M^{3}GPT: An Advanced Multimodal, Multitask Framework for Motion Comprehension and Generation

Mingshuang Luo[1,2,3], **Ruibing Hou**[1*], **Zhuo Li**[4], **Hong Chang**[1,3],
Zimo Liu[2], **Yaowei Wang**[2,5], **Shiguang Shan**[1,3]
[1]Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS),
Institute of Computing Technology, CAS, China
[2]Peng Cheng Laboratory, China, [3]University of Chinese Academy of Sciences, China
[4]WeChat, Tencent Inc, [5]Harbin Institute of Technology, Shenzhen
`mingshuang.luo@vipl.ict.ac.cn,{houruibing,changhong,sgshan}@ict.ac.cn`
`albertzli@tencent.com,liuzm@pcl.ac.cn,wangyaowei@hit.edu.cn`

Abstract

This paper presents M^{3}GPT, an advanced **M**ultimodal, **M**ultitask framework for **M**otion comprehension and generation. M^{3}GPT operates on three fundamental principles. The first focuses on creating a unified representation space for various motion-relevant modalities. We employ discrete vector quantization for multimodal conditional signals, such as text, music and motion/dance, enabling seamless integration into a large language model (LLM) with a single vocabulary. The second involves modeling motion generation directly in the raw motion space. This strategy circumvents the information loss associated with a discrete tokenizer, resulting in more detailed and comprehensive motion generation. Third, M^{3}GPT learns to model the connections and synergies among various motion-relevant tasks. Text, the most familiar and well-understood modality for LLMs, is utilized as a bridge to establish connections between different motion tasks, facilitating mutual reinforcement. To our knowledge, M^{3}GPT is the first model capable of comprehending and generating motions based on multiple signals. Extensive experiments highlight M^{3}GPT's superior performance across various motion-relevant tasks and its powerful zero-shot generalization capabilities for extremely challenging tasks. Project page: `https://luomingshuang.github.io/M3GPT/`.

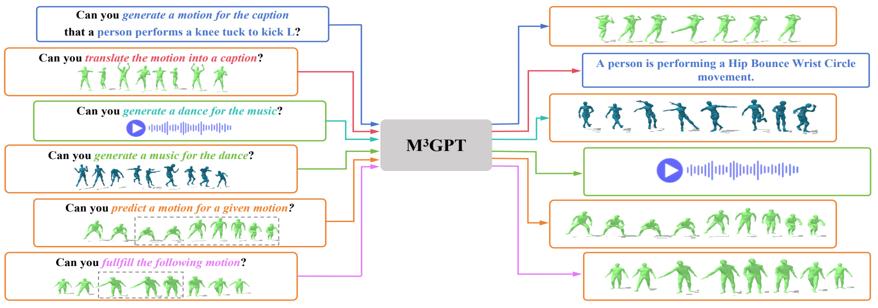

Can you *generate a motion for the caption* that a person performs a knee tuck to kick L?
Can you *translate the motion into a caption*?
Can you *generate a dance for the music*?
Can you *generate a music for the dance*?
Can you *predict a motion for a given motion*?
Can you *fullfill the following motion*?
M3GPT
A person is performing a Hip Bounce Wrist Circle movement.

Figure 1: M^{3}GPT can handle core motion comprehension and generation tasks, including text-to-motion, motion-to-text, music-to-dance, dance-to-music, motion prediction, and motion in-between. The motion sequences within the dashed-line areas are masked in the input.

[*]Corresponding author

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

1 Introduction

Motion comprehension and generation in multimodality are crucial for diverse applications, including AR/VR creation, video games, and virtual reality. Numerous studies [15, 51, 64, 52] focus on motion comprehension, including captioning 3D human motions and generating music from 3D human dances[2]. Recent advancements in AI [14, 44, 42, 48] have paved the way for motion generation, allowing for various control signals including textual descriptions, music pieces, and human poses. A significant shortcoming of most existing works is their focus on single-modality control signals, overlooking the potential for multimodal information integration. More importantly, the comprehension and generation of motions are predominantly studied in isolation. In reality, human motion cognition and communication indispensably require seamless transitions between any motion-relevant modalities. Therefore, it is vital to develop a unified framework for motion comprehension and generation that can efficiently utilize multiple signals simultaneously.

Recent works [12, 62, 63, 58] have shown success in developing a unified multitask motion framework which integrates text-driven and audio-driven motion generation through a single architecture. Employing a large language model (LLM), [60] adeptly handles multimodal control signals, such as text and single-frame pose, to generate consecutive motions. Despite their promising performance in motion generation, these approaches often fall short in comprehending motion. MotionGPT [21], a recent innovation, constructs a unified motion-language model to generate plausible human motions and natural language descriptions through prompt instructions. However, MotionGPT focuses solely on a single non-motion modality, *i.e.*, text. While aligning motion with one additional modality is relatively straightforward, integrating three or more modalities within a single framework and achieving bidirectional alignment among them to cover a broad range of modalities for motion comprehension and generation presents a formidable challenge.

Two main challenges need to be solved for building a unified multimodal framework for motion comprehension and generation. ***The first is how to create a unified representation space across different motion-relevant modalities.*** MotionGPT [21] and SpeechGPT [54] separately treat motion and speech as specific language for seamlessly integrating with text. Inspired by these efforts [21, 54], we view both motion and music as distinct forms of language, facilitating better associations with text via LLMs. Specifically, akin to language, we compress raw motion and music into a sequence of discrete semantic tokens. By encoding motion, music, and language within a single vocabulary, we can build a unified representation space across these different modalities. ***The second is how to model the connections and synergies among various motion tasks.*** Different motion-relevant tasks are interconnected and can mutually enhance each other. Since text is the most familiar and well-understood modality for LLMs, we propose employing text as a bridge to establish connections between different motion tasks. Specifically, to better learn the complex music-to-dance task where both input and output modalities are unfamiliar to LLMs, we introduce two auxiliary tasks: music-to-text and text-to-dance, aimed at aligning music and dance modalities with the structured text embedding space. This strategy enables us to establish connections and synergies between music-to-dance and text-to-motion tasks, facilitating the alignment and collaboration of text, music, and motion/dance modalities across different tasks.

In this work, we propose a uniform **M**ultimodal, **M**ultitask framework for **M**otion comprehension and generation, namely M^{3}GPT, that leverages the strong language generation capability of LLMs for unifying various motion-relevant tasks, as depicted in Fig. 1. M^{3}GPT comprises three tires. **Firstly**, M^{3}GPT is equipped with multimodal tokenizers capable of compressing raw multimodal data, including motion, music, and text, into a sequence of discrete semantic tokens. These discrete representations allow the core LLM to unify motion comprehension and generation in an autoregressive manner, operating at the discrete semantic representation space. **Secondly**, different from [21, 60] that solely optimize LLM in discrete semantic space, we jointly train LLM and motion de-tokenizer, optimizing LLM in both discrete semantic space and raw continuous motion space. This operation enables the motion-space error signals from de-tokenizer to backpropagate to LLM, enhancing LLM's ability to generate the details of motion. **Thirdly**, we construct paired text descriptions for music, and design two auxiliary music-to-text and text-to-dance tasks, which aid in aligning music and dance with the text embedding space. Also, we build up a shared tokenizer for motion and dance data

[2]In this paper, the term "motion" generally includes "dance." We distinguish them when referring to specific tasks or scenes, such as text-to-motion, and music-to-dance.

Methods	T2M	M2T	A2D	D2A	M2M	Random M	Random T	Random A
TM2D[12]	✓	✗	✓	✗	✗	✓	✗	✗
UDE[62]	✓	✗	✓	✗	✗	✓	✗	✗
MotionGPT[60]	✓	✗	✗	✗	✓	✓	✗	✗
MotionGPT[21]	✓	✓	✗	✗	✓	✓	✓	✗
M^{3}GPT (Ours)	✓	✓	✓	✓	✓	✓	✓	✓

Table 1: Comparison of recent multimodal, multitask methods across various motion comprehension and generation tasks. T2M: text-to-motion; M2T: motion-to-text; A2D: music-to-dance; D2A: dance-to-music; M2M: motion-to-motion that includes motion prediction and motion in-between. Random M, Random T, and Random A represent the unconstrained generation of motion, text, and music[3], respectively.

to project them into a shared semantic space. These auxiliary tasks and shared tokenizer establish connections between music-to-dance and text-to-motion, enabling mutual reinforcement.

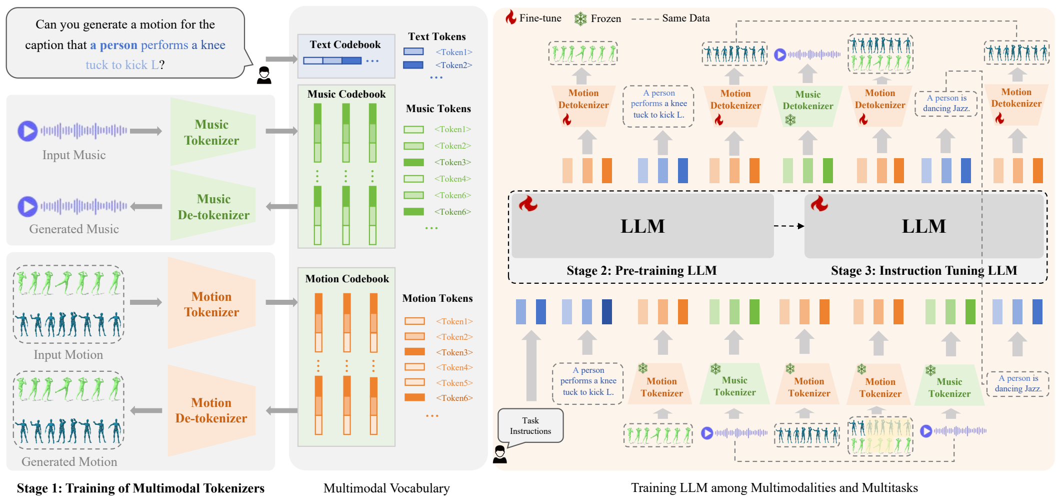
We employ a multimodal pre-training + instruction-tuning pipeline to train M^{3}GPT, enhancing inter-modal alignment and effectively aligning them with human intent. To our knowledge, M^{3}GPT is the first approach to integrate six core tasks of motion comprehension and generation—text-to-motion, motion-to-text, music-to-dance, dance-to-music, motion prediction, and motion in-between—into a uniform framework. Extensive experiments demonstrate that M^{3}GPT achieves competitive performance across multiple motion-relevant tasks. Additionally, through qualitative results, we demonstrate that M^{3}GPT possesses powerful zero-shot generalization capabilities, *e.g.*, long-term dance generation and music-text conditioned dance generation.

2 Related Work

Motion comprehension and Generation. Many existing works focus on studying human appearance, pose, detection, attribute, part parsing and so on [61, 19, 45, 40, 23, 17]. This work focuses on studying human motion, including motion comprehension and motion generation. Motion comprehension involves two core tasks: motion-to-text and dance-to-music. *Motion-to-text* aims to describe human motion with natural language [37]. For example, recurrent networks have been used in [37] to accomplish this task. *Dance-to-music* involves creating a piece of music from a given dance [20, 27, 64]. For example, Zhun *et al.* [64] utilizes a generative adversarial network to generate music from dance videos. On the other hand, motion generation involves generating diverse human motions using multimodal inputs, such as text [44, 56, 15, 5, 57], music [27, 18, 52, 42] and incomplete motion [31, 1, 3]. *Text-to-motion* is one of the most important motion generation tasks. Recent works typically map text to motion using different architectures: diffusion model [57] and autoregressive transformer model [15]. *Music-to-dance* focuses on generating dance movements from music. For example, [42] predicts discrete token sequences conditioned on music, which are then used to regenerate the dance sequence. *Motion Completion* generates motion conditioning on partial motions, such as motion prediction [31, 1] and motion-in-between [34, 43]. Although these methods have shown promising results in various human motion tasks, most are limited to handling a single task. Until recently, some works [12, 21, 60, 62] attempt to integrate two or more tasks into a unified model, as shown in Tab. 1. However, these works either lack the ability of motion comprehension [62, 60] or fail to handle music modality [12, 21]. In this work, we propose a unified motion comprehension and generation framework that can handle multiple control signals simultaneously.

