Text-to-Motion Generation with Discrete Representations and Large Language Models

Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Xi Shen, Xiaojian Shen, Hehe Fan, and Yi Yang, *Senior Member, IEEE*

Abstract—Based on Vector Quantised Variational AutoEncoder (VQ-VAE) and Transformers, we investigate a simple yet effective conditional generative framework for text-to-motion generation. First, we find that, with a few training recipes (EMA and Code Reset), a CNN-based VQ-VAE can learn high-quality discrete motion representations, which largely facilitate motion generation. Second, to better understand motion descriptions, we propose T2M-GPT+ that uses Large Language Models (LLMs) to extract text features. Further, we propose T2M-GIT+ that employs a non-autoregressive method to parallel generate discrete motion representations, thus more efficient than T2M-GPT+ while achieving comparable results. Comprehensive experiments show that our method is superior to existing methods, including competitive diffusion methods. For example, we achieve Top-1 accuracy 0.53 in R-precision and FID 0.10 on the HumanML3D dataset, which largely outperforms MotionDiffuse of Top-1 accuracy 0.49 and FID 0.63. Additionally, we conduct analyses and find that, even for the largest HumanML3D dataset, it may still limit the performance of the proposed approach. This suggests that a larger dataset could bring additional improvement to our approach. Our implementation is available on the project page: https://mael-zys.github.io/T2M-GPT/.

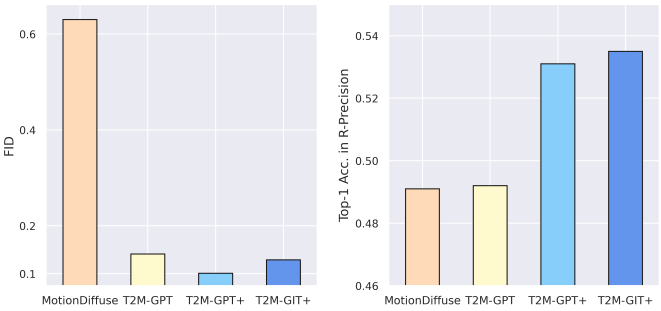
Index Terms—Text-to-Motion Generation, Vector Quantised-Variational AutoEncoder, Generative Model, Large Language Models.

1 Introduction

Generating motion from textual descriptions has recently attracted increasing attention since its potential applications in the game industry, film-making and animating robots. For instance, in the game industry, a typical way to access new motion is to conduct motion capture, which often requires professional actors and is therefore highly costly. Automatically generating human motion from textual descriptions could save time and be more economical.

Because motion and text are from different modalities, generating motion from natural language is challenging. The designed model is expected to learn a precise mapping from the language space to the motion space. To this end, many works propose to learn a joint embedding for language and motion using autoencoders [1], [2], [3] and Variational Autoencoders (VAEs) [4], [5].

MotionClip [3] aligns the motion space to CLIP [6] space. ACTOR [4] and TEMOES [5] propose transformer-based VAEs for action-to-motion and text-to-motion, respectively. These works show promising performances with simple descriptions. However, when textual descriptions become complicated, these works are limited to producing high-quality motion. Therefore, Guo *et al.* proposed two approaches [7], [8] that generate motion sequences with longer textual descriptions. However, the two approaches sometimes fail to generate high-quality motion consistent with the text. Moreover, the methods are not straightforward because they involve three stages for text-to-motion generation.



(a) FID ↓ (b) Top-1 ↑

Fig. 1: Comparison on the HumanML3D [7] dataset. T2M-GPT+ and T2M-GIT+ outperform the previous version T2M-GPT [9] and the state-of-the-art method MotionDiffuse [10] on both (a). FID and (b). Top-1 Acc. in R-Precision

Recently, diffusion-based models [11] have shown impressive results on image generation [12] and motion generation [10], [13], [14], [15]. Unlike image generation, which requires generating images in high resolution (typically 512 × 512), motion generation is relatively easy in output dimensionality (22 joints × 196 frames at most). Therefore, we try to explore classic approaches, such as VQ-VAE [16], to conduct text-driven motion generation.

In this paper, motivated by recent advances in learning the discrete representation for generation [16], [17], [18], [19], [20], [21], [22], [23], we investigate a simple and classic framework, which is based on Vector Quantized Variational Autoencoders (VQ-VAE) [16] and generative transform-

- *Jianrong Zhang, Hehe Fan, and Yi Yang are with CCAI, Zhejiang University, E-mail: jirozhang.ai@gmail.com, {hehefan, yangics}@zju.edu.cn.*
- *Yangsong Zhang is with the Ant Group, E-mail: yangsong.zhang.zys@gmail.com*
- *Xiaodong Cun is with Tencent AI Lab, E-mail: vinthony@gmail.com*
- *Xi Shen is with Intellindust, E-mail: shenxiluc@gmail.com*
- *Xiaojian Shen is with the College of Software, Jilin University, E-mail: shenxj22@mails.jlu.edu.cn*

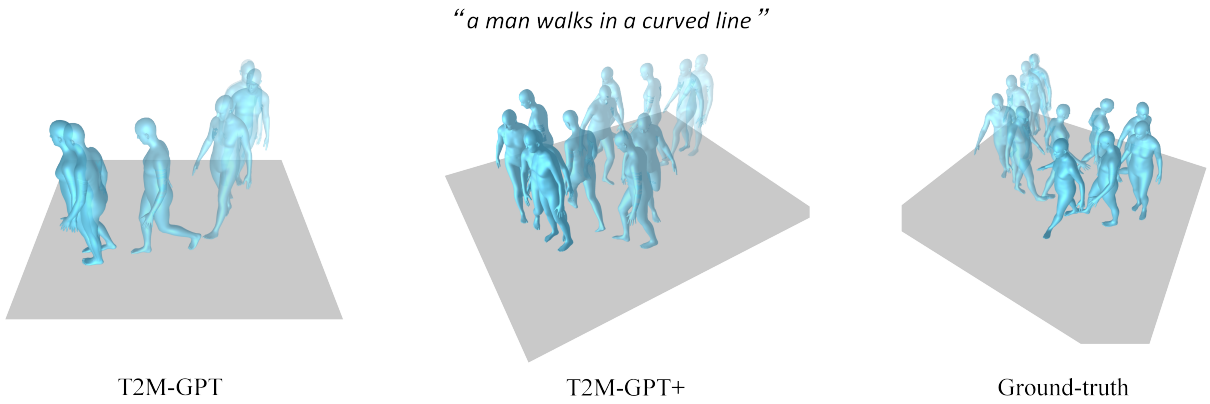


Fig. 2: **Visual results on HumanML3D [7].** Compared with the previous version T2M-GPT [9], our improved version, T2M-GPT+ is able to generate more delicate human motion. More visual results are on the project page.

ers [24], [25], [26], for text-to-motion generation.

Specifically, we propose a two-stage method for motion generation from textual descriptions. In stage 1, we use a standard 1D convolutional network to map motion sequences to discrete code indices. In stage 2, a generative transformer is learned to generate sequences of code indices from pre-trained text embedding. During this progress, we find that the naive training of VQ-VAE [16] suffers from code collapse. One effective solution is to leverage two standard recipes during the training: *EMA* and *Code Reset*. We provide a full analysis of different quantization strategies. For generative transformers, we investigate auto-regressive generation with Generative Pre-trained Transformer (GPT) and non-autoregressive generation with parallel decoding (Generative Image Transformer, GIT [26]). To the best of our knowledge, we are the first to introduce parallel decoding using GIT for human motion generation.

We propose T2M-GPT+ and show that, without any specific fine-tuning, Large Language Models (LLMs) [27] bring a significant improvement for this task. Additionally, a single-step error may lead to totally incorrect motion code prediction for the future generation. To alleviate this problem, we propose to simulate single-step errors by simply corrupting sequences during the training. For T2M-GIT+, we utilize the Mask Token Modeling [26], [28] strategy during training and generate all motion codes in parallel, which is more efficient (about 2.7 times faster than T2M-GPT+ while achieving comparable results).

Our approach can generate high-quality motion sequences that are consistent with challenging text descriptions (Figure 2). Empirically, we achieve better performances than very concurrent diffusion-based approaches MLD [15], MDM [13], and MotionDiffuse [10] on two widely used datasets: HumanML3D [7] and KIT-ML [29]. For example, on HumanML3D, T2M-GPT+ with Llama-13B [27] achieves Top-1 accuracy 0.531 in R-precision and FID 0.101, largely outperforming MotionDiffuse of Top-1 accuracy 0.491 and FID 0.630. As large models require high-quality datasets, we also explore the impact of dataset size for this task. The empirical analysis suggests that the performance of our model can potentially be improved with larger datasets.

