CoMo: Controllable Motion Generation through Language Guided Pose Code Editing

Yiming Huang[1], Weilin Wan[2], Yue Yang[1],
Chris Callison-Burch[1], Mark Yatskar[1], Lingjie Liu[1]

[1] University of Pennsylvania
[2] The University of Hong Kong
ymhuang9@seas.upenn.edu

Abstract. Text-to-motion models excel at efficient human motion generation, but existing approaches lack fine-grained controllability over the generation process. Consequently, modifying subtle postures within a motion or inserting new actions at specific moments remains a challenge, limiting the applicability of these methods in diverse scenarios. In light of these challenges, we introduce **CoMo**, a **Co**ntrollable **Mo**tion generation model, adept at accurately generating and editing motions by leveraging the knowledge priors of large language models (LLMs). Specifically, CoMo decomposes motions into discrete and semantically meaningful *pose codes*, with each code encapsulating the semantics of a body part, representing elementary information such as "left knee slightly bent". Given textual inputs, CoMo autoregressively generates sequences of pose codes, which are then decoded into 3D motions. Leveraging pose codes as interpretable representations, an LLM can directly intervene in motion editing by adjusting the pose codes according to editing instructions. Experiments demonstrate that CoMo achieves competitive performance in motion generation compared to state-of-the-art models while, in human studies, CoMo substantially surpasses previous work in motion editing abilities. Project page: https://yh2371.github.io/como/.

Keywords: Human Motion Synthesis · Human Motion Editing · Text-driven Motion Generation · Language Model Guided Generation

1 Introduction

The diversity of natural and unconstrained human motion holds rich intricacies crucial for fostering a deeper understanding of human behavior. The synthesis of such motion is challenging because models must both create plausible dynamics and reason over many possible solutions to a specification. Various conditional signals have been explored for guiding human motion synthesis, including audio signals [31,34,38,49] and simulated scenes [14,44,46]. Among these, natural language descriptions have emerged as a promising choice because they can communicate a broad set of needs naturally and could be used in design scenarios for animation and immersive technologies.

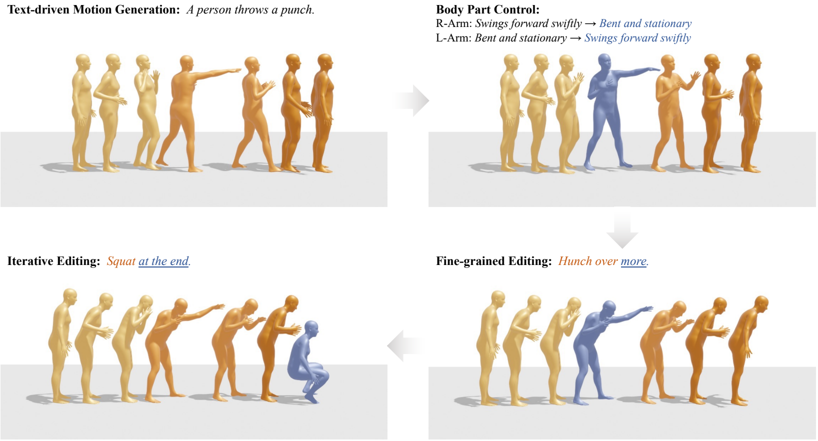
Text-driven Motion Generation: A person throws a punch.
Body Part Control:
R-Arm: Swings forward swiftly → Bent and stationary
L-Arm: Bent and stationary → Swings forward swiftly
Fine-grained Editing: Hunch over more.
Iterative Editing: Squat at the end.

Fig. 1: CoMo, a language-guided human motion synthesis model, enables controllable generation from text inputs. CoMo allows for the control of individual body part movements, facilitates fine-grained editing of each joint and frame, and supports iterative editing that preserves the essence of the original motions.

Several pioneering works in the text-to-motion task have demonstrated the expressiveness of textual descriptions in effectively guiding the creation of human motion sequences [10–12,15,32,33,40,45]. Approaches for this problem typically employ a method that maps a single text description to a latent code of attributes and then generates motion from codes. Such approaches may be poorly suited for fine-grained control of the generated motion. The generations imprecisely correspond to the text because codes contain a superimposed representation of body parts that a generator must then disentangle. Furthermore, capturing semantic relationships between text descriptions and low-level motion states may be too hard when the intermediate states must be simultaneously discovered. The need to capture a broad range of textual inputs amplifies the problem.

To address these challenges, we propose the **Co**ntrollable **Mo**tion Generation model (**CoMo**). As shown in Figure 1, CoMo can generate high-quality human motions from a broad range of text (e.g., *"A person throws a punch."*). Users can control the generation by altering the descriptions for each body part, e.g., switching the left and right arm descriptions changes the punching hand. More importantly, CoMo enables detailed editing across frames and joints (e.g., *"Hunch over more."*), adding actions (e.g., *"Squat at the end."*), and varying speed and even emotion (e.g., *"more dramatic"*), as depicted in Figure 6.

CoMo achieves these capabilities by representing motions as interpretable "pose codes", with each code defining the state of a specific body part at a given moment, e.g., *"right arm straight"*. As demonstrated in Figure 2, our method starts by factorizing a motion sequence into a series of temporal states, each composed of pose codes that describe the motion's kinematic characteristics. This structure allows for precise encoding of human motion sequences in time and fine-grained control of kinematic joint states. Since these encodings are explicitly interpretable, people can interact with and modify the sequences intuitively. We show that this process can also be made instructable with natural language by allowing large language models (LLM) to edit the codes as well.

Leveraging pose codes as interpretable motion representations, the three main components of CoMo work jointly to effectively generate and edit motion: (1) The *Motion Encoder-Decoder* applies heuristic rules to parse motions into sequences of semantically meaningful pose codes and trains a decoder to reconstruct these codes back into motions; (2) The *Motion Generator*, a transformer-based model, generates pose codes conditioned on text inputs and LLM-generated fine-grained descriptions; (3) The *Motion Editor* uses LLMs to modify and refine pose code sequences based on editing instructions. The resulting pose code sequences, whether generated or edited, are subsequently decoded into motion sequences using the previously trained decoder. CoMo allows for intuitive, language-controlled adjustments to the motion sequences, both temporally and kinematically, closely aligning the generated motions with users' creative intentions and the nuances expressed in their textual descriptions, making the process user-friendly and adaptable to diverse applications.

We evaluate the effectiveness of CoMo in text-driven motion generation against state-of-the-art methods on the HumanML3D [10] and KIT [26] datasets, ranking within the top 3 across most metrics. Beyond the competitive motion generation capabilities of CoMo, we also conducted a human evaluation with 54 participants for motion editing. On average, over 70% of annotators preferred the editing results produced by CoMo. CoMo's motion editing abilities allow for potential new applications, such as dialog-based motion generation.

