based on both the KIT-ML dataset and the HumanML3D dataset. Additionally, the generation approach of the stickman used in the evaluation is similar to that of the training process as described in Sec. 3.4, and all stickmen in three positions of the ground-truth sequences are utilized. In contrast to conventional approaches, StickMotion requires alignment with both textual descriptions and stickman figures. As mentioned in Sec. 4.3, these two conditions have an adversarial relationship that can negatively impact StickMotion's performance. Hence, we adjusted the reverse process mentioned in Sec. 3.2 by setting $p(\hat{w} = w) = 20\%$ and $(w_1 = 1, w_2 = 0, w_3 = 0, w_4 = 0)$ to bias the generated results towards the stickman condition. Our approach shows excellent performance compared to previous text-to-motion works as shown in Tab. 1 and Tab. 2. However, it is crucial to assess how much stickman information was incorporated into the generated motions. Therefore, we propose a new metric called Stickman Similarity, which will be discussed as follows.

New Metric: Stickman Similarity (StiSim). In the evaluation, the stickman is generated by sampling a pose from the ground-truth sequence, while the corresponding poses in the generated sequences are determined by the index scores generated from StickMotion. Thus, the Euclidean distance (referred to as *StickmanDistance*) between the sampled ground-truth pose and the generated pose with highest index score indicates how much StickMotion has learned from the input stickman. Moreover, we use *MeanDistance* to denote the mean of Euclidean distances between sam-

Table 1. Comparison on the HumanML3D test set. We mark the best result as red and the second best one as blue.

Methods	R Precision ↑			FID ↓	MM Dist ↓	Diversity ↑	MultiModality ↑
	Top1	Top2	Top3				
Real motions	$0.511^{\pm.003}$	$0.703^{\pm.003}$	$0.797^{\pm.002}$	$0.002^{\pm.000}$	$2.974^{\pm.008}$	$9.503^{\pm.065}$	-
Guo et al. [20]	$0.457^{\pm.002}$	$0.639^{\pm.003}$	$0.740^{\pm.003}$	$1.067^{\pm.002}$	$3.340^{\pm.008}$	$9.188^{\pm.002}$	$2.090^{\pm.083}$
MDM [46]	-	-	$0.611^{\pm.007}$	$0.544^{\pm.044}$	$5.566^{\pm.027}$	$9.559^{\pm.086}$	$2.799^{\pm.072}$
MotionDiffuse [55]	$0.491^{\pm.001}$	$0.681^{\pm.001}$	$0.782^{\pm.001}$	$0.630^{\pm.001}$	$3.113^{\pm.001}$	$9.410^{\pm.049}$	$1.553^{\pm.042}$
T2M-GPT [54]	$0.491^{\pm.003}$	$0.680^{\pm.003}$	$0.775^{\pm.002}$	$0.116^{\pm.004}$	$3.118^{\pm.011}$	$9.761^{\pm.081}$	$1.856^{\pm.011}$
ReMoDiffuse [56]	$0.510^{\pm.005}$	$0.698^{\pm.006}$	$0.795^{\pm.004}$	$0.103^{\pm.004}$	$2.974^{\pm.016}$	$9.018^{\pm.075}$	$1.795^{\pm.043}$
StickMotion (Ours)	$0.518^{\pm.007}$	$0.702^{\pm.003}$	$0.797^{\pm.005}$	$0.107^{\pm.003}$	$2.953^{\pm.021}$	$9.239^{\pm.066}$	$2.256^{\pm.051}$

Table 2. Comparison on the KIT-ML test set.