Language Models and Multimodal. Large language models (LLMs) enabled by extensive datasets and model size, such as T5 [39], Flan-T5 [7], LLaMA [46], LLaMA-2 [47] and Vicuna [6], have demonstrated impressive comprehension and generation capabilities. Researchers have leveraged the capabilities of LLMs to handle multimodal tasks, expanding them to multimodal large language models (MLLMs). For example, AnyGPT [53] employs LLaMA-2 [47] to construct an any-to-any multimodal language model. NExT-GPT [50] employs Vicuna [6] with multimodal adaptors and diffusion decoders to perform tasks across arbitrary combinations of text, images, videos, and audio. Recently, the works [21, 60] attempt to use LLMs for motion-related tasks. [60] uses LLaMA [46] to build a general-purpose motion generator, which, however, lacks the ability to comprehend motion. [21] leverages T5 to construct a unified motion-language model, but cannot deal with music modality.

[3]Note: in this paper, the term "audio" specifically refers to "music". This designation is adopted to avoid confusion between the initial letter "M" shared by both "music" and "motion," which could lead to ambiguity when these modalities are represented by their initials.



Can you generate a motion for the caption that a person performs a knee tuck to kick L?
Input Music
Music Tokenizer
Generated Music
Music De-tokenizer
Input Motion
Motion Tokenizer
Generated Motion
Motion De-tokenizer
Text Codebook
Text Tokens
Music Codebook
Music Tokens
Motion Codebook
Motion Tokens
Fine-tune
Frozen
Same Data
LLM
Stage 2: Pre-training LLM
Stage 3: Instruction Tuning LLM
A person performs a knee tuck to kick L.
A person is dancing Jazz.
Task Instructions
Stage 1: Training of Multimodal Tokenizers
Multimodal Vocabulary
Training LLM among Multimodalities and Multitasks

Figure 2: An overview of the M^{3}GPT framework. M^{3}GPT consists of multimodal tokenizers and a motion-aware language model. The training process of M^{3}GPT consists of three stages: multimodal tokenizers training, modality-alignment pre-training, and instruction tuning.

3 Method

To enhance the comprehension and generation of motion-relevant modalities, we propose a unified multimodal framework named M^{3}GPT. As illustrated in Fig. 2, M^{3}GPT consists of multimodal tokenizers responsible for compressing raw motion and music data into discrete tokens (Sec. 3.1), and a motion-aware language model that learns to understand and generate motion tokens from LLMs by corresponding text and music (Sec. 3.2). To address motion-relevant tasks, we employ a three-stage training scheme encompassing multimodal tokenizers training, modality-alignment pre-training, and instruction tuning (Sec. 3.3). During the inference process, multimodal tokens are decoded back into their original representations by associated *de-tokenizers* (decoders of multimodal tokenizers), enabling various motion-relevant tasks to be executed via instructions (Sec. 3.4).

3.1 Multimodal tokenizers

As shown in Fig. 2, Multimodal tokenizers aim to discretize continuous human motion and music into language-like tokens, allowing the three modalities to be unified within a single language model.

3D Human Motion Tokenizer. To represent motion in discrete semantic tokens, we build a 3D human motion tokenizer based on Vector Quantized Variational Autoencoders (VQ-VAE) following [12, 62, 21, 60]. The motion tokenizer consists of a motion encoder $\mathcal{E}_m$ and a motion decoder $\mathcal{D}_m$, along with a codebook $\mathcal{B}_m = \left\{b^1, b^2, \ldots, b^{N_m}\right\}$ containing N_m discrete semantic vectors. Notably, to facilitate mutual enhancement between motion and dance data, we employ a shared tokenizer for both motions and dances, projecting them into a consistent and shared semantic space. Formally, given a 3D motion sequence $\boldsymbol{m} \in \mathbb{R}^{T_m \times d_m}$, where T_m is the time length and d_m is the dimensionality of each frame's pose, the motion encoder $\mathcal{E}_m$ that consists of several 1-D convolutional layers projects $\boldsymbol{m}$ to a latent embeddings $\boldsymbol{z} \in \mathbb{R}^{L_m \times d}$. Here, L_m is the time interval after downsampling and d is the latent dimension. Next, we transform $\boldsymbol{z}$ into a collection of codebook entries through discrete quantization. Specifically, the quantization process replaces each item of $\boldsymbol{z}$ with its nearest embedding in the codebook $\mathcal{B}_m$, obtaining the quantized latent vectors $\boldsymbol{e} \in \mathbb{R}^{L_m \times d}$ as follows:

$$\boldsymbol{e} = \underset{b^k \in \mathcal{B}_m}{\arg\min} \left\|\boldsymbol{z} - b^k\right\|_2 . \tag{1}$$

The motion decoder $\mathcal{D}_m$, which consists of several 1-D deconvolutional layers, projects the quantized embeddings back to raw motion space as $\hat{\boldsymbol{m}} = \mathcal{D}_m\left(\boldsymbol{e}\right)$. Following [21, 60], the motion tokenizer can be trained by the reconstruction loss, embedding loss and commitment loss as follows:

$$\mathcal{L}_{vq} = \|\hat{\boldsymbol{m}} - \boldsymbol{m}\|_1 + \|\text{sg}\left[\boldsymbol{z}\right] - \boldsymbol{e}\|_2^2 + \beta\|\boldsymbol{z} - \text{sg}\left[\boldsymbol{e}\right]\|_2^2 . \tag{2}$$

where sg $[\cdot]$ is the stop gradient, and β is the factor that adjusts the weight of the commitment loss.

After training the motion tokenizer, a motion sequence $\boldsymbol{m}$ can be represented as a sequence of discrete codebook-indices of quantized embedding vector, namely *motion tokens* $\boldsymbol{q_m} \in \mathbb{R}^{L_m}$, as follows:

$$\boldsymbol{q_m} = \underset{k\in\{1,\dots,N_m\}}{\arg\min} \left\| \mathcal{E}_m\left(\boldsymbol{m}\right) - b^k \right\|_2 . \tag{3}$$

Music Tokenizer. For the music data, we adopt the VQ-VAE in Jukebox [9] as the music tokenizer, which consists of a music encoder $\mathcal{E}_a$, a music decoder $\mathcal{D}_a$ and a music codebook $\mathcal{B}_a$. Notably, the limited number of music samples in dance datasets makes it inadequate for training an effective music tokenizer. To leverage the strong representation ability of the tokenizer trained on the large-scale musical dataset, we use the pre-trained VQ-VAE from Jukebox [9], which has been trained on a dataset of 1.2 million songs. Specifically, we first segment each input music sample into 5-second music segments. Then, for each 5 seconds segment $\boldsymbol{a} \in \mathbb{R}^{T_a \times d_a}$, we use the pre-trained music tokenizer $\{\mathcal{E}_a, \mathcal{B}_a\}$ to encode $\boldsymbol{a}$ into a sequence of discrete codebook-indices $\boldsymbol{q_a} \in \mathbb{R}^{L_a}$ (namely *music tokens*) following Eq. 3.

3.2 Language Model Backbone

Expanding Vocabulary. To incorporate multimodal discrete representations into a pre-trained LLM, we expand the original text vocabulary V_t in LLM with motion vocabulary $\mathcal{B}_m$ and music vocabulary $\mathcal{B}_a$, forming a new unified vocabulary $V = \{V_t, \mathcal{B}_m, \mathcal{B}_a\}$. To accommodate the expanded vocabulary, we extend the corresponding embedding and prediction layer of LLM, where the newly incorporated parameters are initialized randomly.

Unified Multimodal Language Model. Equipped with multimodal tokenizers, we can compress multimodal data into discrete token sequences. To be specific, employing the trained motion tokenizer and music tokenizer, the input motion $\boldsymbol{m} \in \mathbb{R}^{T_m \times d_m}$ and music $\boldsymbol{a} \in \mathbb{R}^{T_a \times d_a}$ can be mapped into a sequence of discrete motion tokens $\boldsymbol{q_m} \in \mathbb{R}^{L_m}$ and music tokens $\boldsymbol{q_a} \in \mathbb{R}^{L_a}$. Then equipped with a unified vocabulary V, we can formulate various motion-relevant tasks in a general format, where both input and output tokens come from the same vocabulary. These tokens can represent natural language, human motion, music, or any combination, depending on the specific task to be solved. This naturally enables the core LLM to unify motion comprehension and generation tasks in an autoregressive manner.

Following [21], we employ T5 [39] as the language model backbone, which is pre-trained on 750 GB of text tokens. By leveraging this pre-trained large language model, we can harness its powerful modeling capabilities and generalizability to develop a more user-friendly, motion-related human-computer interaction model.

3.3 Training Strategy

The training process is divided into three stages. The first stage is Multimodal Tokenizers Training, which focuses on learning the motion/music tokenizer to represent motion/music as discrete tokens. The second stage is Modality-Alignment Pre-training, which aims to align motion, music, and text modalities, and facilitate collaboration across different motion-relevant tasks. The third stage is Instruction Fine-Tuning, aimed at enhancing the model's instruction-following capability.

Stage1: Multimodal Tokenizers Training. We first train a motion tokenizer using the objective defined in Eq. 2. As for the music tokenizer, due to the limited music samples in existing dance datasets, we directly use the pre-trained VQ-VAE model from Jukebox [9]. This process allows any motion sequence and music to be represented as a sequence of tokens, enabling seamless integration with text within LLM. To ensure the stability of LLM training, the encoder of motion tokenizer and whole music tokenizer remain unchanged. Notably, we continue to optimize the decoder of motion tokenizer in subsequent training stages to further enhance the quality of generated motions.

Stage2: Modality-Alignment Pre-training. To enable LLM to handle discrete modalities, we utilize paired motion corpus to train LLM in a next-token prediction task. This process aims to align the text, music, and motion modalities for unified reasoning in LLM.

- **Joint optimization of LLM and motion de-tokenizer.** Human motion (especially dance) encompasses intricate details. Previous works [21, 60] keep the motion de-tokenizer fixed during training LLM, which hinders LLM's ability to perceive the distribution and details of motions. Specifically, in the output space of LLM, different motion tokens are treated as independent classes; therefore, the cost of classifying a motion token as semantic-similar token and semantic-distant token is the same. Apparently, relying solely on LLM's autoregressive loss is insufficient for capturing the details of motion. To address this problem, we jointly optimize LLM and motion de-tokenizer in stage2 and stage3. This strategy enables the reconstruction error signals in raw motion space to backpropagate to LLM, enhancing LLM's ability to generate the details of motion. With the goal of minimizing L1 loss between the predicted and real motion, we search for the motion's token sequence that could minimize this L1 loss in original motion space. As the motion de-tokenizer continuously optimizes, the target motion's token sequence, which supervises LLM training, dynamically changes. This dynamic adjustment reduces L1 loss progressively, achieving joint optimization.

- **Synergy learning of multitasks.** Although aligning text with one additional modality is relatively straightforward, integrating multiple modalities (*e.g.*, motion, text, and music) within a single framework poses a significant challenge. Additionally, as noted in [4], multitask joint training usually achieves inferior performance on each individual task compared to single-task training. This phenomenon is also observed in our *text-to-motion* task, as shown in Tab. 2. We argue that the large modality difference among different motion-relevant tasks (*e.g.*, *music-to-dance* and *text-to-motion*) prevents the model from effectively establishing connections between these tasks. Thus it is difficult for the model to identify a common optimization direction that benefits all tasks.