In summary, our contributions in this paper are threefold:

1. We propose a semantic motion representation that factorizes motion sequences across space and time into explicit and interpretable pose codes.
2. We present a transformer-based model that autoregressively generates sets of low-level pose codes conditioned upon the high-level text description and fine-grained, body-part-specific descriptions generated by LLMs.
3. We demonstrate the capability of using the semantic low-level pose codes as an intuitive motion editing interface for LLMs.

2 Related Work

Text Conditioned Human Motion Generation. Conditional motion synthesis involves an interactive process for generating diverse, human-like motion from multi-modal user input, including text descriptions [3,10,15,18,25,29,32,33, 40,41,45,47,48], action categories [1,12,17] and physics-based signals [8,28,39]. To tackle the challenges in effectively mapping the intricacies of textual descriptions to meaningful movements, building a shared latent space has been a widely adopted solution [1, 25, 32, 32, 36]. Inspired by successful applications in image generation, the Vector Quantized Variational Autoencoder (VQ-VAE) model [23] has been widely applied to represent motion as discretized tokens, which can then be effectively combined with autoregressive transformer architectures for producing coherent motion sequences [11, 15, 40, 45, 48]. Conditional diffusion models are also increasingly powerful for generating high-fidelity results due to their capability of modeling complex distributions [3, 4, 18, 29, 33, 37, 39, 41, 42].

Fine-grained motion generation has gained significant interest due to its extensive practical applications. However, current methods have not fully integrated spatial and temporal details. TEACH [1] demonstrates improvements in motion smoothness by incorporating extensive temporal annotations but requires additional labeled data and overlooks lower-level spatial details. [2, 17, 30] leverage LLMs to generate detailed text descriptions for whole-body motions and individual body parts, aiming to map text to spatiotemporal details. However, without explicit supervision, these fine-grained details may not align with the respective motion. GraphMotion [16] proposes a hierarchical semantic graph that enforces coarse-to-fine topology within text-to-motion diffusion. However, constructing the hierarchical graph depends on the semantic parsing of text details and thus may struggle for ambiguous inputs. Also, the graph nodes do not model the lower-level spatial characteristics and relations. To this end, we propose body-part-specific semantic pose codes for achieving fine-grained representation of motion sequences. In addition, these pose codes are used as context to guide LLMs in generating motion-coherent descriptions for different body parts to enhance the connection between text and fine-grained motion details.

Motion Editing enables users to interactively refine generated motions to suit their expectations. PoseFix [6] automates 3D pose and text modifier generation for supervised editing, but frame-wise pose edits lack efficiency and temporal consistency. TLControl [35] allows motion editing using joint-level trajectories but relies on high-quality trajectory inputs and is not intuitive for interactive motion editing. [13] employs an autoencoder to optimize a motion manifold under positional, bone length, and trajectory constraints for smooth edits. Diffusion-based approaches [9, 18, 37] can achieve zero-shot spatiotemporal editing by infilling particular joints or frames, which may form unnatural discontinuities. Recently, FineMoGen [43] optimizes global attention for fine-grained editing but generates new sequences for each edit, limiting motion consistency. In our approach, we encode motions into semantic pose codes, which serve as context to prompt an LLM to edit the original motion directly, encouraging motion consistency.

3 Method

CoMo is a unified framework for fine-grained, text-driven human motion generation and editing. Figures 2 and 3 present an overview of CoMo, which consists of three key components: 1) **Motion Encoder-Decoder** (Sec. 3.1) decomposes motions into sequences of pose codes. These codes are then mapped back to motions through a decoder; 2) **Motion Generator** (Sec. 3.2) generates sequences of pose codes given high-level text descriptions and LLM-generated fine-grained

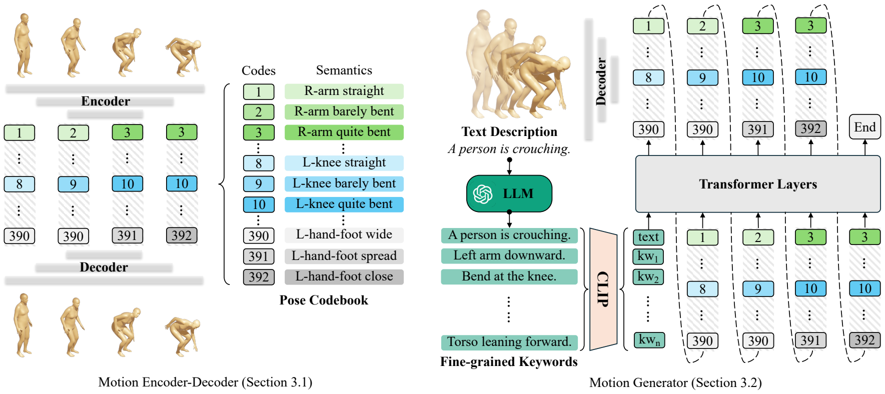

Fig. 2: Overview of CoMo for text-driven motion generation. Motion Encoder-Decoder (left) utilizes a predefined codebook to encode motions into pose codes and learns a decoder to reconstruct the motions. **Motion Generator** (right), a transformer-based model, predicts pose codes autoregressively, conditioned on the text descriptions and LLM-generated fine-grained keywords. The generated pose codes are then decoded back into motions using the previously trained decoder.

keywords; 3) **Motion Editor** (Sec. 3.3) employs an LLM to perform modifications on pose codes in a zero-shot manner.

3.1 Motion Encoder-Decoder

As shown in the left panel of Figure 2, given T frames of motion $X = \{x_i\}_{i=1}^{T}$, the motion encoder $\mathcal{E}$ projects them into a sequence of pose codes using a codebook, denoted as $Z = \mathcal{E}\left(X\right)$, where $Z = \{z_i\}_{i=1}^{L}$. Here, $L = T/l$ represents the length of the pose code sequence, and l is the downsampling rate over poses. The decoder $\mathcal{D}$ then decodes the pose codes back into motions, expressed as $X_{\text{rec}} = \mathcal{D}(\hat{Z})$, where $\hat{Z}$ stands for the latent features of pose codes. In the following, we will explain how to build the pose codebook, encode the motions using these codes, and learn the decoder to recover the motions.

Pose Codebook. Unlike [40], which uses an autoencoder to obtain implicit motion representations, CoMo predefines a semantic pose codebook to achieve interpretable motion factorization across time and space. Following PoseScript [5], we construct a pose codebook with N codes, $\mathcal{C} = \{c_n\}_{n=1}^{N}$, with $c_n \in \mathbb{R}^{d_c}$, where d_c is the dimension of the learnable codebook entry. Each code c_n is associated with a semantic meaning, representing a state of a body part or spatial relationships between body parts, e.g., "left knee slightly bent", "left hand and left foot close". These codes are further grouped into K pose categories, each encompassing different states of the same body parts, e.g., the pose category "left knee angle" includes codes like "left knee slightly bent", "left knee partially bent", etc.