In summary, our contributions include:

- We present a simple yet effective approach for motion generation from textual descriptions. Our approach achieves state-of-the-art performance on HumanML3D [7] and KIT-ML [29] datasets.
- We show that Large Language Models (LLMs) can significantly improve performance without any specific fine-tuning.
- We investigate both auto-regressive (T2M-GPT+) and non-autoregressive (T2M-GIT+) generation. T2M-GIT+ is to our knowledge the first approach to decode in parallel via GIT.
- We provide a detailed analysis of the impact of quantization strategies and dataset size. We show that a larger dataset might still offer a promising prospect to the community.

Visual results and the implementation are available on the project page.

2 Related Work

2.1 VQ-VAE

Vector Quantized Variational Autoencoders (VQ-VAE), which is a variant of VAE [30] and is initially proposed in [16]. VQ-VAE is composed of an AutoEncoder architecture, which aims at learning reconstruction with discrete representations. Recently, VQ-VAE achieves promising performance on generative tasks across different modalities, which includes: image synthesis [17], [18], text-to-image generation [19], speech gesture generation [20], music generation [21], [22], *etc*. The success of VQ-VAE for generation might be attributed to its decoupling of learning the discrete representation and the prior. A naive training of VQ-VAE suffers from the codebook collapse, *i.e.*, only a number of codes are activated, which importantly limited the performances of the reconstruction as well as generation. To alleviate the problem, a number of techniques can be used during training, including stop-gradient along with some losses [16] to optimize the codebook, exponential moving average (EMA) for codebook update [17], and reset inactivated codes during the training (Code Reset [17]), *etc*.

2.2 Human motion synthesis.

Research on human motion synthesis has a long history [31]. One of the most active research fields is human mo-

tion prediction, which aims at predicting the future motion sequence based on past observed motion. Approaches mainly focus on efficiently and effectively fusing spatial and temporal information to generate deterministic future motion through different models: RNN [32], [33], [34], [35], GAN [36], [37], GCN [38], Attention [39] or even simply MLP [40], [41], [42]. Some approaches aim at generating diverse motion through VAE [43], [44], [45]. In addition to synthesizing motion conditioning on past motion, another related topic is generating motion "in-betweening" that takes both past and future poses and fills motion between them [46], [47], [48], [49], [50]. Quaternet [35] considers the generation of locomotion sequences from a given trajectory for simple actions, such as: walking and running. Motion can also be generated with music to produce 3D dance motion [51], [52], [53], [54], [55], [56]. For unconstrained generations, CSGN [57] generates a long sequence altogether by transforming from a sequence of latent vectors sampled from a Gaussian process. In graphics literature, many works focus on animator control. Holden *et al.* [58] learned a convolutional autoencoder to reconstruct motion, the learned latent representation can be used to synthesize and edit motion. PFNN [59] proposes a phase-functioned neural network to perform the control task. Starke *et al.* [60] used a deep auto-regressive framework to scene interaction behaviors. Deepphase [61] proposes to reconstruct motion through periodic features, the learned periodic embedding improves motion synthesis. Recently, inspired by SinGAN [62] for image synthesis, Li *et al.* [63] proposed a generative model approach for motion synthesis from a single sequence.

2.3 Text-driven human motion generation.

Text-driven human motion generation aims at generating 3D human motion from textual descriptions. Text2Action [64] trains an RNN-based model to generate motion conditioned on a short text. Language2Pose [1] employs a curriculum learning approach to learn a joint embedding space for both text and pose. The decoder can thus take text embedding to generate motion sequences. Ghost *et al.* [2] learned two manifold representations for the upper body and the lower body movements, which shows improved performance compared to Language2Pose [1]. Similarly, MotionCLIP [3] also tends to align text and motion embedding but proposes to utilize CLIP [6] as the text encoder and employ rendered images as extra supervision. It shows the ability to generate out-of-distribution motion and enable latent code editing [65]. However, the generated motion sequences are not high-quality and without global translation. ACTOR [4] proposes a transformer-based VAE to generate motion in a non-autoregressive fashion from a pre-defined action class. TEMOS [5] extends the architecture of ACTOR [4] by introducing an additional text encoder, enabling the generation of diverse motion sequences based on text descriptions. TEMOS demonstrates its effect on KIT Motion-Language [29] with mainly short sentences and suffers from out-of-distribution descriptions [5]. TEACH [66] and SINC [67] further extend TEMOS to generate temporal and spatial motion compositions respectively from a series of motion descriptions. Recently, a large-scale dataset HumanML3D is proposed in [7]. Guo *et al.* [7] proposed to incorporate motion length prediction from text to produce motion with reasonable length. TM2T [8] considers text-to-motion and motion-to-text tasks, demonstrating that further improvement can be achieved by jointly training these tasks. As concurrent works, diffusion-based models are introduced for text-to-motion generation by MDM [13] and MotionDiffuse [10]. In this work, we show that without any sophisticated designs, the classic VQ-VAE framework with some standard training recipes could achieve better performance.

2.4 Large language models

Language model pre-training is a crucial task in natural language processing. BERT [28] introduces bidirectional transformer-based architecture with a masked language modeling strategy, which has shown great improvements. Autoregressive language model [25], [68] is another pre-training paradigm, showing significant growth in model size in recent years. GPT-3 [69] increases the model size to 175B parameters, which prompts the development of Large Language Models (LLMs). Subsequently, the emergence of several large language models facilitated the development of the community, such as OPT [70], PaLM [71], Llama [27], Vicuna [72], *etc*. A complete review of large language models is outside the scope of this work, a recent survey can be found in [73]. In this article, we show that the more powerful LLM-based text encoder can be beneficial for motion generation.

3 Method

Our goal is to generate high-quality motion that is consistent with text descriptions. The overall framework consists of two modules: Motion VQ-VAE and Transformers (T2M-GPT+ and T2M-GIT+), which are illustrated in Figure 3. The former learns a mapping between motion data and discrete code sequences, the latter generates code indices conditioned on the text description. With the decoder in Motion VQ-VAE, we are able to recover the motion from the generated code indices. In Section 3.1, we present the VQ-VAE module. We provide details of T2M-GPT+ and T2M-GIT+ in Section 3.2 and Section 3.3, respectively.

3.1 Motion VQ-VAE

VQ-VAE is proposed in [16], enabling the model to learn discrete representations for generative models. Given a motion sequence $X = [x_1, x_2, \ldots, x_T]$ with $x_t \in \mathbb{R}^d$, where T is the number of frames, $t \in [1, 2, ..., T]$ and d is the dimension of the motion, we aim to recover the motion sequence through an autoencoder and a learnable codebook C. The codebook contains K codes and $C = \{c_k\}_{k=1}^{K}$ with $c_k \in \mathbb{R}^{d_c}$, where d_c is the dimension of codes. The overview of VQ-VAE is presented in Figure 3 (a). With encoder and decoder of the autoencoder denoted by E and D, the latent feature Z can be computed as $Z = E(X)$ with $Z = [z_1, z_2, ..., z_{T/l}]$ and $z_i \in \mathbb{R}^{d_c}$, where l represents the temporal downsampling rate of the encoder E. For i-th latent feature z_i, the quantization with the codebook is to find the most similar element