  As 'text' serves as a highly semantic descriptor for other modalities and is the most familiar and well-modeled modality to LLM, we use 'text' as a bridge to align motion, text, and music data, thereby mitigating conflicts in aligning multiple modalities. Initially, we construct paired textual descriptions for music samples in the dance datasets. Specifically, we use the style annotations of the music to create paired texts, such as 'a person is dancing Jazz'. Then, we construct two auxiliary tasks using the generated pairs of music and text, *i.e.*, *music-to-text* and *text-to-dance*. Through these two auxiliary tasks, M^{3}GPT implicitly learns to decompose the complex *music-to-dance* task into two simpler tasks *music-to-text* and *text-to-dance*. Additionally, with a shared tokenizer for motion and dance, *text-to-dance* and *text-to-motion* tasks occupy the same matching space, and thus can mutually reinforce each other. In this way, M^{3}GPT builds the synergies between the two primary motion generation tasks, music-to-dance and text-to-motion, facilitating mutual reinforcement, as shown in Tab. 2.

Combining the above analysis, we jointly train LLM and motion de-tokenizer using a mixture of motion comprehension and generation tasks, along with two auxiliary *music-to-text* and *text-to-dance* tasks. Besides the auxiliary tasks, we consider 2 basic motion comprehension tasks, *i.e.*, motion-to-text and dance-to-music, and 4 basic motion generation tasks, *i.e.*, text-to-motion, music-to-dance, motion prediction and motion in-between. Formally, for a specific task, we denote the source input consisting of a sequence of tokens as $\boldsymbol{q}_s = \left\{\boldsymbol{q}_s^i\right\}_{i=1}^{L_s}$, the target output as $\boldsymbol{q}_t = \left\{\boldsymbol{q}_t^i\right\}_{i=1}^{L_t}$, LLM predicts the probability distribution of potential next token at each step $p_\theta\left(\boldsymbol{q}_t^i|\boldsymbol{q}_t^{<i}, \boldsymbol{q}_s\right)$ in an autoregressive manner. For motion generation tasks, we add a reconstruction loss. Specifically, when the output tokens are motion tokens, we pass them to motion de-tokenizer to generate a motion sequence (denoted as $\hat{\boldsymbol{m}}$), where a reconstruction loss is then employed for guidance. Overall, during this training process, the objective is to maximize the log-likelihood of the data distribution and minimize the reconstruction error within raw motion space:

$$\mathcal{L} = \sum_{i=0}^{L_t-1} \log p_\theta\left(\boldsymbol{q}_t^i|\boldsymbol{q}_t^{<i}, \boldsymbol{q}_s\right) + \lambda \left\|\hat{\boldsymbol{m}} - \boldsymbol{m}\right\|_1, \tag{4}$$

where $\boldsymbol{m}$ denotes the ground-truth for $\hat{\boldsymbol{m}}$ generated by motion de-tokenizer, and λ is a hyper-parameter to adjust the weight for reconstruction loss.

Stage3: Instruction Fine-Tuning. To enhance the generalization and instruction-following capability of M^{3}GPT, we construct a multimodal instruction dataset with resort to GPT-4, building upon existing motion datasets. Specifically, we define 11 core tasks, each comprising $200/50/50$ training/validation/test instruction templates. For example, an instruction for text-to-motion task could be "Create a motion that complements the poetic elements in <Caption_Placeholder>", with <Caption_Placeholder> standing for any text sequence; an instruction for music-to-dance could be "Script a dance that adapts to the tempo shifts in <Audio_Placeholder>", with <Audio_Placeholder> standing for any music sequence. Further details are available in Appendix B.4.

3.4 Inference M^{3}GPT

During inference, we evaluate M^{3}GPT's performance across multiple motion-relevant tasks and datasets (Sec. 4.2 and Appendix (C, D). Also, we consider two challenging dance generation tasks to evaluate the zero-shot generalization ability of M^{3}GPT:

(1) Generating long-duration dances from long music. Long-duration dance generation involves creating uninterrupted, coherent dance sequences based on a single piece of music. Due to the limitation of computational cost and memory overload, we train M^{3}GPT on the task of 5-second music-to-dance generation. Conversely, during inference, we can combine this training task *music-to-dance* to generate an initial 5-second dance segment, and an unseen zero-shot task *music+dance-to-dance* that recursively generates subsequent dance segments conditioned on both music and previously generated dance segments, to perform long-duration and coherent dance generation.

(2) Generating dance controlled by both music and text. Integrating music and text as control signals in dance generation (*music+text-to-dance*) augments music-to-dance task with text modality. This process guides the generated dances to synchronize with particular actions described in input texts. Thanks to the tokenizer mechanism, M^{3}GPT can seamlessly combine music and text in LLM's input, enabling the integration of text instructions to produce a wide variety of dance sequences.

4 Experiments

4.1 Experimental setup

Datasets and Preprocessing. We use a large-scale text-to-motion dataset: Motion-X [29], and two music-to-dance datasets: AIST++ [24] and FineDance [25]. Notably, the 3D pose annotations differ among these datasets, therefore, we standardize their processing for uniform usage. Specifically, we select 22 joints common to these datasets and preprocess the motion samples following [13], resulting in motion sequences with identical representation. Further details on datasets and preprocessing are provided in Appendix B.1.

Evaluation Metrics. Different tasks employ distinct evaluation metrics. We use the most common evaluation metrics to assess the performance of M^{3}GPT for each task. **(1) Text-to-Motion**. Following [21, 12], we use *Frechet Inception Distance (FID)*, *Diversity (Div)*, *R-Precition* that calculates the top-1 motion-to-text retrieval accuracy (R TOP1). **(2) Motion-to-Text**. Following [21], we use linguistic metrics like *BLEU*, *CIDEr*, along with *R-Precision* for evaluating motion-to-text alignment. **(3) Music-to-Dance**. Following [26, 52], we use *FID*, *Diversity* and *Beat Align Score (BAS)* on kinetic features [22] (denoted as "k") to evaluate the dance generation quality. Notably, as noted in [48], the geometric features [33] are unreliable as a measure of dance generation quality. So we only use the kinetic features for evaluation. **(4) Dance-to-Music**. Following [52], we use *Beats Coverage Scores (BCS)*, *Beats Hit Scores (BHS)*, and *F1* score to evaluate music generation quality. **(5) Motion Prediction and In-Between**. Following [21], we use *FID* and *Diversity* to measure the consistency between the provided pose conditions and generated motion. More details and results on other evaluation metrics are provided in Appendix B.2 and C.

Implementation Details. For motion tokenizer, we set the codebook size to 512. As for music tokenizer, we use the pre-trained VQ-VAE from Jukebox with a codebook size of 2048. In term of temporal downsampling rate, the motion encoder uses a rate of 4, while the music encoder uses a rate of 128. We utilize T5 base [39] as our language model backbone. For training the motion tokenizer, we use Adam as the optimizer with a batch size of 1000 and an initial learning rate of 10^{-4}. To train the language model backbone, we employ the Adafactor_dev optimizer and use CosineAnnealingLR as the learning rate scheduler. The learning rate is set to 2×10^{-4} for pre-training stage, and 10^{-4} for instruction fine-tuning stage. For hyperparameter settings, λ in Eq. 4 is set to 0.2, and β in Eq. 2 is set to 0.02. All our experiments are conducted on 8 NVIDIA A40 GPUs. To evaluate the model's performance across different platforms, we also test our trained M^{3}GPT with T5-base on Ascend 910B NPUs. Further details on implementation and hyperparameter analysis are provided in Appendix E.

Table 2: Evaluation of synergy learning and joint optimization of LLM and motion de-tokenizer on Text-to-Motion (Motion-X [29]) and Music-to-Dance (AIST++ [24]). T2M: Text-to-Motion. A2D: Music-to-Dance. T2D: Text-to-Dance. A2T: Music-to-Text. *Trained single task* refers to a model trained and tested on a single task. *Pre-trained* and *Instruction-tuned* indicate the model after pre-training (stage2) and instruction tuning (stage3), followed by direct testing on each task. The arrows (↑) indicate that higher values are better. The arrows (↓) indicate that smaller values are better. **Bold** indicates the best result.

Methods	Re-Optimizing motion de-tokenizer	Text-to-Motion			Music-to-Dance		
		R TOP1 ↑	FID ↓	Div↑	FID_k ↓	Div_k ↑	BAS ↑
Ground Truth	-	0.675	0.009	2.316	17.10	8.19	0.2374
Trained single task		0.645	0.081	2.124	83.33	5.18	0.1835
Trained single task	✓	**0.656**	**0.078**	2.133	75.47	5.57	0.1884
T2M+A2D		0.564	0.094	2.080	51.26	6.73	0.2037
T2M+A2D	✓	0.578	0.092	2.106	47.71	7.47	0.1958
T2M+A2D+T2D+A2T		0.617	0.093	2.110	42.70	7.54	0.2084
T2M+A2D+T2D+A2T	✓	0.626	0.088	2.197	25.24	**7.63**	**0.2217**
M^3GPT (Pretrained without T2D and A2T)		0.526	0.105	2.058	40.71	7.47	0.2030
M^3GPT (Pretrained without T2D and A2T)	✓	0.547	0.104	2.099	37.14	7.61	0.2005
M^3GPT (Pre-trained)		0.598	0.089	2.218	32.71	7.43	0.2090
M^3GPT (Pre-trained)	✓	0.601	0.092	2.251	27.65	7.52	0.2105
M^3GPT (Instruction-tuned)		0.606	0.091	2.251	28.46	7.49	0.2052
M^3GPT (Instruction-tuned)	✓	0.615	0.093	**2.253**	**24.34**	7.50	0.2056

4.2 Ablation Studies

In this section, we conduct ablation studies to validate the effectiveness of our method. We use the same model architecture throughout the experiments. The ablation results are shown in Tab. 2.

Effectiveness of joint optimization of LLM and motion de-tokenizer. Different from previous works [21, 60] that fix motion de-tokenizer during training LLM, we jointly optimize LLM and motion de-tokenizer in stage2 and stage3, as detailed in Sec. 3.3. As shown in Tab. 2, the joint optimization consistently brings performance gains across various metrics and most settings. Specifically, it largely enhances the fidelity of generated dances, reflected in a notable decrease in FID_k score. We also notice a minor increase (less than 0.003) in FID of text-to-motion task in M^3GPT. The possible reason is that the motion patterns controlled by text are relatively simple, making LLM optimized solely in discrete semantic space adequate for text-to-motion. Conversely, dances involve greater complexity, necessitating the joint optimization of motion decoder to accurately capture intricate dance movements without compromising information.

Effectiveness of synergy learning of multitasks. During the training of M^3GPT, we introduce a synergy multitask learning strategy by constructing two auxiliary tasks: Text-to-Dance (T2D) and Music-to-Text (A2T), as detailed in Sec. 3.3. As shown in Tab. 2, the inclusion of T2D and A2T consistently brings performance gains across various metrics on both text-to-motion and music-to-dance tasks. Specifically, for music-to-dance, the FID_k score is decreased by nearly 10 points, indicating that the synergy learning helps generate more realistic dances. We argue that by incorporating these two auxilary tasks, M^3GPT implicitly learns to decompose the complex music-to-dance into two simpler tasks, music-to-text and text-to-dance. This way, the text-to-motion task can assist in learning the music-to-dance task, thereby enhancing its performance.