Methods	R Precision ↑			FID ↓	MM Dist ↓	Diversity ↑	MultiModality ↑
	Top1	Top2	Top3				
Real motions	$0.424^{\pm.005}$	$0.649^{\pm.006}$	$0.779^{\pm.006}$	$0.031^{\pm.004}$	$2.788^{\pm.012}$	$11.08^{\pm.097}$	-
Guo et al. [20]	$0.370^{\pm.005}$	$0.569^{\pm.007}$	$0.693^{\pm.007}$	$2.770^{\pm.109}$	$3.401^{\pm.008}$	$10.91^{\pm.119}$	$1.482^{\pm.065}$
MDM [46]	-	-	$0.396^{\pm.004}$	$0.497^{\pm.021}$	$9.191^{\pm.022}$	$10.847^{\pm.109}$	$1.907^{\pm.214}$
MotionDiffuse [55]	$0.417^{\pm.004}$	$0.621^{\pm.004}$	$0.739^{\pm.004}$	$1.954^{\pm.062}$	$2.958^{\pm.005}$	$11.10^{\pm.143}$	$0.730^{\pm.013}$
T2M-GPT [54]	$0.416^{\pm.006}$	$0.627^{\pm.006}$	$0.745^{\pm.006}$	$0.514^{\pm.029}$	$3.007^{\pm.023}$	$10.921^{\pm.108}$	$1.570^{\pm.039}$
ReMoDiffuse [56]	$0.427^{\pm.014}$	$0.641^{\pm.004}$	$0.765^{\pm.055}$	$0.155^{\pm.006}$	$2.814^{\pm.012}$	$10.80^{\pm.105}$	$1.239^{\pm.028}$
StickMotion (Ours)	$0.430^{\pm.017}$	$0.654^{\pm.010}$	$0.775^{\pm.043}$	$0.141^{\pm.008}$	$2.763^{\pm.018}$	$10.94^{\pm.178}$	$1.457^{\pm.033}$

pled ground-truth poses and all generated poses. Therefore, *StickmanSimilarity* (StiSim) can be calculated as 1 - (*StickmanDistance* / *MeanDistance*). For HumanML3D dataset [20] and KIT-ML dataset [39], values for *StickmanDistance*, *MeanDistance*, and *StickmanSimilarity* are (0.24, 0.41, 41.5%) and (0.27, 0.47, 42.6%), respectively.

4.3. Ablation Study

Analysis on StickMotion Modules. We first conduct ablation experiments on the Condition Fusion and Latent Encoder modules in the Multi-Condition Model (MCM) using the KIT-ML dataset. We replace Condition Fusion with conventional self-attention [48] while ensuring that MCM still supports multi-condition combinations. Additionally, we calculate the Flops of a MCM with a batch size ($B_1 = B_2 = B_3 = B_4 = 256$). As shown in Tab. 3, combining Condition Fusion and Latent Encoder achieves better performance compared to conventional approaches with self-attention module for condition combinations through mask operations, which introduce unnecessary computations when calculating attention for masked conditions.

Analysis on Condition Mixture. As shown in Equ. 4, the condition mixture controls how to mix the outputs based on four combinations of stickman and text conditions, resulting in a bias towards either stickman or text in the generated output. (The combinations of three stickmen in different positions will be discussed next, since we only focus on the these conditions at a higher level in this section.) As discussed in Sec. 3.2, $p(\hat{w} = w)$ determines the extent to which the coarse generations are biased towards the stickman condition during the initial stage of the reverse process. Here (w_1, w_2, w_3, w_4) refines these coarse generations based on these condition combinations in the final stage. According to Tab. 4, there exists an adversarial relationship between stickman and text conditions.

Table 3. Ablation study on Condition Fusion and Latent Encoder for the training / forward process.

Self-Attention	ConditionFusion	LatentEncoder	FID↓	TFlops↓
√			0.31	0.46
√		√	0.18	0.71
	√		0.25	**0.28**
	√	√	**0.14**	0.43

Analysis on Input Combination. Users can provide diverse combinations of inputs, encompassing the textual description and three stickmen positioned at different stages

Table 4. Ablation study on the condition mixture for the inference / reverse process.