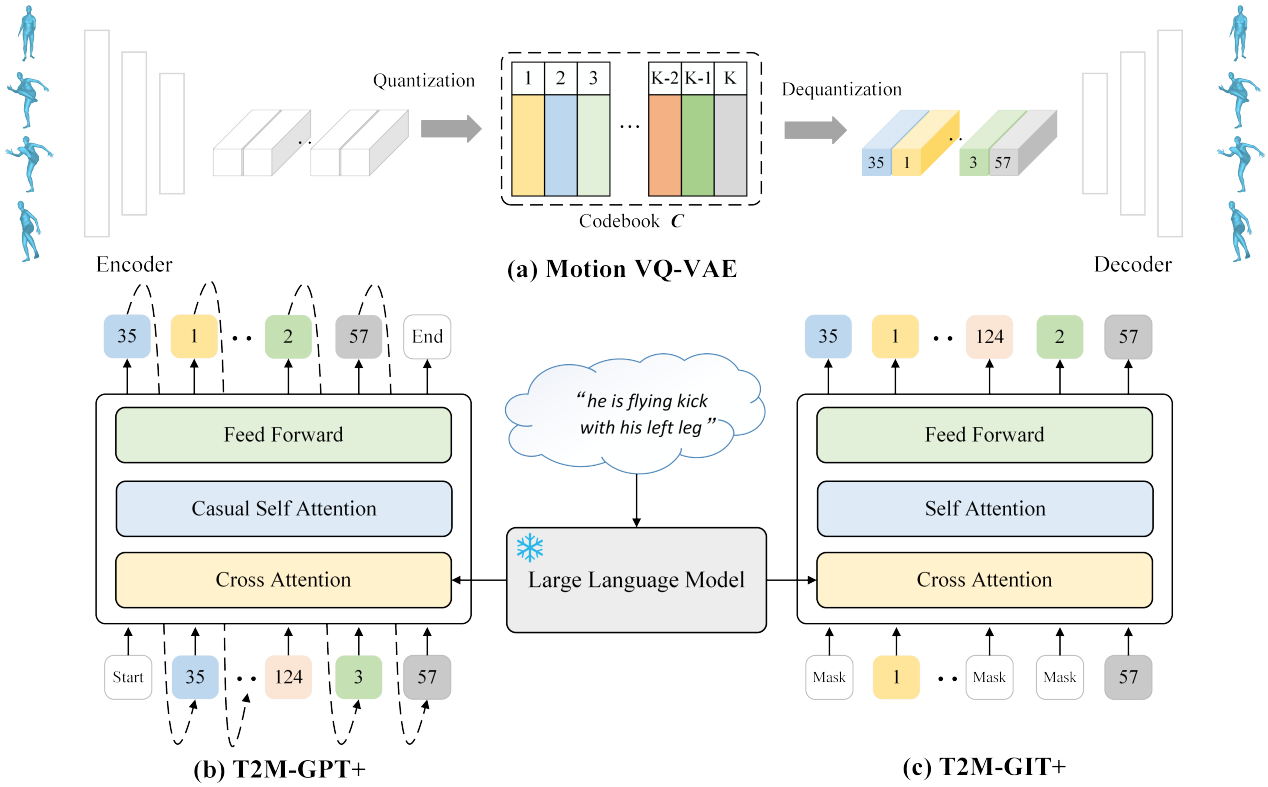


Fig. 3: **Overview of our framework for text-driven motion generation.** It includes two modules: Motion VQ-VAE and Transformers (T2M-GPT+ and T2M-GIT+). Motion VQ-VAE (a) learns a mapping between motion data and discrete code sequences. Either T2M-GPT+ (b) or T2M-GIT+ (c) can be used to generate code indices with text features extracted by Large Language Models.

in C, which is denoted by $\hat{z}_i$. The quantization process can be properly written as:

$$\hat{z}_i = \underset{c_k \in C}{\arg\min} \|z_i - c_k\|_2 \tag{1}$$

Optimization goal. To optimize VQ-VAE, the standard optimization goal [16] $\mathcal{L}_{\text{vq}}$ contains three components: a reconstruction loss $\mathcal{L}_{\text{re}}$, the embedding loss $\mathcal{L}_{embed}$ and the commitment loss $\mathcal{L}_{commit}$.

$$\mathcal{L}_{vq} = \mathcal{L}_{re} + \underbrace{||sg[Z] - \hat{Z}||_2}_{\mathcal{L}_{embed}} + \beta \underbrace{||Z - sg[\hat{Z}]||_2}_{\mathcal{L}_{commit}} \tag{2}$$

where β represents the weight of the commitment loss and sg is the stop-gradient operator. For the reconstruction loss, we find that L1 smooth loss $\mathcal{L}_1^{smooth}$ performs best, and an additional regularization on the velocity enhances the generation quality. Let X_{re} be the reconstructed motion of X (*i.e.*, $X_{\text{re}} = D(\hat{Z})$), and $V(X)$ be the velocity of X where $V = [v_1, v_2, \ldots, v_{T-1}]$ with $v_i = x_{i+1} - x_i$. Therefore, the final reconstruction loss is as follows:

$$\mathcal{L}_{re} = \mathcal{L}_1^{smooth}(X, X_{re}) + \alpha \mathcal{L}_1^{smooth}(V(X), V(X_{re})) \tag{3}$$

where α is a hyper-parameter to balance the two terms. We provide an ablation study on α as well as different reconstruction losses ($\mathcal{L}_1$, $\mathcal{L}_1^{smooth}$ and $\mathcal{L}_2$) in Section 4.4.

Quantization strategy. A naive training of VQ-VAE suffers from codebook collapse [16], [74]. Two training recipes [74] are used to improve the codebook utilization: exponential moving average *(EMA)* and codebook reset *(Code Reset)*. *EMA* makes the codebook C evolve smoothly: $C^t \leftarrow \lambda C^{t-1} + (1-\lambda)C^t$, where C^t is the codebook at iteration t and λ is the exponential moving constant. *Code Reset* finds inactivate codes during the training and reassigns them according to input data. We provide an ablation study on the quantization strategy in Section 4.4.

Architecture. We use a standard convolutional architecture composed of 1D convolution, residual block [75], and ReLU. Our VQ-VAE architecture is illustrated in Figure 4. The architecture is inspired by [18], [56]. We use convolution with stride 2 and nearest interpolation for temporal down-sampling and upsampling, respectively. The downsampling rate is thus $l = 2^L$, where L denotes the number of residual blocks. The detail of the architecture is provided in the Appendix A.

3.2 T2M-GPT+

With a learned motion VQ-VAE, a motion sequence $X = [x_1, x_2, \ldots, x_T]$ can be mapped to a sequence of indices $S = [s_1, s_2, \ldots, s_{T/l}]$ with $s_i \in [1, 2, \ldots, K]$, which are indices of the codebook. Note that two special tokens `[START]` and `[END]` are added at the beginning and end of the sequence, which indicate the start and stop of the motion, respectively. Note this design is different from [7] leveraging an extra model to predict motion length. By

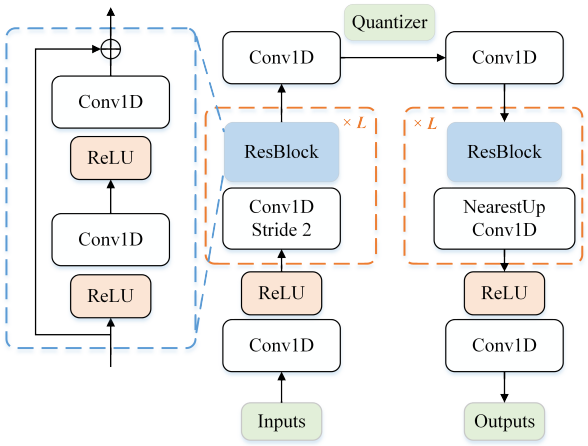


Fig. 4: **Architecture of the motion VQ-VAE.** We use a standard CNN-based architecture with 1D convolution (*Conv1D*), residual block (*ResBlock*) and ReLU activation. 'L' denotes the number of residual blocks. We use convolution with stride 2 and nearest interpolation for temporal downsampling and upsampling.

projecting S back to their corresponding codebook entries, we obtain $\hat{Z} = [\hat{z}_1, \hat{z}_2, \ldots, \hat{z}_{T/l}]$ with $\hat{z}_i = c_{s_i}$, which can be decoded to a motion X_{re} through the decoder D. Therefore, text-to-motion generation can be formulated as an autoregressive next-index prediction: given previous $i-1$ indices, *i.e.*, $S_{<i}$, and text condition c, we aim to predict the distribution of possible next indices $p(s_i|c, S_{<i})$, which can be conducted by transformer [24]. The overview of T2M-GPT+ is shown in Figure 3 (b).

Optimization goal. Denoting the likelihood of the full sequence as $p(S|c) = \prod_{i=1}^{|S|} p(s_i|c, S_{<i})$, we directly maximize the log-likelihood of the data distribution:

$$\mathcal{L}_{GPT} = \mathbb{E}_{S\sim p(S)}[-\log p(S|c)] \tag{4}$$

Large Language Models. Motivated by the great success of Large Language Models [27], [71], [72], we propose to incorporate the recent work Llama [27] as the text feature extractor. Specifically, denoting the text embedding as e with length T', which can be computed as $e = LLM(c)$. We consider integrating the text embedding into the transformer with the cross-attention operator:

$$CA = \text{Softmax}\left(\frac{\mathcal{Q}(m)\cdot\mathcal{K}(e)^\top}{\sqrt{d_k}}\right)\cdot\mathcal{V}(e) \tag{5}$$

where m denotes the embedding of motion code indices, $\mathcal{Q}(m) \in \mathbb{R}^{T\times d_k}$ is the query, $\mathcal{K}(e) \in \mathbb{R}^{T'\times d_k}$ and $\mathcal{V}(e) \in \mathbb{R}^{T'\times d_k}$ are key and value, respectively.

Causal Self-attention. We also apply the causal self-attention [25] in T2M-GPT+. Precisely, the output of the causal self-attention is calculated as follows:

$$CSA = \text{Softmax}\left(\frac{\mathcal{Q}(m)\mathcal{K}(m)^\top \times mask}{\sqrt{d_k}}\right)\cdot\mathcal{V}(m) \tag{6}$$

where $\mathcal{K}(m) \in \mathbb{R}^{T\times d_k}$ and $\mathcal{V}(m) \in \mathbb{R}^{T\times d_k}$ are key and value respectively, while $mask$ is the causal mask with $mask_{i,j} = -\infty \times \mathbf{1}(i < j) + \mathbf{1}(i \geq j)$, where $\mathbf{1}(\cdot)$ is the indicator function. This causal mask ensures that future information is not allowed to attend the calculation of current tokens. For inference, with the `[START]` token, we generate indices in an autoregressive fashion, and the generation process will be stopped if the model predicts the `[END]` token. Note that we are able to generate diverse motions by sampling from the predicted distributions given by the transformer.