4.3 Comparisons with State-of-the-arts

In this section, we compare our M^3GPT with state-of-the-arts on multiple core motion-relevant tasks. We respectively report the comparison results on text-to-motion dataset, Motion-X [29], and music-to-dance datasets, AIST++ [24] and FineDance [25]. More quantitative and qualitative results are provided in Appendix C and D. Also, in the supplementary material's zip file, we provide the render videos of generated motions/dances and generated music files by our M^3GPT.

Main results on text-to-motion dataset. On the text-to-motion dataset, Motion-X, we evaluate M^3GPT on 4 tasks, *i.e.*, text-to-motion, motion-to-text, motion prediction, and motion in-between. The comparison results are shown in Tab. 3. As shown, M^3GPT achieves competitive performance across all evaluated tasks, highlighting its capability to address diverse motion tasks in a single model. Also, for text-to-motion task, *M^3GPT (instruction-tuned only T2M)*, which combines multitask pre-training and instruction fine-tuning solely on T2M task, yields better performance than *Trained single*

Table 3: Comparison results on Motion-X [29] dataset. The evaluation metrics are computed using the encoder introduced in Appendix A. Empty columns of previous methods indicate that they can not handle the task. *Instruction-tuned only T2M* indicates the model that is initially pre-trained on multiple tasks, followed by instruction tuning solely on text-to-motion task.

Methods	Text-to-Motion			Motion-to-Text			Motion Prediction		Motion In-between	
	R TOP1↑	FID↓	Div↑	R TOP3↑	Bleu@4↑	CIDEr↑	FID↓	Div↑	FID↓	Div↑
Real	$0.675^{\pm0.003}$	$0.009^{\pm0.000}$	$2.316^{\pm0.011}$	0.881	-	-	0.009	2.316	0.009	2.316
MLD [5]	$0.612^{\pm0.003}$	$0.122^{\pm0.008}$	$2.267^{\pm0.018}$	-	-	-	-	-	-	-
T2M-GPT [55]	$0.647^{\pm0.002}$	$0.101^{\pm0.005}$	$\underline{2.270}^{\pm0.033}$	-	-	-	0.814	1.755	-	-
MotionDiffuse [57]	$0.659^{\pm0.002}$	$\underline{0.075}^{\pm0.004}$	$2.220^{\pm0.022}$	-	-	-	-	-	-	-
TM2T [15]	$0.581^{\pm0.002}$	$0.148^{\pm0.003}$	$2.005^{\pm0.034}$	0.806	**12.13**	20.16	-	-	-	-
MDM [44]	$0.472^{\pm0.008}$	$0.078^{\pm0.000}$	$2.133^{\pm0.012}$	-	-	-	1.028	1.746	0.831	1.768
MoMask [16]	$\mathbf{0.668}^{\pm0.003}$	$\mathbf{0.074}^{\pm0.004}$	$2.241^{\pm0.016}$	-	-	-	-	-	0.626	$\underline{1.884}$
MotionLCM [8]	$0.658^{\pm0.005}$	$0.078^{\pm0.003}$	$2.206^{\pm0.026}$	-	-	-	-	-		
MotionGPT[21]	$0.659^{\pm0.003}$	$0.078^{\pm0.001}$	$2.166^{\pm0.026}$	$\underline{0.840}$	11.21	$\underline{31.36}$	$\underline{0.701}$	1.818	0.648	1.875
Trained single task	$0.656^{\pm0.002}$	$0.078^{\pm0.002}$	$2.133^{\pm0.012}$	0.767	10.14	22.92	0.774	1.778	0.692	1.810
M^{3}GPT (Pre-trained)	$0.601^{\pm0.002}$	$0.092^{\pm0.002}$	$2.251^{\pm0.012}$	0.834	11.00	24.12	0.707	**1.874**	**0.604**	1.879
M^{3}GPT (Instruction-tuned)	$0.615^{\pm0.003}$	$0.093^{\pm0.002}$	$2.253^{\pm0.026}$	**0.845**	$\underline{11.50}$	**42.93**	**0.682**	$\underline{1.838}$	$\underline{0.612}$	**1.900**
M^{3}GPT (Instruction-tuned only T2M)	$\underline{0.661}^{\pm0.003}$	$0.076^{\pm0.002}$	$\mathbf{2.273}^{\pm0.026}$	-	-	-	-	-	-	-

Table 4: Comparison results on Motion-X [29] dataset based on Ascend 910B NPUs.

Methods	Text-to-Motion			Motion-to-Text			Motion Prediction		Motion In-between	
	R TOP1↑	FID↓	Div↑	R TOP3↑	Bleu@4↑	CIDEr↑	FID↓	Div↑	FID↓	Div↑
Trained single task	$\underline{0.654}^{\pm0.002}$	$\underline{0.081}^{\pm0.003}$	$\underline{2.304}^{\pm0.017}$	0.763	10.16	22.89	0.776	1.818	0.712	$\underline{1.880}$
M^{3}GPT (Pre-trained)	$0.596^{\pm0.002}$	$0.096^{\pm0.003}$	$2.241^{\pm0.018}$	$\underline{0.831}$	$\underline{11.05}$	$\underline{24.03}$	$\underline{0.710}$	**1.882**	**0.608**	1.874
M^{3}GPT (Instruction-tuned)	$0.612^{\pm0.002}$	$0.094^{\pm0.002}$	$2.276^{\pm0.021}$	**0.846**	**11.52**	**42.64**	**0.684**	$\underline{1.841}$	$\underline{0.621}$	**1.903**
M^{3}GPT (Instruction-tuned only T2M)	$\mathbf{0.659}^{\pm0.003}$	$\mathbf{0.078}^{\pm0.003}$	$\mathbf{2.314}^{\pm0.023}$	-	-	-	-	-	-	-

Table 5: Comparison results on AIST++ [24] and FineDance [25] datasets.

Methods	Music-to-Dance on AIST++			Music-to-Dance on FineDance			Dance-to-Music on AIST++		
	FID$_k$ ↓	Div$_k$ ↑	BAS ↑	FID$_k$ ↓	Div$_k$ ↑	BAS ↑	BCS↑	BHS↑	F1↑
Real	17.10	10.60	0.2374	-	-	0.2120	-	-	-
FACT [24]	35.35	5.94	0.2209	113.38	3.36	0.1831	-	-	-
Bailando [42]	28.16	$\underline{7.83}$	0.2332	82.81	7.74	0.2029	-	-	-
EDGE [48]	42.16	3.96	$\underline{0.2334}$	94.34	$\underline{8.13}$	0.2116	-	-	-
Lodge [26]	37.09	5.58	**0.2423**	$\underline{45.56}$	6.75	**0.2397**	-	-	-
Foley [11]	-	-	-	-	-	-	96.4	41.0	57.5
CMT [10]	-	-	-	-	-	-	$\underline{97.1}$	46.2	62.6
D2MGAN [64]	-	-	-	-	-	-	95.6	88.7	93.1
CDCD [65]	-	-	-	-	-	-	96.5	89.3	92.7
LORIS [52]	-	-	-	-	-	-	**98.6**	90.8	94.5
Trained single task	75.47	5.57	0.1884	128.37	6.48	0.2036	93.9	93.6	92.8
M^{3}GPT (Pre-trained)	27.65	7.52	0.2105	92.35	7.67	0.2134	93.4	$\underline{93.8}$	94.2
M^{3}GPT (Instruction-tuned)	$\underline{24.34}$	7.50	0.2056	86.47	7.75	0.2158	93.6	**94.0**	$\underline{94.9}$
M^{3}GPT (Instruction-tuned for single task)	**23.01**	**7.85**	0.2261	**42.66**	**8.24**	$\underline{0.2231}$	94.3	**94.0**	**95.0**

Table 6: Comparison results on AIST++ [24] and FineDance [25] datasets based on Ascend 910B NPUs.

Methods	Music-to-Dance on AIST++			Music-to-Dance on FineDance			Dance-to-Music on AIST++		
	FID$_k$ ↓	Div$_k$ ↑	BAS ↑	FID$_k$ ↓	Div$_k$ ↑	BAS ↑	BCS↑	BHS↑	F1↑
Trained single task	77.32	5.61	0.1860	134.66	6.52	0.2088	$\underline{93.8}$	93.6	92.7
M^{3}GPT (Pre-trained)	27.99	$\underline{7.61}$	$\underline{0.2102}$	91.39	7.71	0.2128	93.4	93.6	94.2
M^{3}GPT (Instruction-tuned)	$\underline{25.05}$	7.48	0.2072	$\underline{88.25}$	$\underline{7.76}$	$\underline{0.2160}$	93.5	$\underline{93.8}$	$\underline{94.7}$
M^{3}GPT (Instruction-tuned for single task)	**23.68**	**7.83**	**0.2264**	**43.78**	**8.39**	**0.2225**	**94.3**	**94.0**	**94.9**

task that only trains the model on T2M task. Tab. 4 presents the results of testing on the Ascend 910B NPUs, where M^{3}GPT also achieves comparably good performance. These results demonstrate that multitask pre-training can enhance the performance of individual tasks across different computation platforms. Further results of M^{3}GPT trained on NPUs will be presented later.

Main results on music-to-dance datasets. On the music-to-dance datasets, AIST++ and FineDance, we evaluate M^{3}GPT on 2 tasks, *i.e.*, music-to-dance and dance-to-music. As shown in Tab. 5, in general, the performance of multitask pre-training and instruction fine-tuning in M^{3}GPT outperforms single-task training, underscoring the effectiveness of multitask training for dance-relevant tasks. Also, M^{3}GPT achieves competitive performance on most metrics. For music-to-dance, the best-performing

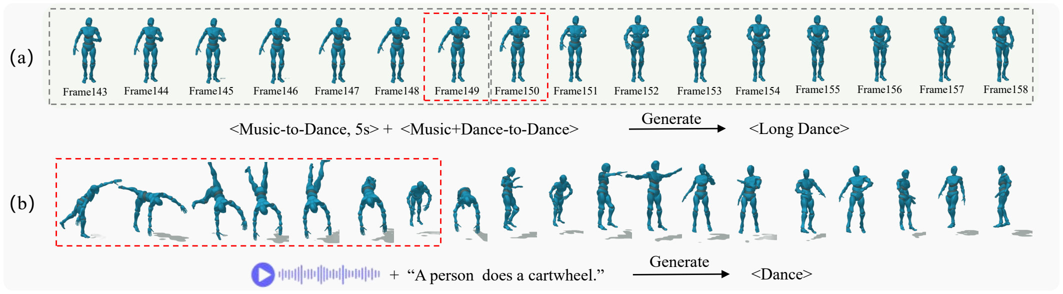


Figure 3: Qualitative results for long-term dance and music-text conditioned dance generation of M^{3}GPT.

method on the FineDance dataset is Lodge [26]. This approach features a specialized two-stage architecture for generating long-duration dance sequences, progressively refining movements from coarse to fine granularity using a diffusion model. On AIST++ dataset, M^{3}GPT reports the best FID$_k$ of 24.34 for music-to-dance task, the best BHS and F1 of 94.0 and 94.9 for dance-to-music task. The results in Tab. 6, tested on the Ascend 910B NPUs, also demonstrate that multitask training can enhance the performance of both music-to-dance and dance-to-music tasks.

4.4 Evaluation on Zero-Shot Tasks

In this section, we explore M^{3}GPT's capabilities in handling zero-shot tasks, as mentioned in Sec. 3.4. Fig. 3 (a) shows the long-term dance generation. As seen, M^{3}GPT can generate a coherent dance sequence by seamlessly integrating the music-to-dance and zero-shot *music+dance-to-dance* tasks. Fig. 3 (b) shows the generated 3D dance with both music and text instruction. We can see that M^{3}GPT maintains plausible visual results in accordance with input text instructions (*cartweel*), underscoring its remarkable zero-shot generalization capability.