Motion Encoder. Given the codebook $\mathcal{C}$ with N pose codes grouped into K categories, we encode motions into K-hot N dimensional vectors, denoted as $Z = \mathcal{E}\left(X\right) = \left\{z_i\right\}_{i=1}^{L}$, where $z_i \in \mathbb{R}^N$. Here, K-hot indicates that K elements of z_i are set to 1, while the others are 0. Specifically, at each time step i, we enforce mutual exclusivity between codes within the same category so that only one pose code per pose category is activated (set to 1 in the vector). To determine whether a pose code $c \in \mathcal{C}$ applies to a motion x, we use an off-the-shelf skeleton parser [5], denoted as $\mathcal{P}(c, x) \rightarrow \{0, 1\}$. The parser $\mathcal{P}$ analyzes the 3D joint positions of a skeleton in SMPL format [20], evaluating whether the pose meets specific heuristic threshold conditions for a given pose code. For instance, if the angle formed by the left shoulder, elbow, and wrist joints is less than 20 degrees, the code "left arm completely bent" is true. Therefore, instead of training an encoder model, we can explicitly factorize the motion sequence as:

$$Z = \mathcal{E}\left(X\right) = \left\{\left\{\mathcal{P}\left(c_n, x_{i\times l}\right)\right\}_{n=1}^{N}\right\}_{i=1}^{L} \tag{1}$$

where $x_{i\times l}$ is the motion frame extracted at the sampling rate l.

Motion Decoder. To develop a meaningful codebook, we train a 1D convolutional decoder [40], denoted as $\mathcal{D}$, over the latent features $\hat{Z}$ to reconstruct the original motion sequence, expressed as $X_{\text{rec}} = \mathcal{D}(\hat{Z})$. The latent features $\hat{Z} \in \mathbb{R}^{L\times d_c}$ are derived by summing the active codebook entries $c_n \in \mathcal{C}$, as indicated by the K-hot vector:

$$\hat{Z} = \left\{\sum_{n=1}^{N} \mathcal{P}\left(c_n, x_{i\times l}\right) \cdot c_n\right\}_{i=1}^{L} \tag{2}$$

We define $V(X) = \left\{x_{i+1} - x_i\right\}_{i=1}^{T-1}$ as the velocity of the T-frame motion sequence X. The reconstruction objective is formulated with smooth L1 loss $\mathcal{L}_1$:

$$\mathcal{L}_{\text{rec}} = \mathcal{L}_1\left(X, X_{\text{rec}}\right) + \lambda \cdot \mathcal{L}_1\left(V\left(X\right), V\left(X_{\text{rec}}\right)\right) \tag{3}$$

where the hyperparameter λ[3] balances the velocity and motion loss. The learned codebook and decoder are then frozen for motion generation and editing.

3.2 Motion Generator

As illustrated in the right panel of Figure 2, the Motion Generator, conditioned on text input, aims to generate a sequence of pose codes, which will be decoded into motions. Utilizing the learned pose codebook, we map a motion sequence $X = \left\{x_i\right\}_{i=1}^{T}$ to a sequence of K-hot, N-dimensional vectors $Z^{1:L} = \left\{\left\{z_i^n\right\}_{n=1}^{N}\right\}_{i=1}^{L}$, where z_i^n is an indicator function that is activated if the corresponding pose code c_n is true at the time index i. To denote the end

[3] Following [40], we set λ to 0.5.

of a motion sequence, we append an `<End>` code to each latent vector, activated when the motion stops, and the dimension of each latent vector z_i becomes $N+1$.

Treating the true label of each indicator z_i^n as an independent Bernoulli random variable, the text-to-motion generation task can be framed as an autoregressive multi-label prediction problem. Given the previous K-hot vectors $Z^{1:i-1}$ and a text condition t, our goal is to predict the Bernoulli distributions for the indicator terms z_i at the next time step i, represented as $P\left(z_i | t, Z^{1:i-1}\right)$.

To achieve this goal, we adopt a decoder-only transformer architecture with causal self-attention [40].[4] The likelihood of the full sequence is:

$$P(Z|t) = \prod_{i=1}^{L} \prod_{n=1}^{N+1} p\left(z_i^n \mid t, z_{1:i-1}^{1:N+1}\right) \tag{4}$$

We implement a binary cross-entropy loss and aim to maximize the average log-likelihood across all Bernoulli distributions:

$$\mathcal{L}_{\text{gen}} = -\frac{1}{L(N+1)} \sum_{i=1}^{L} \sum_{n=1}^{N+1} \mathbb{E}_{z_i^n \sim Ber\left(z_i^n\right)} \left[\log p\left(z_i^n \mid t, z_{1:i-1}^{1:N+1}\right)\right] \tag{5}$$

The predicted sequences of K-hot vectors are mapped back to respective pose codes, which can then be decoded to motion sequences through the decoder $\mathcal{D}$.

Fine-grained Keywords. To help the model capture more fine-grained details, we enhance the text description by using GPT-4 [24] to generate a keyword for each of the 10 body parts[5] and a keyword to describe the overall *mood* of the motion. For example, as shown in Figure 2, GPT-4 generates "bend at the knee", "torso leaning forward", etc., as the keywords for the motion *"crouching"*. We use CLIP [27] to extract text embeddings of the keywords and original description. The embedding of the original text, followed by the embeddings of 11 keywords, form the initial token sequence that conditions the subsequent motion generation. The effectiveness of using these keywords to improve generation performance is validated in Table 3.

3.3 Motion Editor

As illustrated in Figure 3, given an original motion, such as *crouching*, and an editing instruction like "pickup location should be slightly higher", the Motion Editor modifies the original motion to satisfy the requirements. Benefiting from our approach of encoding motions into explicit semantic pose codes, a Large Language Model (LLM) can interpret the motion and utilize its knowledge to reason about and execute editing instructions on an encoded motion sequence.

[4] Following the method for processing image patches in Vision Transformers [7], a linear layer projects the K-hot vectors before input into the transformer architecture.

[5] The 10 body parts are *head*, *torso*, *left arm*, *right arm*, *left hand*, *right hand*, *left leg*, *right leg*, *left feet*, *right feet.*

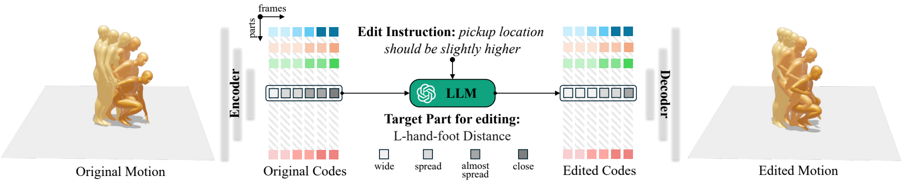

Fig. 3: Overview of CoMo for Fine-Grained Motion Editing: Given an original motion and an editing instruction, CoMo encodes the motion into pose codes, serving as the context to prompt an LLM. The LLM identifies the target codes for editing based on the instructions and updates the corresponding codes accordingly. These edited codes are then decoded back into motions to satisfy the user's requirements.