$p(\hat{w}=w)$	(w_1, w_2, w_3, w_4)	FID↓	StiSim↑
50%	(0, 1, 0, 0)	0.147	44.8%
50%	(0, 0, 1, 0)	**0.135**	26.1%
80%	(1, 0, 0, 0)	0.146	**51.3%**
20%	(1, 0, 0, 0)	0.141	42.6%

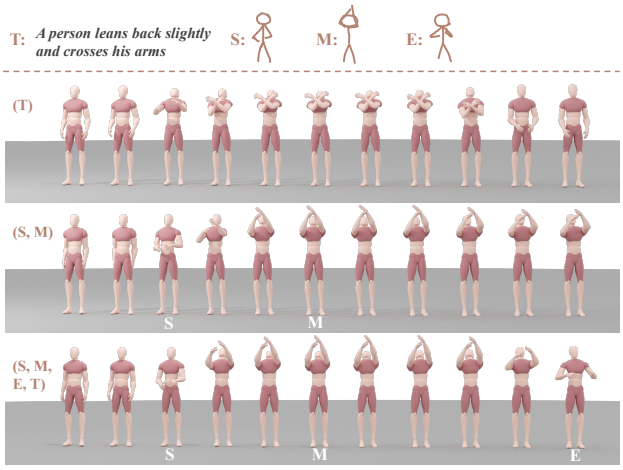


Figure 4. Visualization with various input combinations.

(start, middle, or end) of the motion sequence. The generation quality of those combinations under the settings of $p(\hat{w}=w)=20\%$ and $(w_1=1, w_2=0, w_3=0, w_4=0)$ can be observed in Tab. 5. It should be noticed that the text description has the most significant impact on the performance of StickMotion because the text-to-motion evaluation solely takes into account the semantic information of the generated motion sequences. We also show a visualization of partial combinations in Fig. 4.

User Study. We recruited 20 unrelated volunteers to participate in the user study of StickMotion and the traditional text-to-motion work ReMoDiffuse [56]. These participants were instructed to imagine a specific human motion lasting about 10 seconds (equivalent to 120 frames) and subsequently provide an overall textual description (A), detailed textual description (B), and a stickman figure (S) representing a characteristic pose within the imagined motion. The combination of (A, B) was inputted into ReMoDiffuse, while the combination of (A, S) was inputted into StickMotion. Participants then rated the generated results from both methods on a scale of 0 to 10. The final results regarding time consumption and user scores are shown in Tab. 6. Our StickMotion saves users approximately 51.5% of their time in generating motions consistent with their imagination.

Table 5. Ablation study on combinations of conditions in the generate process. There are four inputs, including stickmen in the position of "Start", "Middle", and "End" along with the textual description.

Start	Middle	End	Text	FID↓
				4.292
√				3.291
	√			3.132
		√		3.450
			√	0.164
√		√		3.548
√		√	√	0.189
	√		√	0.157
√	√	√		2.782
√	√	√	√	**0.141**

Table 6. Comparison between stickman & text-to-motion and text-to-motion task. "TA" and "TB" represent the time cost for overall and detailed descriptions, respectively, while "TS" denotes the time required for drawing a stickman. Additionally, "TI" represents the inference time of the utilized model.

Method	TA	TB	TS	TI	TotalTime↓	Score↑
ReMoDiffuse [56]	8.1	24.5	-	1.2	33.8	7.3
StickMotion	8.1	-	7.7	0.7	16.4	8.5

5. Conclusion

This paper presents a novel stickman condition for motion generation to address users' detailed requirements for body posture through simple textual descriptions. To effectively leverage the stickman condition, we initially developed an algorithm to generate hand-drawn stickmen based on joint positions from existing datasets. Subsequently, we introduce StickMotion, a diffusion-based approach with multi-condition modules, to create motion sequences that align with textual descriptions and stickmen as conditions while maintaining lower computational complexity. Finally, we propose a new StiSim metric for measuring stickman similarity in traditional text-to-motion evaluation. Our objective is to contribute a novel stickman condition to the text-to-motion generation and a Multi-Condition Module for multi-condition scenarios, simplifying and accelerating the process for creators in obtaining desired human motions.

Acknowledgment

The paper is supported by National Natural Science Foundation of China No.62472046, No.82274685, No.62476224, the Fundamental Research Funds for the Central Universities No.500422813.