Corrupted sequences for the training-testing discrepancy. There is a discrepancy between training and testing. For training, $i-1$ correct indices are used to predict the next index. While for inference, there is no guarantee that indices serving as conditions are correct. To alleviate this problem, we adopt a simple data augmentation strategy to simulate errors: we replace $\tau \times 100\%$ ground-truth code indices with random ones during training. τ can be a hyper-parameter or randomly sampled from $\tau \in \mathcal{U}[0, 1]$. We provide an ablation study on this strategy in Section 4.4

3.3 T2M-GIT+

T2M-GPT+ generates motion sequences with the next-index prediction based on previously generated tokens, which is time-consuming and error-sensitive to previous results. We explore a more efficient method called T2M-GIT+ (see Figure 3 (c)), which leverages parallel decoding. It introduces the Masked Token Modeling (MTM) [26] strategy to generate index tokens in parallel. Specifically, with a sequence of indices $S = [s_1, s_2, \ldots, s_{T/l}]$ available, a percentage of $\gamma \times 100\%$ tokens are randomly sampled and replaced by the `[MASK]` token leading to $\tilde{S} = [\tilde{s}_1, \tilde{s}_2, \ldots, \tilde{s}_{T/l}]$ during training, where $\gamma \in (0, 1]$ and $\tilde{s}_i \in [1, 2, \ldots, K, \texttt{[MASK]}]$. Then, we formulate the model training as a non-autoregressive masked code indices prediction: given masked index tokens $\tilde{S}$ and the condition c, we aim to predict the distribution of $p(S|c, \tilde{S})$.

Optimization goal. Similar to T2M-GPT+, we denote the likelihood of the sequence as $p(S|c, \tilde{S}) = \prod_{i=1}^{|S|} p(s_i|c, \tilde{S})$, and maximize the log-likelihood of the distribution:

$$\mathcal{L}_{GIT} = \mathbb{E}_{S\sim p(S)}[-\log p(S|c, \tilde{S})] \tag{7}$$

When generating the corresponding motion index, we follow Section 3.2 to use LLMs and employ cross-attention to attend to the text embedding. Different from T2M-GPT+, we use a standard self-attention operator rather than causal self-attention.

Inference. In terms of inference, we apply iterative decoding [26] to generate motion index sequences. For each iteration, we sample $\frac{T}{l} - M$ most confident tokens based on the predicted probability of each token, and then mask the remaining tokens for the next iteration. M can be calculated according to the scheduling function $f(\cdot)$ with $\lceil f(\frac{n}{N}) \times \frac{T}{l} \rceil$, where $f(\cdot)$ denotes the function of cosine decay and N is the total iteration time step. Therefore, a motion sequence can be generated in N steps where $N < \frac{T}{l}$. We provide an ablation study of the number of iterations N in Section 4.4.

4 Experiments

In this section, we present our experimental results. In Section 4.1, we introduce standard datasets as well as

TABLE 1: **Comparison with the state-of-the-art methods on HumanML3D [7] test set.** We compute standard metrics following Guo *et al.* [7]. For each metric, we repeat the evaluation 20 times and report the average with 95% confidence interval. Red and Blue indicate the best and the second best result. The § reports results using ground-truth motion length.

Methods	R-Precision ↑			FID ↓	MM-Dist ↓	Diversity ↑	MModality ↑
	Top-1	Top-2	Top-3				
Real motion	$0.511^{\pm.003}$	$0.703^{\pm.003}$	$0.797^{\pm.002}$	$0.002^{\pm.000}$	$2.974^{\pm.008}$	$9.503^{\pm.065}$	-
Our VQ-VAE (Recons.)	$0.501^{\pm.002}$	$0.692^{\pm.002}$	$0.785^{\pm.002}$	$0.070^{\pm.001}$	$3.072^{\pm.009}$	$9.593^{\pm.079}$	-
Seq2Seq [76]	$0.180^{\pm.002}$	$0.300^{\pm.002}$	$0.396^{\pm.002}$	$11.75^{\pm.035}$	$5.529^{\pm.007}$	$6.223^{\pm.061}$	-
Language2Pose [1]	$0.246^{\pm.002}$	$0.387^{\pm.002}$	$0.486^{\pm.002}$	$11.02^{\pm.046}$	$5.296^{\pm.008}$	$7.676^{\pm.058}$	-
Text2Gesture [77]	$0.165^{\pm.001}$	$0.267^{\pm.002}$	$0.345^{\pm.002}$	$5.012^{\pm.030}$	$6.030^{\pm.008}$	$6.409^{\pm.071}$	-
Hier [2]	$0.301^{\pm.002}$	$0.425^{\pm.002}$	$0.552^{\pm.004}$	$6.532^{\pm.024}$	$5.012^{\pm.018}$	$8.332^{\pm.042}$	-
MoCoGAN [78]	$0.037^{\pm.000}$	$0.072^{\pm.001}$	$0.106^{\pm.001}$	$94.41^{\pm.021}$	$9.643^{\pm.006}$	$0.462^{\pm.008}$	$0.019^{\pm.000}$
Dance2Music [51]	$0.033^{\pm.000}$	$0.065^{\pm.001}$	$0.097^{\pm.001}$	$66.98^{\pm.016}$	$8.116^{\pm.006}$	$0.725^{\pm.011}$	$0.043^{\pm.001}$
TEMOS [5]§	$0.424^{\pm.002}$	$0.612^{\pm.002}$	$0.722^{\pm.002}$	$3.734^{\pm.028}$	$3.703^{\pm.008}$	$8.973^{\pm.071}$	$0.368^{\pm.018}$
TM2T [8]	$0.424^{\pm.003}$	$0.618^{\pm.003}$	$0.729^{\pm.002}$	$1.501^{\pm.017}$	$3.467^{\pm.011}$	$8.589^{\pm.076}$	$2.424^{\pm.093}$
Guo *et al.* [7]	$0.455^{\pm.003}$	$0.636^{\pm.003}$	$0.736^{\pm.002}$	$1.087^{\pm.021}$	$3.347^{\pm.008}$	$9.175^{\pm.083}$	$2.219^{\pm.074}$
MLD [15]§	$0.481^{\pm.003}$	$0.673^{\pm.003}$	$0.772^{\pm.002}$	$0.473^{\pm.013}$	$3.196^{\pm.010}$	$9.724^{\pm.082}$	$2.413^{\pm.079}$
Zhang *et al.* [79]	-	-	-	0.567	3.775	9.006	-
MDM [13]§	-	-	$0.611^{\pm.007}$	$0.544^{\pm.044}$	$5.566^{\pm.027}$	$9.559^{\pm.086}$	$2.799^{\pm.072}$
MotionDiffuse [10]§	$0.491^{\pm.001}$	$0.681^{\pm.001}$	$0.782^{\pm.001}$	$0.630^{\pm.001}$	$3.113^{\pm.001}$	$9.410^{\pm.049}$	$1.553^{\pm.042}$
Jiang *et al.* [80]	$0.492^{\pm.003}$	$0.681^{\pm.003}$	$0.778^{\pm.002}$	$0.232^{\pm.008}$	$3.096^{\pm.008}$	$9.528^{\pm.071}$	$2.008^{\pm.084}$
T2M-GPT (CLIP) [9]	$0.492^{\pm.003}$	$0.679^{\pm.002}$	$0.775^{\pm.002}$	$0.141^{\pm.005}$	$3.121^{\pm.009}$	$9.722^{\pm.082}$	$1.831^{\pm.048}$
T2M-GPT+ (Llama-7B)	$0.530^{\pm.003}$	$0.724^{\pm.002}$	$0.814^{\pm.003}$	$0.104^{\pm.006}$	$2.878^{\pm.029}$	$9.598^{\pm.127}$	$1.788^{\pm.076}$
T2M-GPT+ (Llama-13B)	$0.531^{\pm.004}$	$0.717^{\pm.005}$	$0.811^{\pm.003}$	$0.101^{\pm.009}$	$2.893^{\pm.006}$	$9.646^{\pm.072}$	$1.759^{\pm.079}$
T2M-GIT+ (Llama-7B)§	$0.527^{\pm.001}$	$0.719^{\pm.001}$	$0.814^{\pm.003}$	$0.135^{\pm.008}$	$2.881^{\pm.015}$	$9.748^{\pm.022}$	$0.452^{\pm.015}$
T2M-GIT+ (Llama-13B)§	$0.535^{\pm.004}$	$0.729^{\pm.004}$	$0.821^{\pm.003}$	$0.129^{\pm.008}$	$2.849^{\pm.010}$	$9.787^{\pm.022}$	$0.404^{\pm.03}$

evaluation metrics. Implementation details are provided in Section 4.2 We compare our results to state-of-the-art approaches in Section 4.3. Finally, we provide analysis and discussion in Section 4.4.