To simplify the task for the LLM, we design a sequential prompting[6] strategy that consists of three steps:

(1) **Identify the frames for editing.** The LLM or user determines the start and end indices of the motion segment where editing is needed. The identified subset of frames is retrieved and passed to the next step.

(2) **Identify the body parts for editing.** The LLM identifies which body parts require editing. The subset of corresponding pose categories is then retrieved. For example, in Figure 3, the category "L-hand-foot distance" is selected for modification to satisfy the editing instruction of "pickup location higher".

(3) **Edit the pose codes.** The LLM reviews each selected pose category to decide how the pose codes should be altered to align with the editing instructions. For instance, as depicted in Figure 3, the pose codes change from *close* to *almost spread* to mirror the instruction "pickup location should be slightly higher".

The edited segments of the pose codes are seamlessly integrated with their unedited counterparts, forming the complete edited sequence of pose codes, which is passed through the decoder $\mathcal{D}$ to reconstruct the final edited motion.

4 Evaluation

In this section, we evaluate CoMo from two perspectives: (1) experiments to demonstrate that CoMo achieves state-of-the-art performance on text-driven **motion generation** (Sec 4.1), and (2) human evaluation to showcase that CoMo is superior over existing methods in **motion editing** (Sec 4.2). We also conduct comprehensive ablation studies (Sec 4.3) to validate our model design.

4.1 Experiments on Motion Generation

Datasets. We evaluate our approach on two standard datasets for text-driven motion generation: HumanML3D [10] and KIT Motion Language (KIT-ML) [26].

[6] The complete prompts we use are available in the Appendix.

HumanML3D is a large-scale, diverse collection of human motion, including 14,616 distinct human motion capture sequences alongside 44,970 textual descriptions composed of 5,371 distinct words. The motion sequences are extracted from the HumanAct12 [12] and AMASS [21] datasets and preprocessed to 20 FPS. KIT-ML consists of 3,911 human motion sequences annotated with 6,278 distinct text annotations, forming a total vocabulary size of 1,623. The motion sequences are extracted from the KIT [22] and CMU [19] motion databases with a frame rate of 12.5 FPS. The motion sequences in KIT-ML and HumanML3D are all padded to 196 frames in length for training. Both datasets are split into 80% training, 5% validation, and 15% test sets as designated in [10].

Evaluation Metrics. We adhere to the protocol proposed in [10] and select the following five performance metrics for evaluation: *Frechet Inception Distance (FID)* measures the similarity between generated and real motion sequences. *R-Precision* and *Multimodal Distances (MM-DIST)* assess the relevance of generated motion sequences to their corresponding text descriptions. *Diversity* and *Multimodality (MModality)* reflect the variability of the generated motion sequences. We employ the pre-trained network from [10] to derive the motion and text feature embeddings necessary for calculating these metrics.[7]

Hyperparameters. In line with [10], the motion sequences from KIT-ML and HumanML3D are transformed into motion features with dimensions of 261 and 263, using 21 and 22 SMPL joints, respectively. These features represent global translations and rotations and local joint positions, velocities, and rotations. Employing PoseScript [5] as the skeleton parser, we define 70 pose categories encompassing 392 pose codes. The codebook size is 392×512.[8] The downsampling rate l is set to 4, and the maximum length of the code sequence produced by the Motion Generator is 50. All hyperparameters are tuned using the HumanML3D validation set (see ablation studies in Sec 4.3).

Training Details. We apply GPT-4 (`gpt-4-0613`) [24] to generate fine-grained descriptions and a frozen CLIP ViT-B/32 [27] to encode text. We employ the AdamW optimizer for training. The Motion Decoder is trained for 200K iterations with a 10^{-4} learning rate and batch size 256. The Motion Generator is trained for 300K iterations with a learning rate of 10^{-4} and batch size of 64. The Motion Decoder and Motion Generator are trained on a single NVIDIA RTX A6000 GPU for approximately 9 and 60 hours, respectively. The model with the best FID score on the validation split of each dataset is retained for testing.

CoMo achieves competitive results on text-driven motion generation. Table 1 and Table 2 present the results of text-driven motion generation on the HumanML3D and KIT-ML datasets, respectively. As shown in the first two rows of both tables, motion reconstruction with discrete pose codes achieves

[7] Details of metric calculations are provided in the Appendix.

[8] The definitions of all pose codes are listed in the Appendix.

Table 1: Comparison with the state-of-the-art methods on HumanML3D [10] test set. The best performance is **bold**, and the second best is underlined.

Method	**R-Precision ↑**			**FID ↓**	**MM-DIST ↓**	**Diversity ↑**	**MModality ↑**
	Top-1	**Top-2**	**Top-3**				
Real Motion	$0.511^{\pm.003}$	$0.703^{\pm.003}$	$0.797^{\pm.002}$	$0.002^{\pm.000}$	$2.974^{\pm.008}$	$9.503^{\pm.085}$	-
CoMo Recons.	$0.508^{\pm.002}$	$0.697^{\pm.002}$	$0.792^{\pm.002}$	$0.041^{\pm.000}$	$3.003^{\pm.006}$	$9.563^{\pm.100}$	-
Guo et al. [10]	$0.457^{\pm.002}$	$0.639^{\pm.003}$	$0.740^{\pm.003}$	$1.067^{\pm.002}$	$3.340^{\pm.008}$	$9.188^{\pm.002}$	$2.090^{\pm.083}$
TM2T [11]	$0.424^{\pm.002}$	$0.618^{\pm.003}$	$0.729^{\pm.002}$	$1.501^{\pm.017}$	$3.467^{\pm.011}$	$8.589^{\pm.086}$	$2.424^{\pm.093}$
TEMOS [25]	$0.424^{\pm.002}$	$0.612^{\pm.002}$	$0.722^{\pm.002}$	$3.734^{\pm.028}$	$3.703^{\pm.008}$	$8.973^{\pm.071}$	$0.368^{\pm.018}$
MDM [33]	$0.320^{\pm.005}$	$0.498^{\pm.004}$	$0.611^{\pm.007}$	$0.544^{\pm.044}$	$5.566^{\pm.027}$	$9.559^{\pm.086}$	$\mathbf{2.799}^{\pm.072}$
MotionDiffuse [41]	$0.491^{\pm.001}$	$0.681^{\pm.001}$	$0.782^{\pm.001}$	$0.630^{\pm.001}$	$3.113^{\pm.001}$	$9.410^{\pm.049}$	$1.533^{\pm.042}$
MLD [3]	$0.481^{\pm.003}$	$0.673^{\pm.003}$	$0.772^{\pm.002}$	$0.473^{\pm.013}$	$3.196^{\pm.010}$	$9.724^{\pm.082}$	$2.413^{\pm.079}$
T2M-GPT [40]	$0.491^{\pm.001}$	$0.680^{\pm.003}$	$0.775^{\pm.002}$	$\mathbf{0.116}^{\pm.004}$	$3.118^{\pm.011}$	$\underline{9.761}^{\pm.081}$	$1.831^{\pm.048}$
MotionGPT [15]	$0.492^{\pm.003}$	$0.681^{\pm.003}$	$0.778^{\pm.002}$	$0.232^{\pm.008}$	$3.096^{\pm.008}$	$9.528^{\pm.071}$	$2.008^{\pm.084}$
GraphMotion [16]	$\mathbf{0.504}^{\pm.003}$	$\mathbf{0.699}^{\pm.002}$	$\underline{0.785}^{\pm.002}$	$\mathbf{0.116}^{\pm.007}$	$3.070^{\pm.008}$	$9.692^{\pm.067}$	$\underline{2.766}^{\pm.096}$
FineMoGen [43]	$\mathbf{0.504}^{\pm.003}$	$0.690^{\pm.002}$	$0.784^{\pm.002}$	$\underline{0.151}^{\pm.008}$	$\mathbf{2.998}^{\pm.008}$	$9.263^{\pm.067}$	$2.696^{\pm.079}$
CoMo (Ours)	$\underline{0.502}^{\pm.002}$	$\underline{0.692}^{\pm.007}$	$\mathbf{0.790}^{\pm.002}$	$0.262^{\pm.004}$	$\underline{3.032}^{\pm.015}$	$\mathbf{9.936}^{\pm.066}$	$1.013^{\pm.046}$