5 Conclusion

In this paper, we present M^{3}GPT, a unified framework for comprehending and generating motion aligned with both text and music modalities. By employing text as a bridge, we build connections and synergies between different motion-relevant tasks, facilitating mutual reinforcement. Additionally, the joint optimization of LLM and motion de-tokenizer further enriches the details of generated motion, enhancing overall motion generation quality. Our extensive evaluations across various motion-relevant tasks demonstrate the effectiveness of M^{3}GPT in both motion comprehension and generation. Besides, M^{3}GPT exhibits strong zero-shot generalization abilities, enabling it to handle previously unseen and challenging motion-relevant tasks.

Limitations and Broader Impacts. Although our M^{3}GPT has successfully processed signals from motion, text, and music modalities for motion comprehension and generation, it focuses on modeling human body movements, excluding hands and faces details. Future research can extend the scope of M^{3}GPT to include hands and facial modeling.

Acknowledgments. This work is sponsored by the National Natural Science Foundation of China (NSFC) under Grant 62306301, the National Postdoctoral Program for Innovative Talents under Grant BX20220310, and the CAAI-CANN Open Fund developed on the OpenI Community. It is also supported by the project of Peng Cheng Laboratory (PCL2023A08).

References

[1] Judith Butepage, Michael J Black, Danica Kragic, and Hedvig Kjellstrom. Deep representation learning for human motion prediction and classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6158–6166, 2017.

[2] Ke Chen, Yusong Wu, Haohe Liu, Marianna Nezhurina, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Musicldm: Enhancing novelty in text-to-music generation using beat-synchronous mixup strategies. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1206–1210. IEEE, 2024.

[3] Ling-Hao Chen, Jiawei Zhang, Yewen Li, Yiren Pang, Xiaobo Xia, and Tongliang Liu. Humanmac: Masked motion completion for human motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9544–9555, 2023.

[4] Ting Chen, Saurabh Saxena, Lala Li, Tsung-Yi Lin, David J Fleet, and Geoffrey E Hinton. A unified sequence interface for vision tasks. *Advances in Neural Information Processing Systems*, 35:31333–31346, 2022.

[5] Xin Chen, Biao Jiang, Wen Liu, Zilong Huang, Bin Fu, Tao Chen, and Gang Yu. Executing your commands via motion diffusion in latent space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18000–18010, 2023.

[6] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, march 2023. *URL https://lmsys. org/blog/2023-03-30-vicuna*, 3(5), 2023.

[7] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.

[8] Wenxun Dai, Ling-Hao Chen, Jingbo Wang, Jinpeng Liu, Bo Dai, and Yansong Tang. Motionlcm: Real-time controllable motion generation via latent consistency model. *arXiv preprint arXiv:2404.19759*, 2024.

[9] Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*, 2020.

[10] Shangzhe Di, Zeren Jiang, Si Liu, Zhaokai Wang, Leyan Zhu, Zexin He, Hongming Liu, and Shuicheng Yan. Video background music generation with controllable music transformer. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 2037–2045, 2021.

[11] Chuang Gan, Deng Huang, Peihao Chen, Joshua B Tenenbaum, and Antonio Torralba. Foley music: Learning to generate music from videos. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 758–775. Springer, 2020.

[12] Kehong Gong, Dongze Lian, Heng Chang, Chuan Guo, Zihang Jiang, Xinxin Zuo, Michael Bi Mi, and Xinchao Wang. Tm2d: Bimodality driven 3d dance generation via music-text integration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9942–9952, 2023.

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

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

[15] Chuan Guo, Xinxin Zuo, Sen Wang, and Li Cheng. Tm2t: Stochastic and tokenized modeling for the reciprocal generation of 3d human motions and texts. In *European Conference on Computer Vision*, pages 580–597. Springer, 2022.

[16] Chuan Guo, Yuxuan Mu, Muhammad Gohar Javed, Sen Wang, and Li Cheng. Momask: Generative masked modeling of 3d human motions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1900–1910, 2024.

[17] Hao Guo, Xiaochuan Fan, and Song Wang. Visual attention consistency for human attribute recognition. *International Journal of Computer Vision*, 130(4):1088–1106, 2022.

[18] Bo Han, Yi Ren, and Yuheng Li. Dance2midi: Dance-driven multi-instruments music generation. *arXiv preprint arXiv:2301.09080*, 2:3, 2023.

[19] Ruibing Hou, Bingpeng Ma, Hong Chang, Xinqian Gu, Shiguang Shan, and Xilin Chen. Feature completion for occluded person re-identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44 (9):4894–4912, 2021.

[20] 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.

[21] Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. Motiongpt: Human motion as a foreign language. *Advances in Neural Information Processing Systems*, 36, 2024.

[22] and Jessica K Hodgins Kensuke Onuma, Christos Faloutsos. Fmdistance: A fast and effective distance function for motion capture data. In *Eurographics*, pages 83–86, 2008.

[23] Peike Li, Yunqiu Xu, Yunchao Wei, and Yi Yang. Self-correction for human parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(6):3260–3271, 2020.

[24] Ruilong Li, Shan Yang, David A Ross, and Angjoo Kanazawa. Ai choreographer: Music conditioned 3d dance generation with aist++. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13401–13412, 2021.

[25] Ronghui Li, Junfan Zhao, Yachao Zhang, Mingyang Su, Zeping Ren, Han Zhang, Yansong Tang, and Xiu Li. Finedance: A fine-grained choreography dataset for 3d full body dance generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10234–10243, 2023.

[26] Ronghui Li, YuXiang Zhang, Yachao Zhang, Hongwen Zhang, Jie Guo, Yan Zhang, Yebin Liu, and Xiu Li. Lodge: A coarse to fine diffusion network for long dance generation guided by the characteristic dance primitives. *arXiv preprint arXiv:2403.10518*, 2024.

[27] Sifei Li, Weiming Dong, Yuxin Zhang, Fan Tang, Chongyang Ma, Oliver Deussen, Tong-Yee Lee, and Changsheng Xu. Dance-to-music generation with encoder-based textual inversion of diffusion models. *arXiv preprint arXiv:2401.17800*, 2024.

[28] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[29] Jing Lin, Ailing Zeng, Shunlin Lu, Yuanhao Cai, Ruimao Zhang, Haoqian Wang, and Lei Zhang. Motion-x: A large-scale 3d expressive whole-body human motion dataset. *Advances in Neural Information Processing Systems*, 2023.

[30] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 851–866. 2023.

[31] Julieta Martinez, Michael J Black, and Javier Romero. On human motion prediction using recurrent neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2891–2900, 2017.

[32] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In *SciPy*, pages 18–24, 2015.

[33] Meinard Müller, Tido Röder, and Michael Clausen. Efficient content-based retrieval of motion capture data. In *ACM SIGGRAPH*, pages 677–685. 2005.

[34] Boris N Oreshkin, Antonios Valkanas, Félix G Harvey, Louis-Simon Ménard, Florent Bocquelet, and Mark J Coates. Motion in-betweening via deep $delta$-interpolator. *IEEE Transactions on Visualization and Computer Graphics*, 2023.

[35] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[36] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10975–10985, 2019.

[37] Matthias Plappert, Christian Mandery, and Tamim Asfour. Learning a bidirectional mapping between human whole-body motion and natural language using deep recurrent neural networks. *Robotics and Autonomous Systems*, 109:13–26, 2018.

[38] 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 *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[39] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[40] Mikel Rodriguez, Ivan Laptev, Josef Sivic, and Jean-Yves Audibert. Density-aware person detection and tracking in crowds. In *2011 International Conference on Computer Vision*, pages 2423–2430. IEEE, 2011.

[41] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *arXiv preprint arXiv:2201.02610*, 2022.

[42] Li Siyao, Weijiang Yu, Tianpei Gu, Chunze Lin, Quan Wang, Chen Qian, Chen Change Loy, and Ziwei Liu. Bailando: 3d dance generation by actor-critic gpt with choreographic memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11050–11059, 2022.

[43] Paul Starke, Sebastian Starke, Taku Komura, and Frank Steinicke. Motion in-betweening with phase manifolds. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 6(3):1–17, 2023.

[44] Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-or, and Amit Haim Bermano. Human motion diffusion model. In *The Eleventh International Conference on Learning Representations*, 2022.

[45] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1653–1660, 2014.

[46] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[47] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[48] Jonathan Tseng, Rodrigo Castellon, and Karen Liu. Edge: Editable dance generation from music. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 448–458, 2023.

[49] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015.

[50] Shengqiong Wu, Hao Fei, Leigang Qu, Wei Ji, and Tat-Seng Chua. Next-gpt: Any-to-any multimodal llm. *arXiv preprint arXiv:2309.05519*, 2023.

[51] Tatsuro Yamada, Hiroyuki Matsunaga, and Tetsuya Ogata. Paired recurrent autoencoders for bidirectional translation between robot actions and linguistic descriptions. *IEEE Robotics and Automation Letters*, 3(4): 3441–3448, 2018.

[52] Jiashuo Yu, Yaohui Wang, Xinyuan Chen, Xiao Sun, and Yu Qiao. Long-term rhythmic video soundtracker. In *International Conference on Machine Learning*, pages 40339–40353. PMLR, 2023.

[53] Jun Zhan, Junqi Dai, Jiasheng Ye, Yunhua Zhou, Dong Zhang, Zhigeng Liu, Xin Zhang, Ruibin Yuan, Ge Zhang, Linyang Li, et al. Anygpt: Unified multimodal llm with discrete sequence modeling. *arXiv preprint arXiv:2402.12226*, 2024.

[54] Dong Zhang, Shimin Li, Xin Zhang, Jun Zhan, Pengyu Wang, Yaqian Zhou, and Xipeng Qiu. Speechgpt: Empowering large language models with intrinsic cross-modal conversational abilities. *arXiv preprint arXiv:2305.11000*, 2023.

[55] Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Shaoli Huang, Yong Zhang, Hongwei Zhao, Hongtao Lu, and Xi Shen. T2m-gpt: Generating human motion from textual descriptions with discrete representations. *arXiv preprint arXiv:2301.06052*, 2023.

[56] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14730–14740, 2023.

[57] Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. Motiondiffuse: Text-driven human motion generation with diffusion model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[58] Mingyuan Zhang, Daisheng Jin, Chenyang Gu, Fangzhou Hong, Zhongang Cai, Jingfang Huang, Chongzhi Zhang, Xinying Guo, Lei Yang, Ying He, et al. Large motion model for unified multi-modal motion generation. *arXiv preprint arXiv:2404.01284*, 2024.

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

[60] 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7368–7376, 2024.

[61] Jiahe Zhao, Ruibing Hou, Hong Chang, Xinqian Gu, Bingpeng Ma, Shiguang Shan, and Xilin Chen. Clothes-changing person re-identification with feasibility-aware intermediary matching. *arXiv preprint arXiv:2404.09507*, 2024.

[62] Zixiang Zhou and Baoyuan Wang. Ude: A unified driving engine for human motion generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5632–5641, 2023.

[63] Zixiang Zhou, Yu Wan, and Baoyuan Wang. A unified framework for multimodal, multi-part human motion synthesis. *arXiv preprint arXiv:2311.16471*, 2023.

[64] Ye Zhu, Kyle Olszewski, Yu Wu, Panos Achlioptas, Menglei Chai, Yan Yan, and Sergey Tulyakov. Quantized gan for complex music generation from dance videos. In *European Conference on Computer Vision*, pages 182–199. Springer, 2022.

[65] Ye Zhu, Yu Wu, Kyle Olszewski, Jian Ren, Sergey Tulyakov, and Yan Yan. Discrete contrastive diffusion for cross-modal music and image generation. *arXiv preprint arXiv:2206.07771*, 2022.