References

[1] Chaitanya Ahuja and Louis-Philippe Morency. Language2pose: Natural language grounded pose forecasting. In *3DV*, pages 719–728. IEEE, 2019. 3

[2] Tenglong Ao, Zeyi Zhang, and Libin Liu. Gesturediffuclip: Gesture diffusion model with clip latents. *TOG*, 42(4):1–18, 2023. 2

[3] Zhongang Cai, Jianping Jiang, Zhongfei Qing, Xinying Guo, Mingyuan Zhang, Zhengyu Lin, Haiyi Mei, Chen Wei, Ruisi Wang, Wanqi Yin, et al. Digital life project: Autonomous 3d characters with social intelligence. In *CVPR*, pages 582–592, 2024. 3

[4] Hanqun Cao, Cheng Tan, Zhangyang Gao, Yilun Xu, Guangyong Chen, Pheng-Ann Heng, and Stan Z Li. A survey on generative diffusion models. *TKDE*, 2024. 3

[5] Pablo Cervantes, Yusuke Sekikawa, Ikuro Sato, and Koichi Shinoda. Implicit neural representations for variable length human motion generation. In *ECCV*, pages 356–372. Springer, 2022. 2

[6] Ling-Hao Chen, Jiawei Zhang, Yewen Li, Yiren Pang, Xiaobo Xia, and Tongliang Liu. Humanmac: Masked motion completion for human motion prediction. In *ICCV*, pages 9544–9555, 2023. 2

[7] Wenhu Chen, Hexiang Hu, Chitwan Saharia, and William W Cohen. Re-imagen: Retrieval-augmented text-to-image generator. *arXiv preprint arXiv:2209.14491*, 2022. 2

[8] Baptiste Chopin, Hao Tang, and Mohamed Daoudi. Bipartite graph diffusion model for human interaction generation. In *WACV*, pages 5333–5342, 2024. 3

[9] Jieming Cui, Tengyu Liu, Nian Liu, Yaodong Yang, Yixin Zhu, and Siyuan Huang. Anyskill: Learning open-vocabulary physical skill for interactive agents. In *CVPR*, pages 852–862, 2024. 3

[10] Bruno Degardin, Joao Neves, Vasco Lopes, Joao Brito, Ehsan Yaghoubi, and Hugo Proença. Generative adversarial graph convolutional networks for human action synthesis. In *WACV*, pages 1150–1159, 2022. 2

[11] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *NeuIPS*, 34:8780–8794, 2021. 3

[12] Christian Diller and Angela Dai. Cg-hoi: Contact-guided 3d human-object interaction generation. In *CVRP*, pages 19888–19901, 2024. 3

[13] Chen Gao, Si Liu, Defa Zhu, Quan Liu, Jie Cao, Haoqian He, Ran He, and Shuicheng Yan. Interactgan: Learning to generate human-object interaction. In *ACM MM*, pages 165–173, 2020. 3

[14] Xin Gao, Li Hu, Peng Zhang, Bang Zhang, and Liefeng Bo. Dancemeld: Unraveling dance phrases with hierarchical latent codes for music-to-dance synthesis. *arXiv preprint arXiv:2401.10242*, 2023. 2

[15] Saeed Ghorbani, Ylva Ferstl, Daniel Holden, Nikolaus F Troje, and Marc-André Carbonneau. Zeroeggs: Zero-shot example-based gesture generation from speech. In *Computer Graphics Forum*, pages 206–216. Wiley Online Library, 2023. 2

[16] Anindita Ghosh, Noshaba Cheema, Cennet Oguz, Christian Theobalt, and Philipp Slusallek. Synthesis of compositional animations from textual descriptions. In *ICCV*, pages 1396–1406, 2021. 3

[17] Anindita Ghosh, Rishabh Dabral, Vladislav Golyanik, Christian Theobalt, and Philipp Slusallek. Remos: Reactive 3d motion synthesis for two-person interactions. *arXiv preprint arXiv:2311.17057*, 2023. 3

[18] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 3

[19] Chuan Guo, Xinxin Zuo, Sen Wang, Shihao Zou, Qingyao Sun, Annan Deng, Minglun Gong, and Li Cheng. Action2motion: Conditioned generation of 3d human motions. In *ACM MM*, pages 2021–2029, 2020. 2

[20] Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *CVPR*, pages 5152–5161, 2022. 3, 6, 7