Table 2: Comparison with the state-of-the-art methods on KIT [26] test set. The best performance is **bold**, and the second best is underlined, the third best is *italic*.

Method	**R-Precision ↑**			**FID ↓**	**MM-DIST ↓**	**Diversity ↑**	**MModality ↑**
	Top-1	**Top-2**	**Top-3**				
Real Motion	$0.424^{\pm.005}$	$0.649^{\pm.006}$	$0.779^{\pm.006}$	$0.031^{\pm.006}$	$2.788^{\pm.012}$	$11.08^{\pm.097}$	-
CoMo Recons.	$0.387^{\pm.005}$	$0.603^{\pm.005}$	$0.730^{\pm.005}$	$0.254^{\pm.007}$	$3.046^{\pm.011}$	$10.73^{\pm.128}$	-
Guo et al. [10]	$0.370^{\pm.005}$	$0.569^{\pm.007}$	$0.693^{\pm.007}$	$2.770^{\pm.109}$	$3.401^{\pm.008}$	$10.91^{\pm.119}$	$1.482^{\pm.065}$
TM2T [11]	$0.280^{\pm.005}$	$0.463^{\pm.006}$	$0.587^{\pm.005}$	$3.599^{\pm.153}$	$4.591^{\pm.026}$	$9.473^{\pm.117}$	$\underline{3.292}^{\pm.081}$
TEMOS [25]	$0.370^{\pm.005}$	$0.569^{\pm.007}$	$0.693^{\pm.007}$	$2.770^{\pm.109}$	$3.401^{\pm.008}$	$10.91^{\pm.119}$	$0.532^{\pm.034}$
MDM [33]	$0.164^{\pm.004}$	$0.291^{\pm.004}$	$0.396^{\pm.004}$	$0.497^{\pm.021}$	$9.191^{\pm.022}$	$10.85^{\pm.109}$	$1.907^{\pm.214}$
MotionDiffuse [41]	$0.417^{\pm.004}$	$0.621^{\pm.004}$	$0.739^{\pm.004}$	$1.954^{\pm.062}$	$\mathit{2.958}^{\pm.005}$	$\underline{11.10}^{\pm.143}$	$0.730^{\pm.013}$
MLD [3]	$0.390^{\pm.008}$	$0.609^{\pm.008}$	$0.734^{\pm.007}$	$0.404^{\pm.027}$	$3.204^{\pm.027}$	$10.80^{\pm.117}$	$2.192^{\pm.071}$
T2M-GPT [40]	$0.416^{\pm.006}$	$0.627^{\pm.006}$	$\mathit{0.745}^{\pm.006}$	$0.514^{\pm.029}$	$3.007^{\pm.023}$	$10.92^{\pm.108}$	$1.570^{\pm.039}$
MotionGPT [15]	$0.366^{\pm.005}$	$0.558^{\pm.004}$	$0.680^{\pm.005}$	$0.510^{\pm.016}$	$3.527^{\pm.021}$	$10.35^{\pm.084}$	$\mathit{2.328}^{\pm.117}$
GraphMotion [16]	$\mathit{0.429}^{\pm.007}$	$\underline{0.648}^{\pm.006}$	$\underline{0.769}^{\pm.008}$	$\underline{0.313}^{\pm.013}$	$3.076^{\pm.022}$	$\mathbf{11.12}^{\pm.135}$	$\mathbf{3.627}^{\pm.113}$
FineMoGen [43]	$\mathbf{0.432}^{\pm.006}$	$\mathbf{0.649}^{\pm.005}$	$\mathbf{0.772}^{\pm.008}$	$\mathbf{0.178}^{\pm.007}$	$\mathbf{2.869}^{\pm.014}$	$10.85^{\pm.115}$	$1.877^{\pm.093}$
CoMo (Ours)	$\underline{0.422}^{\pm.009}$	$\mathit{0.638}^{\pm.007}$	$\mathit{0.765}^{\pm.011}$	$\mathit{0.332}^{\pm.045}$	$\underline{2.873}^{\pm.021}$	$\mathit{10.95}^{\pm.196}$	$1.249^{\pm.008}$

high fidelity and closely matches ground-truth text-motion consistency, providing a strong foundation for motion generation. CoMo attains either the best or the second-best performance on HumanML3D across five metrics and ranks within the top three in six metrics on KIT. Compared to existing state-of-the-art methods, CoMo not only achieves competitive motion fidelity (FID) but also enhances motion diversity and consistency between generated motion and text descriptions (R-Precision, MM-DIST, Diversity). MModality measures the diversity of motions generated from the same text description. Due to the use of semantically meaningful pose codes, CoMo builds a stronger binding between text and the generated motion sequence, which potentially causes a slightly lower MModality score compared to prior methods as a trade-off for consistency.