4.1 Datasets and evaluation metrics

We conduct experiments on two standard datasets for text-driven motion generations: HumanML3D [7] and KIT Motion-Language (KIT-ML) [29]. Both datasets are commonly used in the community. We follow the evaluation protocol proposed in [7].

HumanML3D. HumanML3D [7] is currently the largest 3D human motion dataset with textual descriptions. The dataset contains 14,616 human motions and 44,970 text descriptions. The entire textual descriptions are composed of 5,371 distinct words. The motion sequences are originally from AMASS [81] and HumanAct12 [82] but with specific pre-processing: motions are scaled to 20 FPS; those that are longer than 10 seconds are randomly cropped to 10-second ones; they are then re-targeted to a default human skeletal template and properly rotated to face the Z+ direction. Each motion is paired with at least 3 precise textual descriptions. The average length of descriptions is approximately 12. According to [7], the dataset is split into training, validation, and test sets with proportions of 80%, 5%, and 15%, respectively. We select the best FID model on the validation set and report its performance on the test set.

KIT Motion-Language (KIT-ML). KIT-ML [29] contains 3,911 human motion sequences and 6,278 textual annotations. The total vocabulary size, which counts unique words while disregarding capitalization and punctuation, is 1,623. Motion sequences are selected from KIT [83] and CMU [84] datasets but downsampled into 12.5 frame-per-second (FPS). Each motion sequence is described in ranging from 1 to 4 sentences. The average length of descriptions is approximately 8. Following [7], [8], the dataset is split into training, validation, and test sets with proportions of 80%, 5%, and 15%, respectively. We select the model that achieves the best FID on the validation set and report its performance on the test set.

Evaluation metrics. Following [7], global representations of the motion and text description are first extracted with the pre-trained network in [7], and then measured by the following five metrics:

- *R-Precision.* Given one motion sequence and 32 text descriptions (1 ground-truth and 31 randomly selected mismatched descriptions), we rank the Euclidean distances between the motion and text embeddings. Top-1, Top-2 and Top-3 accuracy of motion-to-text retrieval are reported.
- *Frechet Inception Distance (FID).* We calculate the distribution distance between the generated and real motion using FID [85] based on the extracted motion features. FID is obtained by:

$$\text{FID} = \|\mu_{gt} - \mu_{pred}\|^2 - \text{Tr}(\Sigma_{gt} + \Sigma_{pred} - 2(\Sigma_{gt}\Sigma_{pred})^{\frac{1}{2}}) \tag{8}$$

  where μ_{gt} and μ_{pred} are mean of f_{gt} and f_{pred}. Σ is the covariance matrix and *Tr* denotes the trace of the matrix.
- *Multimodal Distance (MM-Dist).* MM-Dist measures the

TABLE 2: **Comparison with the state-of-the-art methods on KIT-ML [29] test set.** We compute standard metrics following Guo *et al.* [7]. For each metric, we repeat the evaluation 20 times and report the average with 95% confidence interval. Red and Blue indicate the best and the second best result. The § reports results using ground-truth motion length.

Methods	R-Precision ↑			FID ↓	MM-Dist ↓	Diversity ↑	MModality ↑
	Top-1	Top-2	Top-3				
Real motion	$0.424^{\pm.005}$	$0.649^{\pm.006}$	$0.779^{\pm.006}$	$0.031^{\pm.004}$	$2.788^{\pm.012}$	$11.08^{\pm.097}$	-
Our VQ-VAE (Recons.)	$0.399^{\pm.005}$	$0.614^{\pm.005}$	$0.740^{\pm.006}$	$0.472^{\pm.011}$	$2.986^{\pm.027}$	$10.994^{\pm.120}$	-
Seq2Seq [76]	$0.103^{\pm.003}$	$0.178^{\pm.005}$	$0.241^{\pm.006}$	$24.86^{\pm.348}$	$7.960^{\pm.031}$	$6.744^{\pm.106}$	-
Language2Pose [1]	$0.221^{\pm.005}$	$0.373^{\pm.004}$	$0.483^{\pm.005}$	$6.545^{\pm.072}$	$5.147^{\pm.030}$	$9.073^{\pm.100}$	-
Text2Gesture [77]	$0.156^{\pm.004}$	$0.255^{\pm.004}$	$0.338^{\pm.005}$	$12.12^{\pm.183}$	$6.964^{\pm.029}$	$9.334^{\pm.079}$	-
Hier [2]	$0.255^{\pm.006}$	$0.432^{\pm.007}$	$0.531^{\pm.007}$	$5.203^{\pm.107}$	$4.986^{\pm.027}$	$9.563^{\pm.072}$	-
MoCoGAN [78]	$0.022^{\pm.002}$	$0.042^{\pm.003}$	$0.063^{\pm.003}$	$82.69^{\pm.242}$	$10.47^{\pm.012}$	$3.091^{\pm.043}$	$0.250^{\pm.009}$
Dance2Music [51]	$0.031^{\pm.002}$	$0.058^{\pm.002}$	$0.086^{\pm.003}$	$115.4^{\pm.240}$	$10.40^{\pm.016}$	$0.241^{\pm.004}$	$0.062^{\pm.002}$
TEMOS [5], [15]§	$0.353^{\pm.002}$	$0.561^{\pm.002}$	$0.687^{\pm.002}$	$3.717^{\pm.028}$	$3.417^{\pm.008}$	$10.84^{\pm.071}$	$0.532^{\pm.018}$
TM2T [8]	$0.280^{\pm.006}$	$0.463^{\pm.007}$	$0.587^{\pm.005}$	$3.599^{\pm.051}$	$4.591^{\pm.019}$	$9.473^{\pm.100}$	$3.292^{\pm.034}$
Guo *et al.* [7]	$0.361^{\pm.006}$	$0.559^{\pm.007}$	$0.681^{\pm.007}$	$3.022^{\pm.107}$	$3.488^{\pm.028}$	$10.72^{\pm.145}$	$2.052^{\pm.107}$
MLD [15]§	$0.390^{\pm.008}$	$0.609^{\pm.008}$	$0.734^{\pm.007}$	$0.404^{\pm.027}$	$3.204^{\pm.027}$	$10.80^{\pm.117}$	$2.192^{\pm.071}$
Zhang *et al.* [79]	-	-	-	0.597	3.394	10.54	-
MDM [13]§	-	-	$0.396^{\pm.004}$	$0.497^{\pm.021}$	$9.191^{\pm.022}$	$10.847^{\pm.109}$	$1.907^{\pm.214}$
MotionDiffuse [10]§	$0.417^{\pm.004}$	$0.621^{\pm.004}$	$0.739^{\pm.004}$	$1.954^{\pm.062}$	$2.958^{\pm.005}$	$11.10^{\pm.143}$	$0.730^{\pm.013}$
T2M-GPT (CLIP) [9]	$0.416^{\pm.006}$	$0.627^{\pm.006}$	$0.745^{\pm.006}$	$0.514^{\pm.029}$	$3.007^{\pm.023}$	$10.921^{\pm.108}$	$1.570^{\pm.039}$
T2M-GPT+ (Llama-13B)	$0.406^{\pm.008}$	$0.617^{\pm.008}$	$0.749^{\pm.007}$	$0.574^{\pm.029}$	$3.064^{\pm.018}$	$10.704^{\pm.108}$	$1.716^{\pm.060}$
T2M-GIT+ (Llama-13B)§	$0.417^{\pm.007}$	$0.638^{\pm.008}$	$0.753^{\pm.004}$	$0.571^{\pm.040}$	$2.932^{\pm.026}$	$10.973^{\pm.127}$	$0.751^{\pm.025}$

TABLE 3: **Impact of iteration number and inference speed on HumanML3D [7] test set.** We report R-Precision, FID, MM-Dist, and one-time evaluation speed for all the settings, and we use the same text encoder (Llama-13B). Red and Blue indicate the best and the second best result.