[21] Chuan Guo, Yuxuan Mu, Muhammad Gohar Javed, Sen Wang, and Li Cheng. Momask: Generative masked modeling of 3d human motions. In *CVPR*, pages 1900–1910, 2024. 3

[22] Wen Guo, Yuming Du, Xi Shen, Vincent Lepetit, Xavier Alameda-Pineda, and Francesc Moreno-Noguer. Back to mlp: A simple baseline for human motion prediction. In *WACV*, pages 4809–4819, 2023. 2

[23] Mohamed Hassan, Partha Ghosh, Joachim Tesch, Dimitrios Tzionas, and Michael J Black. Populating 3d scenes by learning human-scene interaction. In *CVPR*, pages 14708–14718, 2021. 3

[24] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 3, 5

[25] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 4

[26] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeuIPS*, 33:6840–6851, 2020. 3, 4

[27] Ruozi Huang, Huang Hu, Wei Wu, Kei Sawada, Mi Zhang, and Daxin Jiang. Dance revolution: Long-term dance generation with music via curriculum learning. *arXiv preprint arXiv:2006.06119*, 2020. 2

[28] Siyuan Huang, Zan Wang, Puhao Li, Baoxiong Jia, Tengyu Liu, Yixin Zhu, Wei Liang, and Song-Chun Zhu. Diffusion-based generation, optimization, and planning in 3d scenes. In *CVPR*, pages 16750–16761, 2023. 3

[29] Jihoon Kim, Jiseob Kim, and Sungjoon Choi. Flame: Free-form language-based motion synthesis & editing. In *AAAI*, pages 8255–8263, 2023. 2

[30] Taras Kucherenko, Dai Hasegawa, Gustav Eje Henter, Naoshi Kaneko, and Hedvig Kjellström. Analyzing input and output representations for speech-driven gesture generation. In *Proceedings of the 19th ACM International Conference on Intelligent Virtual Agents*, pages 97–104, 2019. 2

[31] Buyu Li, Yongchi Zhao, Shi Zhelun, and Lu Sheng. Danceformer: Music conditioned 3d dance generation with parametric motion transformer. In *AAAI*, pages 1272–1279, 2022. 2

[32] Han Liang, Wenqian Zhang, Wenxuan Li, Jingyi Yu, and Lan Xu. Intergen: Diffusion-based multi-human motion generation under complex interactions. *International Journal of Computer Vision*, pages 1–21, 2024. 3

[33] Donggeun Lim, Cheongi Jeong, and Young Min Kim. Mammos: Mapping multiple human motion with scene understanding and natural interactions. In *ICCV*, pages 4278–4287, 2023. 3

[34] Pei Lin, Sihang Xu, Hongdi Yang, Yiran Liu, Xin Chen, Jingya Wang, Jingyi Yu, and Lan Xu. Handdiffuse: Generative controllers for two-hand interactions via diffusion models. *arXiv preprint arXiv:2312.04867*, 2023. 3

[35] Xinpeng Liu, Haowen Hou, Yanchao Yang, Yong-Lu Li, and Cewu Lu. Revisit human-scene interaction via space occupancy. *arXiv preprint arXiv:2312.02700*, 2023. 3

[36] Qiujing Lu, Yipeng Zhang, Mingjian Lu, and Vwani Roychowdhury. Action-conditioned on-demand motion generation. In *ACM MM*, pages 2249–2257, 2022. 2

[37] Tiezheng Ma, Yongwei Nie, Chengjiang Long, Qing Zhang, and Guiqing Li. Progressively generating better initial guesses towards next stages for high-quality human motion prediction. In *CVPR*, pages 6437–6446, 2022. 2

[38] Mathis Petrovich, Michael J Black, and Gül Varol. Action-conditioned 3d human motion synthesis with transformer vae. In *ICCV*, pages 10985–10995, 2021. 2

[39] Matthias Plappert, Christian Mandery, and Tamim Asfour. The kit motion-language dataset. *Big data*, 4(4):236–252, 2016. 3, 6, 7