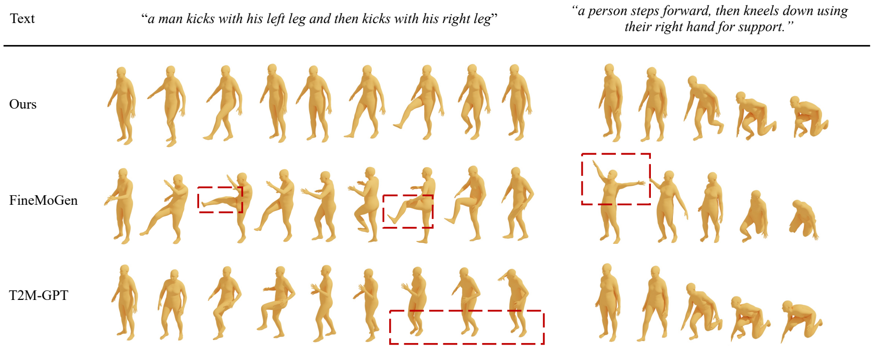
Text
"a man kicks with his left leg and then kicks with his right leg"
"a person steps forward, then kneels down using their right hand for support."
Ours
FineMoGen
T2M-GPT

Fig. 4: Qualitative examples of Motion Generation on the HumanML3D test set [10]. The motion sequences progress from left to right. The **red** boxes identify misalignments between the generated motion sequence and the text description. CoMo achieves competitive results in motion generation compared to T2M-GPT [40] and FineMoGen [43]. More visual results are available in the Appendix.

More importantly, our approach provides an intuitive interface for LLM-based zero-shot motion editing, which will be evaluated in Section 4.2 via a user study.

4.2 Human Evaluation on Motion Editing

We define the task of motion editing as the modification of a source motion according to a given textual editing instruction. Prior methods rely on modifying textual descriptions of the source motion based on edit instructions and subsequently generating new sequences from the updated descriptions to achieve motion editing [43]. In contrast, CoMo distinguishes itself by directly interpreting and manipulating the source motion sequence to facilitate editing.

We conducted a user study to assess the motion editing quality of CoMo in comparison with two state-of-the-art models for fine-grained text-to-motion generation: T2M-GPT [40] and FineMoGen [43]. We randomly selected 20 examples from the HumanML3D test set, annotating each with a motion edit instruction. As illustrated in Figure 6, these edit instructions encompass four types of motion editing: 1) body part modification (e.g., *"keep knees more deeply bent"*), 2) speed change (e.g., *"bend down slower"*), 3) style/emotion change (e.g., *"more dramatic"*) and 4) action addition/deletion (e.g., *"raise left hand at the end"*).

Baselines. To form strong baselines, we prompt GPT-4 to generate an updated description in the style of HumanML3D annotations using the original description and edit instructions as context[9]. Wcription to generate the edited motion for T2M-GPT and FineMoGen. For CoMo, we decompose the source motion as pose codes and prompt GPT-4 to edit the code sequence based on the original

[9] Prompts and updated descriptions are available in the Appendix.

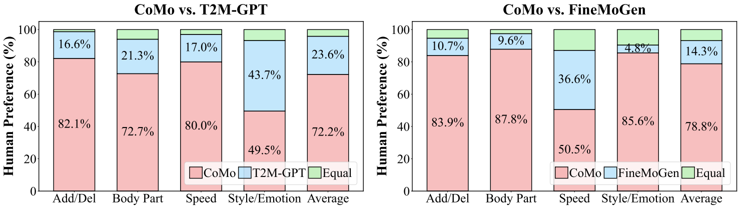

Fig. 5: Human preference on Motion Editing by comparing CoMo with T2M-GPT [40] and FineMoGen [43]. We report the scores on five editing types and average results.

Source Text	A person lowers their arms, and then moves them back up to shoulder height.	The person bent down and dodge something towards the left.	A person walks forward and then appears to bump into something, then continues walking forward.	A person beginning to run in a straight line.
Edit Text	Keep knees more deeply bent.	Bend down slower.	Make the bump more dramatic.	Raise left hand at the end.
Edit Type	Body part modification	Speed change	Style/emotion change	Add/delete actions

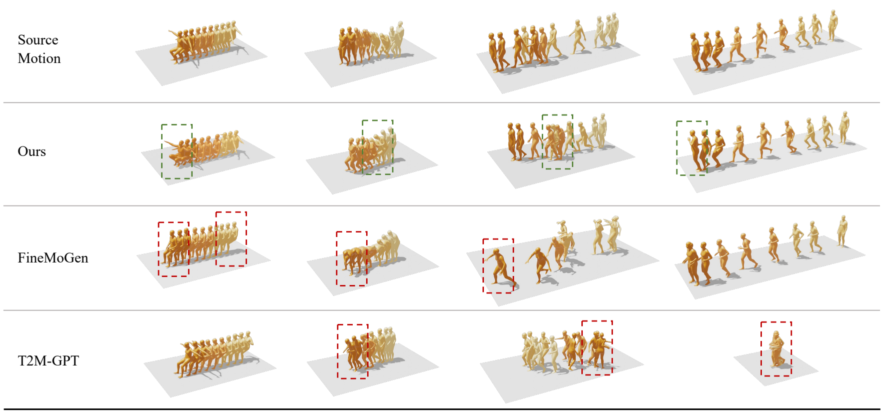

Fig. 6: Qualitative examples of Motion Editing on the HumanML3D test set [10]. The **green** words/boxes highlight successful edits. The **red** words/boxes identify misalignments between edited and source motions. Compared to other methods, CoMo achieves accurate edits while preserving key characteristics of the source motion.

description, edit instruction and pose code semantics without fine-tuning. The edited code sequence is decoded to reconstruct the edited motion.

User Study Setup. We have 54 graduate students evaluate 20 editing scenarios, with 6 scenarios each for body part modification and add/delete actions edits and 4 scenarios each for speed change and style/emotion change. The edit categories are evenly divided between the two baselines. For each edit scenario, participants are shown a pair of edited motion sequences, one from a baseline model and one from CoMo, along with the source motion sequence, source motion description (e.g., "a person is waving"), and an edit instruction (e.g., "raise the left hand higher"). The users are tasked with comparing the edited motions and choosing which best reflects the provided instruction and preserves the char-

Table 3: Ablation study of LLM-generated fine-grained keywords on HumanML3D and KIT. −Fine stands for the model without augmented keywords.

Method	HumanML3D				KIT Motion-Language			
	Top-1 ↑	**FID ↓**	**MM-DIST ↓**	**Diversity ↑**	**Top-1 ↑**	**FID ↓**	**MM-DIST ↓**	**Diversity ↑**
CoMo	**0.502**	**0.262**	**3.032**	**9.936**	**0.422**	**0.332**	**2.873**	10.95
−Fine	0.487	0.263	3.044	9.519	0.399	0.399	2.898	**11.26**

Table 4: Ablation study of different codebook sizes N. We report the reconstruction performance on the HumanML3D validation set. The number of angle/distance cutoffs stands for the granularity of parsing the joint position.

number of codes (N)	**angle cutoffs**	**distance cutoffs**	**Top-1 ↑**	**FID ↓**	**MM-DIST ↓**	**Diversity ↑**
205	3	2	0.500	0.059	3.030	9.592
261	6	4	0.505	0.049	3.023	9.620
392	18	10	**0.517**	**0.034**	**2.770**	**10.030**
661	18	20	0.507	0.047	3.009	9.553
733	36	20	0.504	0.037	3.019	9.520

Table 5: Ablation study of different sampling rates l. We report the reconstruction performance on the HumanML3D validation set.