Methods	Timesteps	R-Precision ↑			FID ↓	MM-Dist ↓	Hours
		Top-1	Top-2	Top-3			
T2M-GIT+	$N=6$	$0.527^{\pm.003}$	$0.719^{\pm.003}$	$0.812^{\pm.002}$	$0.137^{\pm.006}$	$2.878^{\pm.007}$	3.6
	$N=10$	$0.530^{\pm.003}$	$0.722^{\pm.003}$	$0.818^{\pm.002}$	$0.136^{\pm.004}$	$2.862^{\pm.009}$	5.6
	$N=14$	$0.535^{\pm.004}$	$0.729^{\pm.004}$	$0.821^{\pm.003}$	$0.129^{\pm.008}$	$2.849^{\pm.010}$	8.4
	$N=18$	$0.526^{\pm.002}$	$0.719^{\pm.003}$	$0.814^{\pm.002}$	$0.134^{\pm.005}$	$2.874^{\pm.009}$	10.1
T2M-GPT+	T/l	$0.531^{\pm.004}$	$0.717^{\pm.005}$	$0.811^{\pm.003}$	$0.101^{\pm.009}$	$2.893^{\pm.006}$	22.2

distance between the text embedding and the generated motion feature. Given N_m randomly generated samples, the MM-Dist measures the feature-level distance between the motion and the text. Precisely, it computes the average Euclidean distances between each text feature and the generated motion feature as follows:

$$\text{MM-Dist} = \frac{1}{N_m}\sum_{i=1}^{N_m}\|f_{pred,i} - f_{text,i}\| \tag{9}$$

where $f_{text,i}$ and $f_{pred,i}$ are features of the i-th text-motion pair.

- *Diversity.* Diversity measures the variance of the whole motion sequences across the dataset. We randomly sample S_{dis} pairs of motion and each pair of motion features is denoted by $f_{pred,i}$ and $f'_{pred,i}$. The diversity can be calculated by:

$$\text{Diversity} = \frac{1}{S_{dis}}\sum_{i=1}^{S_{dis}}||f_{pred,i} - f'_{pred,i}|| \tag{10}$$

In our experiments, we set S_{dis} to 300 as [7].

- *Multimodality (MModality).* MModality measures the diversity of human motion generated from the same text description. Precisely, for the i-th text description, we generate motion N_t = 30 times and then sample two subsets containing 10 motions. We denote features of the j-th pair of the i-th text description by $(f_{pred,i,j}, f'_{pred,i,j})$. The MModality is defined as follows:

$$\text{MModality} = \frac{1}{10N_t}\sum_{i=1}^{N_t}\sum_{j=1}^{10}\|f_{pred,i,j} - f'_{pred,i,j}\| \tag{11}$$

4.2 Implementation details

We use the same motion representations as [7]. Each pose is represented by $(\dot{r}^a, \dot{r}^x, \dot{r}^z, r^y, j^p, j^v, j^r, c^f)$, where $\dot{r}^a \in \mathbb{R}$ is the global root angular velocity; $\dot{r}^x \in \mathbb{R}, \dot{r}^z \in \mathbb{R}$ are the global root velocity in the X-Z plan; $j^p \in \mathbb{R}^{3j}, j^v \in \mathbb{R}^{3j}, j^r \in \mathbb{R}^{6j}$ are the local pose positions, velocity and rotation with j the number of joints; $c^f \in \mathbb{R}^4$ is the foot contact features calculated by the heel and toe joint velocity.

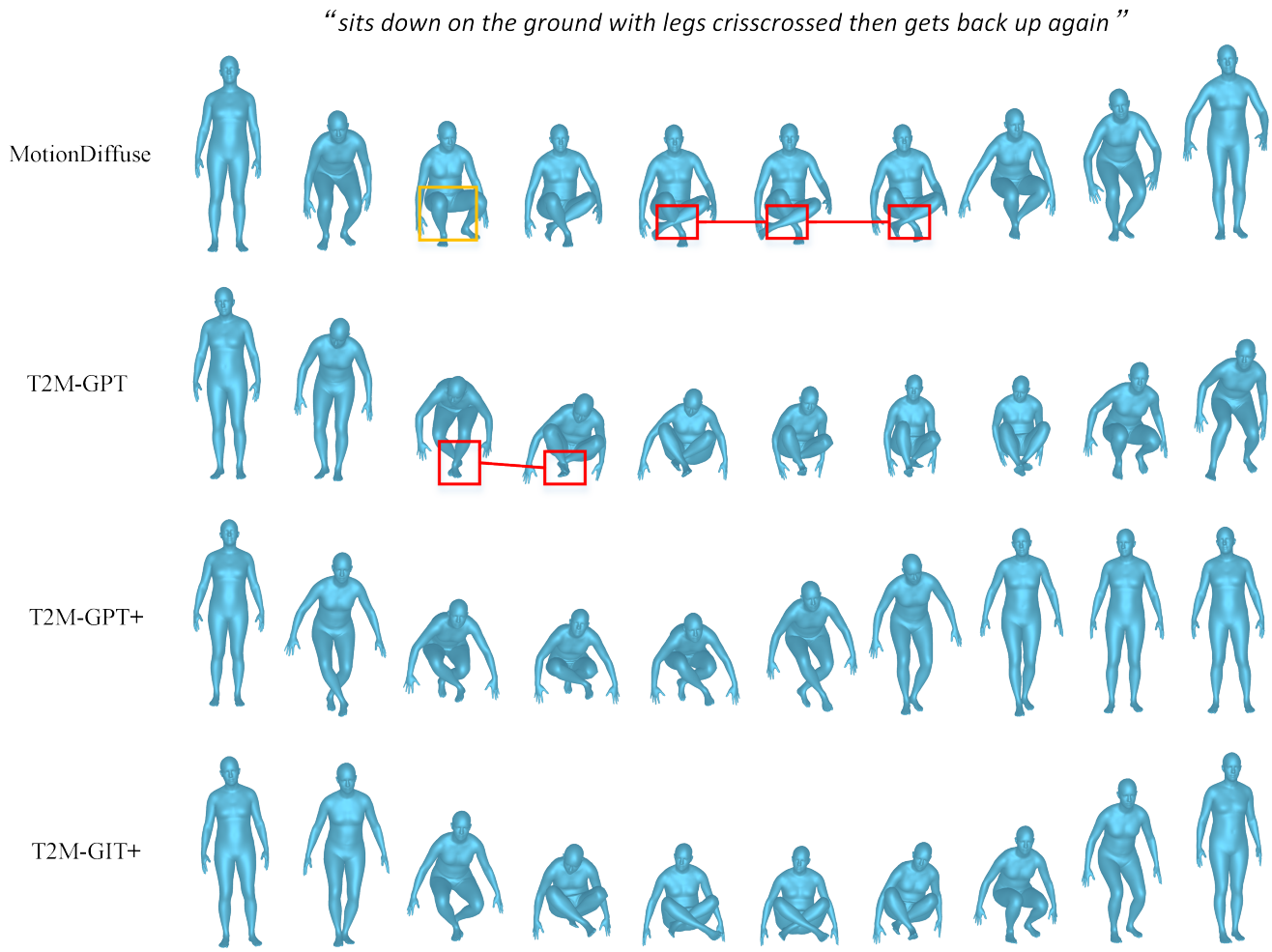


Fig. 5: **Visual results on HumanML3D [7] dataset.** We compare our generations with MotionDiffuse [10], and T2M-GPT [9]. Distorted motions (red) and overhang problem (yellow) are highlighted. More visual results can be found on the project page.

For Motion VQ-VAE, the codebook size is set to 512×512. The downsampling rate l is 4. We provide an ablation on the number of codes in Section 4.4. For both HumanML3D [7] and KIT-ML [29], the motion sequences are cropped to $T = 64$ for training. We use AdamW [86] optimizer with $[\beta_1, \beta_2] = [0.9, 0.99]$, batch size of 256, and exponential moving constant $\lambda = 0.99$. We train the first 200K iterations with a learning rate of 2e-4, and 100K with a learning rate of 1e-5. β and α in $\mathcal{L}_{\text{vq}}$ and $\mathcal{L}_{\text{re}}$ are set to 0.02 and 0.5. Following [7], the dataset HumanML3D and KIT-ML are extracted into motion features with dimensions 263 and 251 respectively, which correspond to local joint position, velocity, and rotations in root space as well as global translation and rotations. These features are computed from 22 and 21 joints of SMPL [87].

For the T2M-GPT+ and T2M-GIT+, we employ 18 transformer [24] layers with a dimension of 1,024 and 16 heads. We experiment with both Llama-13B and Llama-7B [27] models as the text feature extractor. Following Guo *et al*. [7], the maximum length of the motion is 196 on both datasets, and the minimum lengths are 40 and 24 for HumanML3D [7] and KIT-ML [29] respectively. The maximum length of the code index sequence is $T' = 50$. We train two extra `[START]` and `[END]` tokens as signals to start and stop index generation in T2M-GPT+. The transformer is optimized using AdamW [86] with $[\beta_1, \beta_2] = [0.5, 0.99]$ and batch size 256. The initialized learning rate is set to 2e-4 for 150K iterations and decayed to 1e-5 for another 150K iterations. Training Motion VQ-VAE, and transformers(T2M-GPT+ and T2M-GIT+) take about 12 hours and 3.8 days respectively on a single NVIDIA A100 GPU.