Appendix

A Text-Motion Alignment Model

Due to the lack of a powerful and publicly available text-motion alignment model, we independently leverage existing datasets to develop a functional text-motion alignment model, which is used to evaluate tasks such as text-to-motion, motion-to-text, motion prediction, and motion in-between. We adopt the motion encoder architecture E_m from [14] and use the pretrained CLIP [38] ViT-B/32 model for the text encoder E_t. As depicted in Fig. 4, the training of the text-motion alignment model is split into two phases: pre-training the motion auto-encoder and text-motion contrastive learning. During the pre-training phase, we use motion data from the text-to-motion, and dance data from the music-to-dance tasks. This stage employs a reconstruction loss to ensure the model achieves a robust initial state capable of extracting an expressive motion representation. In the text-motion contrastive learning phase, we utilize text-motion pair data from the text-to-motion task. To train a more robust text-motion alignment model, we employ text-motion training dataset data along with motion data reconstructed by Motion VQ-VAE for text-motion contrastive training. We incorporate an adapter MLP layer into both the motion and text encoders to align the dimensions of $\mathbf{z}_m$ and $\mathbf{z}_t$ at 512. This setup facilitates the alignment of text and motion in the representational space. The motion reconstruction loss $L_{\text{recon\_motion}}$ for the pre-training stage and the contrastive learning loss L_{infoNCE} for the aligning stage are used to optimize this model, as follows,

$$\mathcal{L}_{\text{recon\_motion}} = \|x - \tilde{x}\|^2 \tag{5}$$

$$\mathcal{L}_{\text{infoNCE}} = -\frac{1}{N}\sum_{i=1}^{K} \log\left(\frac{\exp(\langle z_i', z_i^t\rangle/\tau)}{\sum_{j=1}^{K}\exp(\langle z_i', z_j^t\rangle/\tau)}\right) \tag{6}$$

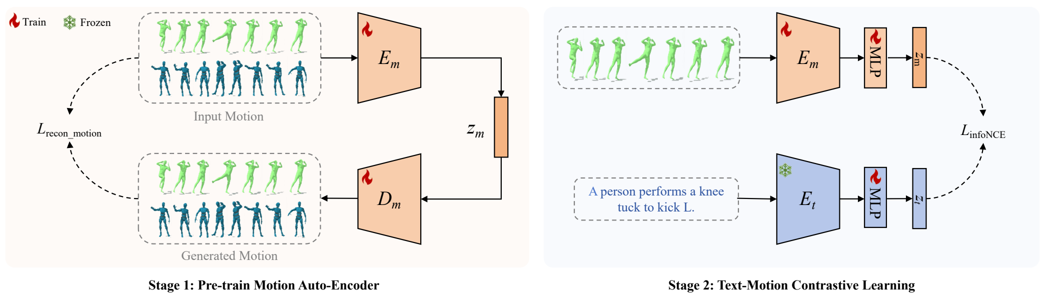

Train
Frozen
Input Motion
E_m
$L_{\text{recon\_motion}}$
z_m
D_m
Generated Motion
Stage 1: Pre-train Motion Auto-Encoder
E_m
MLP
z_m
L_{infoNCE}
A person performs a knee tuck to kick L.
E_t
MLP
z_t
Stage 2: Text-Motion Contrastive Learning

Figure 4: Pipeline of Text-Motion Alignment Model. The training of the text-motion alignment model includes two stages: pre-training motion auto-encoder and text-motion contrastive learning.

B Details for Training and Evaluating

B.1 Data Introductions and Preprocessing

We leverage the largest available text-to-motion dataset, Motion-X [29], along with widely-used music-to-dance datasets, AIST++ [24] and FineDance [25], for our multitask training regimen. Motion-X is used for text-to-motion, motion-to-text, motion prediction, and motion in-between tasks, while AIST++ and FineDance datasets support both music-to-dance and dance-to-music tasks, and are also adapted for motion prediction and in-between tasks to enhance our training resources.

Motion-X includes 15.6 million precise 3D whole-body SMPL-X [36] pose annotations across 81.1K motion sequences with sequence-level semantic text descriptions. AIST++ contains 1,409 dance motion sequences across 10 dance genres with SMPL [30] pose annotations, and FineDance provides 7.7 hours of dance, totaling 831,600 frames with SMPL-H [41] poses at 30 fps across 16 dance genres. Tab. 7 shows the training datasets and their corresponding sample numbers that we use to train our model.

We standardize data across these datasets by selecting the 22 common joints and normalizing each motion sequence to face the same direction and to run at 30 fps. We use a processing technique consistent with prior research [14, 5, 21] that integrates joint velocities, positions, and rotations for consistent motion representation, facilitating effective utilization across tasks. For the music data, we use the Librosa toolkit [32] to load raw .wav data at a sampling rate of 22050 Hz, processed into features by the Jukebox encoder [9]. To optimize the use of these datasets, we strategically employ data from specific datasets for different tasks. During training for music and dance tasks, we randomly select 5-second segments from complete music tracks and corresponding dance segments as training samples, setting the sample length to 6.25 seconds for motion prediction and in-between tasks with AIST++ and FineDance. When assessing music-to-dance on the FineDance dataset, we don't evaluate all 5-second samples directly. Instead, we generate continuous long dance sequences using music-to-dance and motion prediction, then segment these into 30-second samples for evaluation to align with Lodge's testing methodology.

Table 7: The training datasets and sample numbers for different tasks.

Tasks	T2M	M2T	Motion Prediction/In-between	A2D	D2A	A2T	T2D
Training dataset	Motion-X	Motion-X	Motion-X/AIST++/FineDance	AIST++/FineDance	AIST++/FineDance	AIST++/FineDance	AIST++/FineDance
Training samples number	64867	64867	64867/952/177	952/177	952/177	952/177	952/177

B.2 Comprehensive Evaluation Metrics

Different tasks utilize specific evaluation metrics. We use the consistent evaluation metrics following prior research [14, 21, 12, 26, 64, 52].

- Text-to-Motion. We measure the discrepancy between generated and actual motion features using Frechet Inception Distance (FID), assess diversity (Div) among all generated motion sequences, and evaluate motion-text semantic correlation with R-precision. Multimodal Distance (MM Dist) quantifies the disparity between motions and texts. A specialized model developed for evaluating the text-to-motion task on the Motion-X dataset with 22 joints is detailed in Appendix A.
- Motion-to-Text. We use linguistic metrics including BLEU [35], ROUGE [28], CIDEr [49], and BertScore [59], along with R-Precision and MM Dist to assess alignment between generated text and motion.
- Music-to-Dance. We employ the evaluation framework recommended by FACT [24] and Bailando [42], utilizing FID, Diversity, and Beat Align Score (BAS) to gauge dance quality. In our paper, we use kinetic features to compute FID and Diversity.
- Dance-to-Music. We use metrics from [52, 64] such as Beats Coverage Scores (BCS), Beats Hit Scores (BHS), F1 scores, Beats Coverage Std (CSD), and Beats Hit Std (HSD) to evaluate music generation quality.
- Motion Prediction and Motion In-between. We use Average Displacement Error (ADE) and Final Displacement Error (FDE) to assess the quality of predicted motion. The text-motion alignment model aids in evaluating motion prediction performance.

B.3 Distributed Training for Multitasks

We employ a single-node multi-GPU distributed training strategy for M^{3}GPT, distributing each task across separate GPUs to facilitate multitask training through shared model parameters. This method allows us to tailor the maximum token length of the LLM for each task, based on the longest sample sequence typical for that task. Specifically, we set the maximum LLM token length to 192 for tasks such as text-to-motion, motion-to-text, motion prediction, and motion in-between. For tasks involving the music modality, such as music-to-dance and dance-to-music, the maximum length is set at 980. This task-specific configuration enables us to optimize batch sizes effectively, thus maximizing GPU utilization.

In our experimental setup, the batch size is set to 40 for the text-to-motion, text-to-dance, and motion-to-text tasks, and 4 for the music-to-dance, dance-to-music, and music-to-text tasks. For motion prediction and motion in-between tasks, the batch size is set to 45. We also establish the number of iterations for pre-training at 1,000,000, instruction fine-tuning at 200,000. This structured approach ensures that each task is optimally processed, enhancing the efficiency and effectiveness of our training regimen.

B.4 Tasks for Pre-training and Instruction tuning

As shown in Fig. 5, we define 11 core motion-related tasks for the instruction tuning of M^{3}GPT, including text-to-motion, random-to-motion, motion-to-text, random-to-text, motion prediction, motion in-between, music-to-dance, dance-to-music, random-to-music, text-to-dance, and music-to-text. The tasks of text-to-dance and music-to-text were specifically constructed based on the music-to-dance datasets. Random-to-X represents the unconstrained generation of motion, text and music. In Tab. 8, we present a selection of command templates for each task, where <Motion_Placeholder>, <Caption_Placeholder>, and <Music_Placeholder> respectively represent the motion sequence (including dance sequence), textual description, and music segment from the training data.

C Quantitative Results and Comparisons with SOTA Methods

In this section, we compare the performance of our M^{3}GPT with existing SOTA methods on a broader set of metrics for each task across three datasets: Motion-X [29], AIST++ [24], and FineDance [25]. Tab. 9 shows

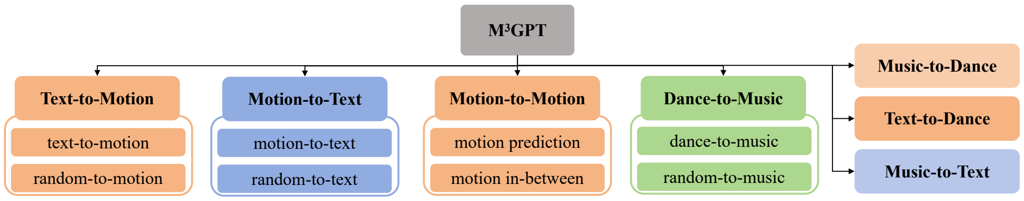


Figure 5: Tasks for M^{3}GPT pre-training and instruction tuning. Random represents the unconstrained generation of motion/text/music in the corresponding task.

Table 8: Examples of instruction templates for each task when instruction tuning M^{3}GPT.

Task	Instruction template	Output
text-to-motion	Design a motion that illustrates the emotions conveyed by <Caption_Placeholder>. How could you express the resilience mentioned in <Caption_Placeholder> through motion? Can you develop a motion that captures the existential debate in <Caption_Placeholder>?	<Motion_Placeholder>
random-to-motion	Can you generate a motion randomly? Please generate a random motion. Display a motion for me.	<Motion_Placeholder>
motion-to-text	What themes are explored through the motion in <Motion_Placeholder>? Can you describe the motion <Motion_Placeholder> with texts? How would you interpret the actions depicted in <Motion_Placeholder>?	<Caption_Placeholder>
random-to-text	Can you generate a text description for motion randomly? Give me a caption which describes a action. How can we describe a motion with texts?	<Caption_Placeholder>
motion prediction	What movements would suitably follow the thematic climax of <Motion_Placeholder>? What steps might deepen the emotional expression seen in <Motion_Placeholder>? What movements could follow to resolve the suspense built in <Motion_Placeholder>?	<Motion_Placeholder>
motion in-between	What new character dynamics could the middle section of <Motion_Placeholder> explore? Infer the type of dramatic climax that the masked section of <Motion_Placeholder> might contain. What potential themes of ascent or descent could be explored in the middle of <Motion_Placeholder>?	<Motion_Placeholder>
music-to-dance	Script a dance that adapts to the tempo shifts in <Music_Placeholder>. Create a dance that would visually mimic the lyrical journey in <Music_Placeholder>. Compose a dance that explores the genre characteristics of <Music_Placeholder>.	<Motion_Placeholder>
dance-to-music	Can you design a music for this dance <Motion_Placeholder>? Please create a music based on this dance <Motion_Placeholder>. Arrange a symphony that captures the shifts in <Motion_Placeholder>.	<Music_Placeholder>
random-to-music	Please generate a music for a dance randomly. Can you generate a music with dance style? Creat a music for any style dance.	<Music_Placeholder>
text-to-dance	Please generate a dance based on the caption <Caption_Placeholder>. Create a dance for the text <Caption_Placeholder>. Generate a dance that corresponds to the textual description <Caption_Placeholder>.	<Motion_Placeholder>
music-to-text	Describe the dance movements that correspond to the given music <Music_Placeholder>. Describe the dance actions that match the provided music <Music_Placeholder>. Detail the dance steps associated with the specified music <Music_Placeholder>.	<Caption_Placeholder>

the comparison of text-to-motion on Motion-X dataset. Tab. 10 shows the comparison of motion-to-text on Motion-X dataset. Tab. 11 shows the comparison of music-to-dance on AIST++ and FineDance datasets. Tab. 12 shows the comparison of dance-to-music on AIST++ and FineDance datasets. Tab. 13 and Tab. 14 shows the comparison of motion prediction and motion in-between on Motion-X dataset. Tab. 15 shows the comparison of music-to-text, text-to-dance and text-to-music on AIST++ dataset. Tab. 16 and Tab. 17 show the comparison of motion-related tasks among different size of T5. As shown in these tables, M^{3}GPT can achieve competitive performance with SOTAs across all evaluated tasks.