Sampling Rate (l)	**Top-1 ↑**	**Top-2 ↑**	**Top-3 ↑**	**FID ↓**	**MM-DIST ↓**	**Diversity ↑**
2	**0.523**	**0.723**	**0.821**	**0.021**	**2.751**	10.048
4	0.517	0.718	0.815	0.034	2.770	10.030
8	0.513	0.712	0.809	0.065	2.824	**10.084**
16	0.486	0.682	0.782	0.176	3.021	10.039

acteristics of the source motion that are not affected by the edit.

Humans prefer CoMo for Motion Editing. We present the average percentage of users' preferences for CoMo versus T2M-GPT and FineMoGen in Figure 5. The results show a clear preference (over 70% on average) for motion editing with CoMo, especially in scenarios thatdify fine-grained motion details (e.g. body part modification and add/delete action). As depicted in Figure 6, while T2M-GPT and FineMoGen can produce motion relevant to the updated description, they often struggle to generate well-aligned details from scratch without any spatial-temporal context of the source motion. In contrast, CoMo leverages the semantics of pose codes to interpret the source motion effectively and achieve detailed motion editing. The observed decrease in preference for holistic motion edits involving changes in emotion or speed suggests that textual descriptions may provide more guidance in creating comprehensive edits that impact multiple aspects of a motion sequence. With its capability for high-level text-guided generation and iterative fine-grained pose code editing, CoMo emerges as a competitive approach for text-driven, controllable motion generation.

4.3 Ablation

This section ablates the fine-grained keywords, codebook size, and downsampling rate. Additional ablation studies are available in the Appendix.

Fine-grained Keywords. Table 3 shows an ablation study on the role of LLM-generated fine-grained keywords conducted on the HumanML3D dataset. The performance of the model decreases without those keywords. The specific details introduced in fine-grained keywords allow for a more consistent mapping between the original text description and generated motion.

Codebook Size. The number of pose codes indicates how fine-grained the heuristic thresholds of the skeleton parser are. More fine-grained thresholds capture more details at the expense of computational speed and learning complexity. We evaluate motion reconstruction quality for different codebook sizes by varying the angle and distance threshold settings, with finer thresholds corresponding to larger codebooks. Results in Table 4 indicate that our current codebook setting of 392 pose codes best balances complexity and reconstruction quality.

Downsampling Rate. We also investigate the effect of temporal downsampling on reconstruction quality. Similar to codebook size, smaller downsampling rates reserve details but result in longer sequences that increase task complexity and computation time. Comparing the results in Table 5, we select a downsampling rate of 4 to balance model performance and complexity.

5 Conclusion

In this paper, we propose CoMo, a controllable human motion synthesis system capable of generating and editing motion through language inputs. CoMo adopts a semantically meaningful pose code representation for encoding motion sequences across space and time. The interpretable pose codes within CoMo enable large language models to understand motion sequences and perform both kinematic and semantic motion editing effectively in a zero-shot manner. CoMo achieves state-of-the-art results in text-driven motion generation and human preferences confirm its superiority over alternative systems for motion editing.

Limitations. Although CoMo enhances controllability through keywords and a motion editing interface, the current formulation of keywords and pose codes focuses more on local kinematic descriptions. Further expanding the types of keywords and pose codes to include global descriptors of speed, style, trajectory, and motion repetition may allow for more flexibility in text-driven motion generation and editing. In addition. the semantics of pose codes enable zero-shot motion editing capabilities with LLMs but do not strictly constrain the motion edits to create physically feasible motion sequences. In the future, we aim to incorporate physical priors to guide the reasoning of LLMs during motion editing on pose codes to further enhance performance.

Acknowledgment

This research is supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via the HIATUS Program contract #2022-22072200005. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

1. Athanasiou, N., Petrovich, M., Black, M.J., Varol, G.: Teach: Temporal action compositions for 3d humans. In: International Conference on 3D Vision (3DV) (September 2022)
2. Athanasiou, N., Petrovich, M., Black, M.J., Varol, G.: SINC: Spatial composition of 3D human motions for simultaneous action generation. In: ICCV (2023)
3. Chen, X., Jiang, B., Liu, W., Huang, Z., Fu, B., Chen, T., Yu, G.: Executing your commands via motion diffusion in latent space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18000–18010 (2023)
4. Dabral, R., Mughal, M.H., Golyanik, V., Theobalt, C.: Mofusion: A framework for denoising-diffusion-based motion synthesis. In: Computer Vision and Pattern Recognition (CVPR) (2023)
5. Delmas, G., Weinzaepfel, P., Lucas, T., Moreno-Noguer, F., Rogez, G.: Posescript: 3d human poses from natural language (2022)
6. Delmas, G., Weinzaepfel, P., Moreno-Noguer, F., Rogez, G.: Posefix: Correcting 3d human poses with natural language (2023)
7. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2021), https://openreview.net/forum?id=YicbFdNTTy
8. Dou, Z., Chen, X., Fan, Q., Komura, T., Wang, W.: C· ase: Learning conditional adversarial skill embeddings for physics-based characters. In: SIGGRAPH Asia 2023 Conference Papers. pp. 1–11 (2023)
9. Goel, P., Wang, K.C., Liu, C.K., Fatahalian, K.: Iterative motion editing with natural language (2023)
10. Guo, C., Zou, S., Zuo, X., Wang, S., Ji, W., Li, X., Cheng, L.: Generating diverse and natural 3d human motions from text. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5152–5161 (June 2022)
11. Guo, C., Zuo, X., Wang, S., Cheng, L.: Tm2t: Stochastic and tokenized modeling for the reciprocal generation of 3d human motions and texts. In: European Conference on Computer Vision. pp. 580–597. Springer (2022)
12. Guo, C., Zuo, X., Wang, S., Zou, S., Sun, Q., Deng, A., Gong, M., Cheng, L.: Action2motion: Conditioned generation of 3d human motions. In: Proceedings of the 28th ACM International Conference on Multimedia. pp. 2021–2029 (2020)