4.3 Comparison to state-of-the-art approaches

Quantitative results. We show the comparison results in Table 1 and Table 2 on HumanML3D [7] test set and KIT-ML [29] test set. On both datasets, our reconstruction with VQ-VAE reaches close performances to real motion. This indicates that our VQ-VAE effectively learns high-quality discrete representations. For the generation, our approach significantly outperforms the state-of-the-art method MotionDiffuse [10]. Compared to T2M-GPT [9], simply using LLMs as the text encoder improves the performance on text-motion consistency (R-Precision and MM-Dist) a lot. T2M-GIT+ achieves comparable performance as T2M-GPT+ but with a much faster decoding. A more detailed analysis of inference speed is provided in Section 4.4. Note that we

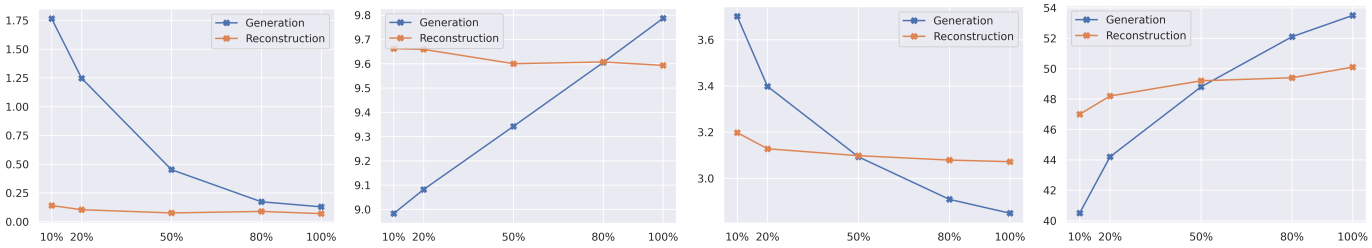
Fig. 6: **Visual results of T2M-GIT+ generation process on HumanML3D [7] dataset.** We compare the visual results of different generation steps of T2M-GIT+.

implicitly learn the motion length through an additional `[END]` token for T2M-GPT+, and evaluate the T2M-GIT+ model with the ground-truth motion length.

Qualitative comparison. Figure 5 shows visual results on HumanML3D [7]. We compare our generations with the current state-of-the-art models: MotionDiffuse [10] and T2M-GPT [9]. One can figure out that T2M-GPT+ and T2M-GIT+ generate human motions with better quality than competitive approaches. We highlight unrealistic motions (one leg passes through another leg) in red that are generated by MotionDiffuse [10] and T2M-GPT [9]. Some consecutive motion frames generated by MotionDiffuse [10] are suspended in the air (highlighted in yellow).

4.4 Analysis and discussion

Analysis of iteration number N and inference speed. In Table 3, we study the iteration number N as well as the inference time of both T2M-GIT+ and T2M-GPT+. As shown in Table 3, $N = 14$ achieves the best performance and fewer iterations would lead to similar performances. In terms of inference time, it can be seen that T2M-GIT+ is significantly more efficient. For example, T2M-GIT+ with $N = 14$ is



(a) FID (b) Diversity (c) MM-Dist (d) Top-1 accuracy

Fig. 7: **Impact of dataset size on HumanML3D [7].** We train our motion VQ-VAE (*Reconstruction*) and T2M-GPT+ (*Generation*) on the subsets of HumanML3D [7]. The subsets are composed of 10%, 20%, 50%, 80%, and 100% of the training set. All the models are evaluated on the entire test set. We report FID, Diversity, MM-Dist, and Top-1 accuracy for all the models. Results suggest that our model might benefit from more training data.

TABLE 4: **Analysis of VQ-VAE quantizers on HumanML3D [7] test set.** For all the quantizers, we use the same architectures described in Section 4.1. We report FID and Top-1 for reconstruction. For each metric, we repeat the evaluation 20 times and report the average with 95% confidence interval.

Quantizer		Reconstruction	
Code Reset	EMA	FID ↓	Top-1 ↑
		$0.492^{\pm.004}$	$0.436^{\pm.003}$
	✓	$0.097^{\pm.001}$	$0.499^{\pm.002}$
✓		$0.102^{\pm.001}$	$0.494^{\pm.003}$
✓	✓	$\mathbf{0.070}^{\pm.001}$	$\mathbf{0.501}^{\pm.002}$

TABLE 5: Ablation of losses for VQ-VAE on HumanML3D [7] test set. We report FID and Top1 metrics for the models trained 300K iterations.

$\mathcal{L}_{cons}$	α	Reconstruction	
		FID ↓	Top-1 (%)
L1	0	$0.095^{\pm.001}$	$0.493^{\pm.002}$
L1	0.5	$0.144^{\pm.001}$	$0.495^{\pm.003}$
L1	1	$0.160^{\pm.001}$	$0.496^{\pm.003}$
L1-Smooth	0	$0.112^{\pm.001}$	$0.496^{\pm.003}$
L1-Smooth	0.5	$\mathbf{0.070}^{\pm.001}$	$\mathbf{0.501}^{\pm.002}$
L1-Smooth	1	$0.128^{\pm.001}$	$0.499^{\pm.003}$
L2	0	$0.321^{\pm.002}$	$0.478^{\pm.003}$
L2	0.5	$0.292^{\pm.002}$	$0.483^{\pm.002}$
L2	1	$0.213^{\pm.002}$	$0.490^{\pm.003}$

about 2.7 times faster than T2M-GPT+. Visual results of different N are provided in Figure 6, we figure out that in the first few steps, the motion is not realistic (especially the walking part), and then after some iterations, T2M-GIT+ generates motion with better quality.

Impact of dataset size. We further analyze the impact of dataset size. To understand whether the largest dataset HumanML3D [7] contains enough data for motion generation, we train our motion VQ-VAE and T2M-GPT+ on different subsets of the training data, which consists of 10%, 20%, 50%, 80%, and 100% of the training data respectively. The trained models are evaluated on the entire test set. The results are illustrated in Figure 7. We evaluate reconstruction for our motion VQ-VAE and generation for our T2M-GPT+ using four metrics: FID, Diversity, MM-Dist, and Top-1 accuracies. Several insights can be figured out: *i)* metric for motion quality (FID) and metric for motion-text consistency (MM-Dist and Top-1) should be considered at the same time. With only 10% data, the motion might be of good quality, however, the model is not able to generate a correct motion that corresponds to the text description; *ii)* the performances become better with more training data. This trend suggests that additional training data could bring non-negligible improvement to both reconstruction and generation.

TABLE 6: Study on the number of code in codebook on HumanML3D [7] test set.

Num. code	Reconstruction	
	FID ↓	Top-1 (%)
256	$0.145^{\pm.001}$	$0.497^{\pm.002}$
512	$\mathbf{0.070}^{\pm.001}$	$\mathbf{0.501}^{\pm.002}$
1024	$0.090^{\pm.001}$	$0.498^{\pm.003}$

Quantization strategies. We investigate the impact of different quantization strategies presented in Section 3.1. The results are illustrated in Table 4. We notice that naive VQ-VAE training is not able to reconstruct nor generate high-quality motion. However, training with *EMA* or *Code Reset* can importantly boost the performances for both reconstruction and generation.

Impact of the reconstruction loss in motion VQ-VAE. We study the effect of the reconstruction loss ($\mathcal{L}_{re}$ in Equation 3) and the hyper-parameter α (Equation 3). The results are presented in Table 5. We find that L1 Smooth achieves the best performance on reconstruction, and the performance of L1 loss is close to L1 Smooth loss. For the hyper-parameter α, we find that $\alpha = 0.5$ leads to the best performance.

Ablation study of the number of codes in VQ-VAE. We study the number of codes in the codebook in Table 6. We find that the performance of 512 codes is slightly better than 1,024 codes. The results show that 256 codes are not sufficient for reconstruction.

5 Conclusion

In this work, we investigated a classic framework based on VQ-VAE and Transformers (T2M-GPT+ and T2M-GIT+) to synthesize human motion from textual descriptions. Our method achieved much better performances than concurrent diffusion-based approaches and T2M-GPT. We demonstrated that the large language model can significantly improve performance without any specific fine-tuning. We also investigated parallel decoding and showed that T2M-GIT+ largely improves the inference speed while obtaining comparable results as T2M-GPT+. Moreover, we explored in detail the effect of various quantization strategies and we provided an analysis of the dataset size. Our finding suggests that a larger dataset could still bring additional improvement to our approach.

References

[1] C. Ahuja and L.-P. Morency, "Language2pose: Natural language grounded pose forecasting," in *International Conference on 3D Vision (3DV)*, 2019.

[2] A. Ghosh, N. Cheema, C. Oguz, C. Theobalt, and P. Slusallek, "Synthesis of compositional animations from textual descriptions," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[3] G. Tevet, B. Gordon, A. Hertz, A. H. Bermano, and D. Cohen-Or, "Motionclip: Exposing human motion generation to clip space," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[4] M. Petrovich, M. J. Black, and G. Varol, "Action-conditioned 3D human motion synthesis with transformer VAE," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[5] ——, "TEMOS: Generating diverse human motions from textual descriptions," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[6] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning (ICML)*, 2021.