D Qualitative Results and Comparison with SOTA Methods

Fig. 6 presents visualizations for a variety of tasks, including text-to-motion, motion-to-text, motion prediction, motion in-between, music-to-dance, long-term dance generation, and music-text conditioned dance generation. The visualization results show that our method can generate realistic results across various motion-relevant tasks. Fig. 7 presents the qualitative results between different methods for text-to-motion and music-to-dance.

E Additional Experiments

The performance of text-to-motion based on different λ. In Tab. 18, the performance of M^{3}GPT on Motion-X dataset is analyzed across different values of λ for the text-to-motion task. The results indicate that as λ increases, the model's recall precision (Top1, Top2, Top3) initially improves but subsequently declines. The optimal performance is achieved at $\lambda = 0.2$, as evidenced by the highest scores in RPrecision and modality metrics.

Further increases in λ to 0.3 and 0.4 lead to deteriorating performance, particularly in FID and RPrecision, suggesting that excessive λ values may result in over-regularization or reduced adaptability of the model.

Table 9: Comparison of Text-to-Motion on Motion-X dataset [29]. The arrows (↑) indicate that higher values are better. The arrows (↓) indicate that smaller values are better. **Bold** and underline indicate the best and the second best result.

Methods	RPrecision ↑			FID ↓	MM Dist ↓	Div ↑	MModality ↑
	Top1	Top2	Top3				
Real	$0.675^{\pm 0.003}$	$0.821^{\pm 0.003}$	$0.878^{\pm 0.002}$	$0.009^{\pm 0.000}$	$2.938^{\pm 0.007}$	$2.316^{\pm 0.011}$	-
MDM [44]	$0.472^{\pm 0.008}$	$0.616^{\pm 0.005}$	$0.704^{\pm 0.003}$	$0.161^{\pm 0.006}$	$5.404^{\pm 0.031}$	$2.234^{\pm 0.015}$	$\underline{2.241}^{\pm 0.043}$
MLD [5]	$0.612^{\pm 0.003}$	$0.743^{\pm 0.002}$	$0.808^{\pm 0.004}$	$0.122^{\pm 0.008}$	$3.117^{\pm 0.035}$	$2.267^{\pm 0.018}$	$2.210^{\pm 0.055}$
T2M-GPT [55]	$0.647^{\pm 0.002}$	$0.785^{\pm 0.004}$	$0.845^{\pm 0.003}$	$0.101^{\pm 0.005}$	$3.046^{\pm 0.028}$	$\underline{2.270}^{\pm 0.033}$	$\mathbf{2.226}^{\pm 0.036}$
MotionDiffuse [55]	$\underline{0.659}^{\pm 0.002}$	$\underline{0.802}^{\pm 0.004}$	$\mathbf{0.865}^{\pm 0.002}$	$\mathbf{0.075}^{\pm 0.004}$	$2.944^{\pm 0.004}$	$2.220^{\pm 0.022}$	$2.102^{\pm 0.036}$
Trained single task	$0.656^{\pm 0.002}$	$0.795^{\pm 0.001}$	$0.843^{\pm 0.001}$	$0.078^{\pm 0.000}$	$\underline{2.942}^{\pm 0.001}$	$2.133^{\pm 0.012}$	$2.046^{\pm 0.052}$
M^{3}GPT (Pre-trained)	$0.601^{\pm 0.002}$	$0.751^{\pm 0.003}$	$0.803^{\pm 0.002}$	$0.092^{\pm 0.002}$	$2.945^{\pm 0.001}$	$2.251^{\pm 0.012}$	$2.188^{\pm 0.074}$
M^{3}GPT (Fine-tuned)	$0.615^{\pm 0.003}$	$0.757^{\pm 0.004}$	$0.815^{\pm 0.003}$	$0.093^{\pm 0.002}$	$2.944^{\pm 0.002}$	$2.253^{\pm 0.023}$	$2.204^{\pm 0.058}$
M^{3}GPT(Fine-tuned only T2M)	$\mathbf{0.661}^{\pm 0.003}$	$\mathbf{0.804}^{\pm 0.004}$	$\underline{0.861}^{\pm 0.003}$	$\underline{0.076}^{\pm 0.002}$	$\mathbf{2.940}^{\pm 0.002}$	$\mathbf{2.273}^{\pm 0.026}$	$2.131^{\pm 0.032}$

Table 10: Comparison of Motion-to-Text on Motion-X [29].

Methods	RPrecision ↑			MM Dist ↓	Bleu@1 ↑	Bleu@4 ↑	Rouge ↑	CIDEr ↑	BertScore ↑
	Top1	Top2	Top3						
Real	0.681	0.824	0.881	2.897	-	-	-	-	-
TM2T [15]	0.574	0.726	0.806	2.988	30.54	**12.13**	32.52	20.16	25.37
Trained single task	0.565	0.706	0.767	3.011	31.07	10.14	31.65	22.92	28.19
M^{3}GPT (Pre-trained)	$\underline{0.627}$	$\underline{0.773}$	$\underline{0.834}$	**2.946**	$\underline{33.31}$	11.00	$\underline{34.10}$	$\underline{24.12}$	$\underline{30.96}$
M^{3}GPT (Fine-tuned)	**0.631**	**0.783**	**0.845**	$\underline{2.950}$	**34.27**	$\underline{11.50}$	**34.55**	**42.93**	**31.49**

Table 11: Comparison of Music-to-Dance on AIST++ [24] and FineDance [25].

Methods	Music-to-Dance on AIST++			Music-to-Dance on FineDance		
	FID$_k$ ↓	Div$_k$ ↑	BAS↑	FID$_k$ ↓	Div$_k$ ↑	BAS↑
Real	17.10	8.19	0.2374	-	9.73	0.2120
FACT [24]	35.35	5.94	0.2209	113.38	3.36	0.1831
Bailando [42]	28.16	**7.83**	0.2332	$\underline{82.81}$	7.74	0.2029
EDGE [48]	42.16	3.96	$\underline{0.2334}$	94.34	**8.13**	0.2116
Lodge [26]	37.09	5.58	**0.2423**	**45.56**	6.75	**0.2397**
Trained single task	75.47	5.57	0.1884	128.37	6.48	0.2036
M^{3}GPT (Pre-trained)	$\underline{27.65}$	$\underline{7.52}$	0.2105	92.35	7.67	0.2134
M^{3}GPT (Fine-tuned)	**24.34**	7.50	0.2056	86.47	$\underline{7.75}$	$\underline{0.2158}$

Table 12: Comparison of Dance-to-Music on AIST++ [24] and FineDance [25].

Methods	Dance-to-Music on AIST++					Dance-to-Music on FineDance				
	BCS ↑	CSD ↓	BHS ↑	HSD ↓	F1 ↑	BCS ↑	CSD ↓	BHS ↑	HSD ↓	F1 ↑
Foley [11]	96.4	6.9	41.0	15.0	57.5	-	-	-	-	-
CMT [10]	$\underline{97.1}$	$\underline{6.4}$	46.2	18.6	62.6	-	-	-	-	-
D2MGAN [64]	95.6	9.4	88.7	19.0	93.1	-	-	-	-	-
CDCD [65]	96.5	9.1	89.3	18.1	92.7	-	-	-	-	-
LORIS [52]	**98.6**	**6.1**	90.8	13.9	$\underline{94.5}$	-	-	-	-	-
Trained single task	93.9	9.2	93.6	12.8	92.8	**84.84**	21.61	51.35	27.13	63.97
M^{3}GPT (Pre-trained)	93.4	10.9	$\underline{93.8}$	$\underline{11.5}$	94.2	83.16	19.95	73.65	23.90	78.12
M^{3}GPT (Fine-tuned)	93.6	10.1	**94.0**	**10.6**	**94.9**	84.10	**18.36**	**74.66**	**23.45**	**78.23**

Table 13: Comparison of Motion Prediction and Motion In-between on Motion-X [29].

Methods	Motion Prediction				Motion In-between		
	FID ↓	Diversity ↑	ADE ↓	FDE ↓	FID ↓	Diversity ↑	ADE ↓
Ground Truth	0.009	2.316	-	-	0.009	2.316	-
MDM [44]	1.028	1.746	8.057	11.266	0.831	1.768	6.542
Trained single task	0.774	1.778	7.840	9.575	0.692	1.810	6.690
M^{3}GPT-pretrain	0.707	**1.874**	6.954	8.684	**0.604**	1.897	5.692
M^{3}GPT-finetune	**0.682**	1.838	**6.898**	**8.091**	0.612	**1.900**	**5.649**

Table 14: Comparison of MPJPE on Motion Prediction and Motion In-between on Motion-X [29].

Methods on Motion-X	Motion Prediction (MPJPE ↓)	Motion In-between (MPJPE ↓)
T2M-GPT [55]	80.2	63.7
MoMask [16]	67.9	55.2
MotionGPT [21]	71.3	59.9
M^{3}GPT (Instruction-tuned)	**54.2**	**51.0**

Table 15: Comparison of Music-to-Text (A2T), Text-to-Dance (T2D) and Text-to-Music (T2A) on AIST++.

Methods on AIST++	Music-to-Text		Text-to-Dance		Text-to-Music	
	Bleu@4 ↑	CIDEr ↑	R-TOP1 ↑	FID ↓	BCS ↑	BHS ↑
M^{3}GPT (Single task training for A2T)	9.24	24.55	-	-	-	-
M^{3}GPT (Single task training for T2D)	-	-	0.541	0.095	-	-
MusicLDM [2]	-	-	-	-	**74.5**	73.8
Mubert	-	-	-	-	73.3	73.0
M^{3}GPT (Instruction-tuned)	**11.95**	**28.88**	**0.588**	**0.077**	**74.5**	**74.7**

Table 16: Comparison of Text-to-Motion and Motion-to-Text with different size of T5.

Methods on Motion-X	LLM	Training time	Text-to-Motion			Motion-to-Text		
			R-TOP1 ↑	FID ↓	Div ↑	R-TOP3 ↑	Bleu4 ↑	CIDEr ↑
M^{3}GPT	T5-small (60M)	5 days	0.598	0.096	2.202	0.822	10.43	38.22
M^{3}GPT	T5-base (220M)	7 days	0.615	0.093	2.253	0.845	11.50	42.93
M^{3}GPT	T5-large (770M)	8 days	**0.619**	**0.090**	**2.256**	**0.848**	**11.64**	**43.05**

Table 17: Comparison of Music-to-Dance and Dance-to-Music with different size of T5.