[40] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PMLR, 2021. 5

[41] Zhuoran Shen, Mingyuan Zhang, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Efficient attention: Attention with linear complexities. In *WACV*, pages 3531–3539, 2021. 5

[42] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, pages 2256–2265. PMLR, 2015. 3

[43] Ilia Sucholutsky, Lukas Muttenthaler, Adrian Weller, Andi Peng, Andreea Bobu, Been Kim, Bradley C Love, Erin Grant, Iris Groen, Jascha Achterberg, et al. Getting aligned on representational alignment. *arXiv preprint arXiv:2310.13018*, 2023. 2

[44] Mikihiro Tanaka and Kent Fujiwara. Role-aware interaction generation from textual description. In *ICCV*, pages 15999–16009, 2023. 3

[45] Guy Tevet, Brian Gordon, Amir Hertz, Amit H Bermano, and Daniel Cohen-Or. Motionclip: Exposing human motion generation to clip space. In *ECCV*, pages 358–374. Springer, 2022. 2, 3

[46] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H. Bermano. Human motion diffusion model, 2022. 7

[47] Jonathan Tseng, Rodrigo Castellon, and Karen Liu. Edge: Editable dance generation from music. In *CVPR*, pages 448–458, 2023. 2

[48] A Vaswani. Attention is all you need. *NeuIPS*, 2017. 2, 3, 7

[49] Xinshun Wang, Qiongjie Cui, Chen Chen, and Mengyuan Liu. Gcnext: Towards the unity of graph convolutions for human motion prediction. In *AAAI*, pages 5642–5650, 2024. 2

[50] Zeqi Xiao, Tai Wang, Jingbo Wang, Jinkun Cao, Wenwei Zhang, Bo Dai, Dahua Lin, and Jiangmiao Pang. Unified human-scene interaction via prompted chain-of-contacts. *arXiv preprint arXiv:2309.07918*, 2023. 3

[51] Sirui Xu, Zhengyuan Li, Yu-Xiong Wang, and Liang-Yan Gui. Interdiff: Generating 3d human-object interactions with physics-informed diffusion. In *ICCV*, pages 14928–14940, 2023. 3

[52] Youngwoo Yoon, Bok Cha, Joo-Haeng Lee, Minsu Jang, Jaeyeon Lee, Jaehong Kim, and Geehyuk Lee. Speech gesture generation from the trimodal context of text, audio, and speaker identity. *TOG*, 39(6):1–16, 2020. 2

[53] Ping Yu, Yang Zhao, Chunyuan Li, Junsong Yuan, and Changyou Chen. Structure-aware human-action generation. In *ECCV*, pages 18–34. Springer, 2020. 2

[54] Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Yong Zhang, Hongwei Zhao, Hongtao Lu, Xi Shen, and Ying Shan. Generating human motion from textual descriptions with discrete representations. In *CVPR*, pages 14730–14740, 2023. 7

[55] Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. Motiondiffuse: Text-driven human motion generation with diffusion model. *arXiv preprint arXiv:2208.15001*, 2022. 2, 7

[56] Mingyuan Zhang, Xinying Guo, Liang Pan, Zhongang Cai, Fangzhou Hong, Huirong Li, Lei Yang, and Ziwei Liu. Remodiffuse: Retrieval-augmented motion diffusion model. In *ICCV*, pages 364–373, 2023. 2, 3, 5, 7, 8

[57] Mingyuan Zhang, Huirong Li, Zhongang Cai, Jiawei Ren, Lei Yang, and Ziwei Liu. Finemogen: Fine-grained spatio-temporal motion generation and editing. *NeuIPS*, 36, 2024. 2

[58] Yaqi Zhang, Di Huang, Bin Liu, Shixiang Tang, Yan Lu, Lu Chen, Lei Bai, Qi Chu, Nenghai Yu, and Wanli Ouyang. Motiongpt: Finetuned llms are general-purpose motion generators. In *AAAI*, pages 7368–7376, 2024. 2, 3

[59] Yufei Zhang, Jeffrey O Kephart, and Qiang Ji. Incorporating physics principles for precise human motion prediction. In *WACV*, pages 6164–6174, 2024. 2