13. Holden, D., Saito, J., Komura, T.: A Deep Learning Framework for Character Motion Synthesis and Editing. Association for Computing Machinery, New York, NY, USA, 1 edn. (2023), https://doi.org/10.1145/3596711.3596789
14. Huang, S., Wang, Z., Li, P., Jia, B., Liu, T., Zhu, Y., Liang, W., Zhu, S.C.: Diffusion-based generation, optimization, and planning in 3d scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
15. Jiang, B., Chen, X., Liu, W., Yu, J., Yu, G., Chen, T.: Motiongpt: Human motion as a foreign language. Advances in Neural Information Processing Systems **36** (2024)
16. Jin, P., Wu, Y., Fan, Y., Sun, Z., Wei, Y., Yuan, L.: Act as you wish: Fine-grained control of motion diffusion model with hierarchical semantic graphs. In: NeurIPS (2023)
17. Kalakonda, S.S., Maheshwari, S., Sarvadevabhatla, R.K.: Action-gpt: Leveraging large-scale language models for improved and generalized action generation (2023)
18. Kim, J., Kim, J., Choi, S.: Flame: Free-form language-based motion synthesis editing. In: Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence. AAAI'23/IAAI'23/EAAI'23, AAAI Press (2023). https://doi.org/10.1609/aaai.v37i7.25996, https://doi.org/10.1609/aaai.v37i7.25996
19. Lab, C.M.U.G.: Cmu graphics lab motion capture database (2004)
20. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: SMPL: A skinned multi-person linear model. ACM Trans. Graphics (Proc. SIGGRAPH Asia) **34**(6), 248:1–248:16 (Oct 2015)
21. Mahmood, N., Ghorbani, N., Troje, N.F., Pons-Moll, G., Black, M.J.: AMASS: Archive of motion capture as surface shapes. In: International Conference on Computer Vision. pp. 5442–5451 (Oct 2019)
22. Mandery, C., Terlemez, O., Do, M., Vahrenkamp, N., Asfour, T.: Unifying representations and large-scale whole-body motion databases for studying human motion. IEEE Transactions on Robotics **32**(4), 796–809 (2016)
23. van den Oord, A., Vinyals, O., Kavukcuoglu, K.: Neural discrete representation learning. CoRR **abs/1711.00937** (2017), http://arxiv.org/abs/1711.00937
24. OpenAI, R.: Gpt-4 technical report. arXiv pp. 2303–08774 (2023)
25. Petrovich, M., Black, M.J., Varol, G.: Temos: Generating diverse human motions from textual descriptions. In: Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII. p. 480–497. Springer-Verlag, Berlin, Heidelberg (2022)
26. Plappert, M., Mandery, C., Asfour, T.: The kit motion-language dataset. Big data **4**(4), 236–252 (2016)
27. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision (2021)
28. Ren, J., Yu, C., Chen, S., Ma, X., Pan, L., Liu, Z.: Diffmimic: Efficient motion mimicking with differentiable physics. ICLR (2022)
29. Shafir, Y., Tevet, G., Kapon, R., Bermano, A.H.: Human motion diffusion as a generative prior (2023)
30. Shi, X., Luo, C., Peng, J., Zhang, H., Sun, Y.: Generating fine-grained human motions using chatgpt-refined descriptions (2023)

31. Siyao, L., Yu, W., Gu, T., Lin, C., Wang, Q., Qian, C., Loy, C.C., Liu, Z.: Bailando: 3d dance generation via actor-critic gpt with choreographic memory. In: CVPR (2022)
32. Tevet, G., Gordon, B., Hertz, A., Bermano, A.H., Cohen-Or, D.: Motionclip: Exposing human motion generation to clip space. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII. pp. 358–374. Springer (2022)
33. Tevet, G., Raab, S., Gordon, B., Shafir, Y., Cohen-Or, D., Bermano, A.H.: Human motion diffusion model. arXiv preprint arXiv:2209.14916 (2022)
34. Tseng, J., Castellon, R., Liu, K.: Edge: Editable dance generation from music. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 448–458 (2023)
35. Wan, W., Dou, Z., Komura, T., Wang, W., Jayaraman, D., Liu, L.: Tlcontrol: Trajectory and language control for human motion synthesis (2023)
36. Wan, W., Huang, Y., Wu, S., Komura, T., Wang, W., Jayaraman, D., Liu, L.: Diffusionphase: Motion diffusion in frequency domain. arXiv preprint arXiv:2312.04036 (2023)
37. Xie, Y., Jampani, V., Zhong, L., Sun, D., Jiang, H.: Omnicontrol: Control any joint at any time for human motion generation (2023)
38. Yi, H., Liang, H., Liu, Y., Cao, Q., Wen, Y., Bolkart, T., Tao, D., Black, M.J.: Generating holistic 3d human motion from speech. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 469–480 (June 2023)
39. Yuan, Y., Song, J., Iqbal, U., Vahdat, A., Kautz, J.: Physdiff: Physics-guided human motion diffusion model. arXiv preprint arXiv:2212.02500 (2022)
40. Zhang, J., Zhang, Y., Cun, X., Huang, S., Zhang, Y., Zhao, H., Lu, H., Shen, X.: T2m-gpt: Generating human motion from textual descriptions with discrete representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
41. Zhang, M., Cai, Z., Pan, L., Hong, F., Guo, X., Yang, L., Liu, Z.: Motiondiffuse: Text-driven human motion generation with diffusion model. arXiv preprint arXiv:2208.15001 (2022)
42. Zhang, M., Guo, X., Pan, L., Cai, Z., Hong, F., Li, H., Yang, L., Liu, Z.: Remodiffuse: Retrieval-augmented motion diffusion model. arXiv preprint arXiv:2304.01116 (2023)
43. Zhang, M., Li, H., Cai, Z., Ren, J., Yang, L., Liu, Z.: Finemogen: Fine-grained spatio-temporal motion generation and editing. NeurIPS (2023)
44. Zhang, X., Bhatnagar, B.L., Starke, S., Guzov, V., Pons-Moll, G.: Couch: Towards controllable human-chair interactions. In: Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part V. p. 518–535. Springer-Verlag, Berlin, Heidelberg (2022). https://doi.org/10.1007/978-3-031-20065-6_30, https://doi.org/10.1007/978-3-031-20065-6_30
45. Zhang, Y., Huang, D., Liu, B., Tang, S., Lu, Y., Chen, L., Bai, L., Chu, Q., Yu, N., Ouyang, W.: Motiongpt: Finetuned llms are general-purpose motion generators (2023)
46. Zhao, K., Zhang, Y., Wang, S., Beeler, T., , Tang, S.: Synthesizing diverse human motions in 3d indoor scenes. In: International conference on computer vision (ICCV) (2023)
47. Zhou, W., Dou, Z., Cao, Z., Liao, Z., Wang, J., Wang, W., Liu, Y., Komura, T., Wang, W., Liu, L.: Emdm: Efficient motion diffusion model for fast, high-quality motion generation. arXiv preprint arXiv:2312.02256 (2023)

48. Zhou, Z., Wang, B.: Ude: A unified driving engine for human motion generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5632–5641 (June 2023)
49. Zhu, L., Liu, X., Liu, X., Qian, R., Liu, Z., Yu, L.: Taming diffusion models for audio-driven co-speech gesture generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10544–10553 (2023)