Methods on AIST++	LLM	Training time	Music-to-Dance			Dance-to-Music	
			FID$_k$ ↓	DIV$_k$ ↑	BAS ↑	BCS ↑	BHS ↑
M^{3}GPT	T5-small (60M)	5 days	28.05	5.96	0.2091	89.1	91.2
M^{3}GPT	T5-base (220M)	7 days	24.34	7.50	0.2056	93.6	94.0
M^{3}GPT	T5-large (770M)	8 days	**23.26**	**7.54**	**0.2061**	**93.8**	**94.1**

Table 18: Hyper-parameter analysis of λ. Comparison of Text-to-Motion on Motion-X [29] with different values of λ. For this ablation study, M^{3}GPT is trained solely on the text-to-motion task to examine the impact of λ. This study is conducted during the pre-training stage.

Methods	RPrecision ↑			FID ↓	MM Dist ↓	Diversity →	MModality ↑
	Top1	Top2	Top3				
Real	$0.675^{\pm 0.003}$	$0.821^{\pm 0.003}$	$0.878^{\pm 0.002}$	$0.009^{\pm 0.000}$	$2.938^{\pm 0.007}$	$2.316^{\pm 0.011}$	-
λ=0.0	$0.645^{\pm 0.002}$	$0.778^{\pm 0.003}$	$0.826^{\pm 0.002}$	$0.081^{\pm 0.002}$	$2.944^{\pm 0.002}$	$2.124^{\pm 0.011}$	$2.025^{\pm 0.021}$
λ=0.1	$0.649^{\pm 0.002}$	$0.787^{\pm 0.002}$	$0.838^{\pm 0.002}$	$\mathbf{0.078}^{\pm 0.002}$	$2.944^{\pm 0.003}$	$2.128^{\pm 0.022}$	$2.039^{\pm 0.049}$
λ=0.2	$\mathbf{0.656}^{\pm 0.002}$	$\mathbf{0.795}^{\pm 0.001}$	$\mathbf{0.843}^{\pm 0.001}$	$\mathbf{0.078}^{\pm 0.000}$	$2.942^{\pm 0.001}$	$\mathbf{2.133}^{\pm 0.012}$	$\mathbf{2.046}^{\pm 0.052}$
λ=0.3	$0.629^{\pm 0.002}$	$0.716^{\pm 0.002}$	$0.784^{\pm 0.001}$	$0.095^{\pm 0.002}$	$\mathbf{2.941}^{\pm 0.001}$	$2.113^{\pm 0.037}$	$2.036^{\pm 0.041}$
λ=0.4	$0.573^{\pm 0.001}$	$0.725^{\pm 0.002}$	$0.793^{\pm 0.002}$	$0.102^{\pm 0.002}$	$2.942^{\pm 0.002}$	$2.090^{\pm 0.007}$	$2.030^{\pm 0.086}$

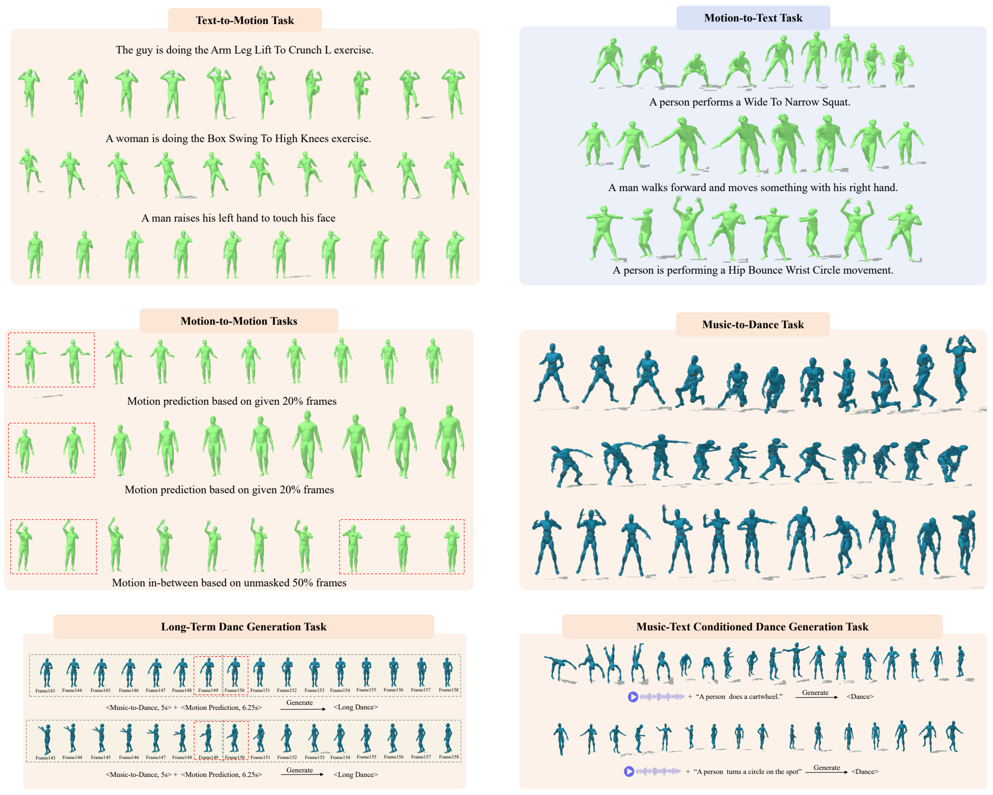
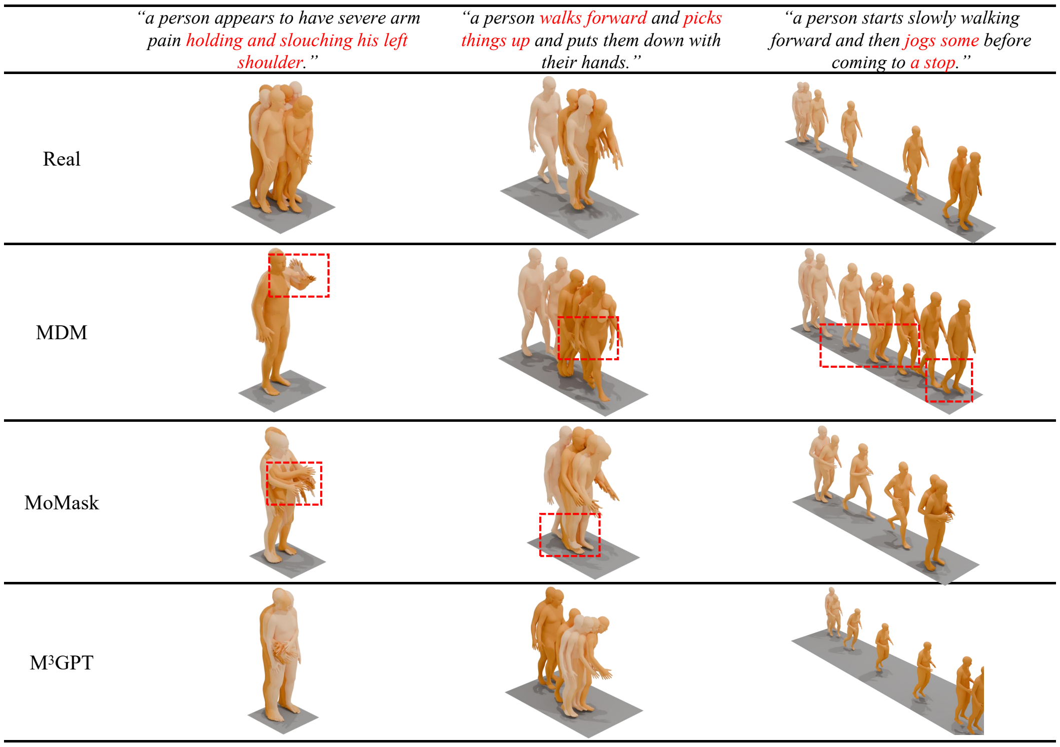
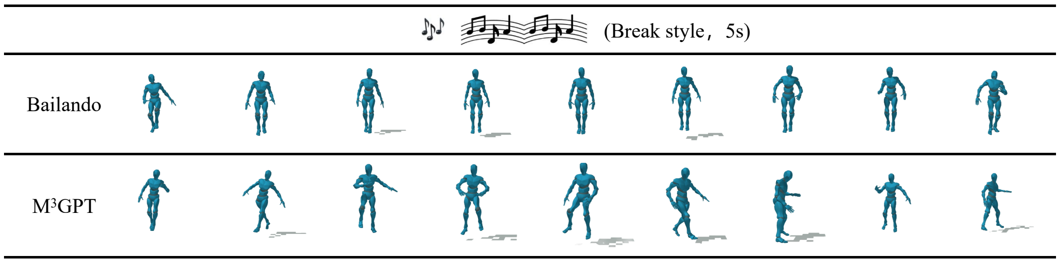

Text-to-Motion Task
The guy is doing the Arm Leg Lift To Crunch L exercise.
A woman is doing the Box Swing To High Knees exercise.
A man raises his left hand to touch his face
Motion-to-Text Task
A person performs a Wide To Narrow Squat.
A man walks forward and moves something with his right hand.
A person is performing a Hip Bounce Wrist Circle movement.
Motion-to-Motion Tasks
Motion prediction based on given 20% frames
Motion prediction based on given 20% frames
Motion in-between based on unmasked 50% frames
Music-to-Dance Task
Long-Term Danc Generation Task
Frame143 Frame144 Frame145 Frame146 Frame147 Frame148 Frame149 Frame150 Frame151 Frame152 Frame153 Frame154 Frame155 Frame156 Frame157 Frame158
<Music-to-Dance, 5s> + <Motion Prediction, 6.25s> Generate <Long Dance>
Frame143 Frame144 Frame145 Frame146 Frame147 Frame148 Frame149 Frame150 Frame151 Frame152 Frame153 Frame154 Frame155 Frame156 Frame157 Frame158
<Music-to-Dance, 5s> + <Motion Prediction, 6.25s> Generate <Long Dance>
Music-Text Conditioned Dance Generation Task
+ "A person does a cartwheel." Generate <Dance>
+ "A person turns a circle on the spot" Generate <Dance>

Figure 6: The qualitative results for different motion comprehension and generation tasks.


"a person appears to have severe arm pain holding and slouching his left shoulder."
"a person walks forward and picks things up and puts them down with their hands."
"a person starts slowly walking forward and then jogs some before coming to a stop."
Real
MDM
MoMask
M³GPT

(a) Qualitative comparison on text-to-motion task.



(b) Qualitative comparison on music-to-dance task.

Figure 7: Qualitative comparisons for text-to-motion task and music-to-dance task. (a) refers to the qualitative comparison between Real, MDM, MoMask and M3GPT on text-to-motion task. The red words and boxes highlight the misaligned motions. The results demonstrate that our M3GPT shows good text understanding for motion generation. (b) refers to the qualitative comparison between Bailando and M3GPT on music-to-dance task. The input is an 5-second-long piece of music in the Break style.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: A summary of the paper's contributions is provided in the introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See Section 5

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: Not Applicable

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.

- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: See Section 4 and Appendix for implementation details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: See `https://github.com/luomingshuang/M3GPT`.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.

- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: See Section 4 and Appendix for implementation details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [NA]

   Justification: Not Applicable.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See Section 4 and Appendix for implementation details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: See Section 5

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Not Applicable.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: See References.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: Not Applicable.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Not Applicable.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

Justification: Not Applicable.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

https://doi.org/10.52202/079017-0879