[7] C. Guo, S. Zou, X. Zuo, S. Wang, W. Ji, X. Li, and L. Cheng, "Generating diverse and natural 3d human motions from text," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[8] C. Guo, X. Zuo, S. Wang, and L. Cheng, "Tm2t: Stochastic and tokenized modeling for the reciprocal generation of 3d human motions and texts," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[9] J. Zhang, Y. Zhang, X. Cun, Y. Zhang, H. Zhao, H. Lu, X. Shen, and Y. Shan, "Generating human motion from textual descriptions with discrete representations," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[10] M. Zhang, Z. Cai, L. Pan, F. Hong, X. Guo, L. Yang, and Z. Liu, "Motiondiffuse: Text-driven human motion generation with diffusion model," *arXiv*, 2022.

[11] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[12] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[13] G. Tevet, S. Raab, B. Gordon, Y. Shafir, A. H. Bermano, and D. Cohen-Or, "Human motion diffusion model," *arXiv*, 2022.

[14] R. Dabral, M. H. Mughal, V. Golyanik, and C. Theobalt, "Mofusion: A framework for denoising-diffusion-based motion synthesis," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[15] C. Xin, B. Jiang, W. Liu, Z. Huang, B. Fu, T. Chen, J. Yu, and G. Yu, "Executing your commands via motion diffusion in latent space," *arXiv*, 2022.

[16] A. Van Den Oord, O. Vinyals *et al.*, "Neural discrete representation learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[17] W. Williams, S. Ringer, T. Ash, D. MacLeod, J. Dougherty, and J. Hughes, "Hierarchical quantized autoencoders," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[18] P. Esser, R. Rombach, and B. Ommer, "Taming transformers for high-resolution image synthesis," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[19] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," in *International Conference on Machine Learning (ICML)*, 2021.

[20] T. Ao, Q. Gao, Y. Lou, B. Chen, and L. Liu, "Rhythmic gesticulator: Rhythm-aware co-speech gesture synthesis with hierarchical neural embeddings," in *SIGGRAPH Asia*, 2022.

[21] S. Dieleman, A. van den Oord, and K. Simonyan, "The challenge of realistic music generation: modelling raw audio at scale," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[22] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, "Jukebox: A generative model for music," *arXiv*, 2020.

[23] T. Lucas, F. Baradel, P. Weinzaepfel, and G. Rogez, "Posegpt: Quantization-based 3d human motion generation and forecasting," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[25] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, "Improving language understanding by generative pre-training," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[26] H. Chang, H. Zhang, L. Jiang, C. Liu, and W. T. Freeman, "Maskgit: Masked generative image transformer," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[27] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "Llama: Open and efficient foundation language models," *arXiv*, 2023.

[28] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv*, 2018.

[29] M. Plappert, C. Mandery, and T. Asfour, "The kit motion-language dataset," *Big data*, 2016.

[30] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *International Conference on Learning Representations (ICLR)*, 2014.

[31] N. I. Badler, C. B. Phillips, and B. L. Webber, *Simulating humans: computer graphics animation and control*. Oxford University Press, 1993.

[32] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik, "Recurrent network models for human dynamics," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2015.

[33] J. Martinez, M. J. Black, and J. Romero, "On human motion prediction using recurrent neural networks," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[34] J. Butepage, M. J. Black, D. Kragic, and H. Kjellstrom, "Deep representation learning for human motion prediction and classification," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[35] D. Pavllo, D. Grangier, and M. Auli, "Quaternet: A quaternion-based recurrent model for human motion," in *Proceedings of the British Machine Vision Conference (BMVC)*, 2018.

[36] A. Hernandez, J. Gall, and F. Moreno-Noguer, "Human motion prediction via spatio-temporal inpainting," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[37] E. Barsoum, J. Kender, and Z. Liu, "Hp-gan: Probabilistic 3d human motion prediction via gan," in *Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018.

[38] W. Mao, M. Liu, M. Salzmann, and H. Li, "Learning trajectory dependencies for human motion prediction," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2019.

[39] W. Mao, M. Liu, and M. Salzmann, "History repeats itself: Human motion prediction via motion attention," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[40] W. Guo, Y. Du, X. Shen, V. Lepetit, X. Alameda-Pineda, and F. Moreno-Noguer, "Back to mlp: A simple baseline for human motion prediction," in *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, 2022.

[41] A. Bouazizi, A. Holzbock, U. Kressel, K. Dietmayer, and V. Belagiannis, "Motionmixer: Mlp-based 3d human body pose forecasting," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2022.

[42] Y. Du, R. Kips, A. Pumarola, S. Starke, A. Thabet, and A. Sanakoyeu, "Avatars grow legs: Generating smooth human motion from sparse tracking inputs with diffusion model," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[43] I. Habibie, D. Holden, J. Schwarz, J. Yearsley, and T. Komura, "A recurrent variational autoencoder for human motion synthesis," in *Proceedings of the British Machine Vision Conference (BMVC)*, 2017.

[44] X. Yan, A. Rastogi, R. Villegas, K. Sunkavalli, E. Shechtman, S. Hadap, E. Yumer, and H. Lee, "Mt-vae: Learning motion transformations to generate multimodal human dynamics," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[45] S. Aliakbarian, F. S. Saleh, M. Salzmann, L. Petersson, and S. Gould, "A stochastic conditioning scheme for diverse human motion prediction," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[46] F. G. Harvey and C. Pal, "Recurrent transition networks for character locomotion," in *SIGGRAPH Asia 2018 Technical Briefs*, 2018.

[47] M. Kaufmann, E. Aksan, J. Song, F. Pece, R. Ziegler, and O. Hilliges, "Convolutional autoencoders for human motion infilling," in *International Conference on 3D Vision (3DV)*, 2020.

[48] F. G. Harvey, M. Yurick, D. Nowrouzezahrai, and C. Pal, "Robust motion in-betweening," *ACM Transactions on Graphics (TOG)*, 2020.

[49] Y. Duan, T. Shi, Z. Zou, Y. Lin, Z. Qian, B. Zhang, and Y. Yuan, "Single-shot motion completion with transformer," *arXiv*, 2021.

[50] X. Tang, H. Wang, B. Hu, X. Gong, R. Yi, Q. Kou, and X. Jin, "Real-time controllable motion transition for characters," *ACM Transactions on Graphics (TOG)*, 2022.

[51] H.-Y. Lee, X. Yang, M.-Y. Liu, T.-C. Wang, Y.-D. Lu, M.-H. Yang, and J. Kautz, "Dancing to music," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[52] J. Li, Y. Yin, H. Chu, Y. Zhou, T. Wang, S. Fidler, and H. Li, "Learning to generate diverse dance motions with transformer," *arXiv*, 2020.

[53] R. Li, S. Yang, D. A. Ross, and A. Kanazawa, "Ai choreographer: Music conditioned 3d dance generation with aist++," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[54] A. Aristidou, A. Yiannakidis, K. Aberman, D. Cohen-Or, A. Shamir, and Y. Chrysanthou, "Rhythm is a dancer: Music-driven motion synthesis with global structure," *arXiv*, 2021.

[55] K. Chen, Z. Tan, J. Lei, S.-H. Zhang, Y.-C. Guo, W. Zhang, and S.-M. Hu, "Choreomaster: choreography-oriented music-driven dance synthesis," *ACM Transactions on Graphics (TOG)*, 2021.

[56] S. Li, W. Yu, T. Gu, C. Lin, Q. Wang, C. Qian, C. C. Loy, and Z. Liu, "Bailando: 3d dance generation by actor-critic gpt with choreographic memory," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[57] S. Yan, Z. Li, Y. Xiong, H. Yan, and D. Lin, "Convolutional sequence generation for skeleton-based action synthesis," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2019.

[58] D. Holden, J. Saito, and T. Komura, "A deep learning framework for character motion synthesis and editing," *ACM Transactions on Graphics (TOG)*, 2016.

[59] D. Holden, T. Komura, and J. Saito, "Phase-functioned neural networks for character control," *ACM Transactions on Graphics (TOG)*, 2017.

[60] S. Starke, H. Zhang, T. Komura, and J. Saito, "Neural state machine for character-scene interactions." *ACM Transactions on Graphics (TOG)*, 2019.

[61] S. Starke, I. Mason, and T. Komura, "Deepphase: periodic autoencoders for learning motion phase manifolds," *ACM Transactions on Graphics (TOG)*, 2022.

[62] T. R. Shaham, T. Dekel, and T. Michaeli, "Singan: Learning a generative model from a single natural image," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[63] P. Li, K. Aberman, Z. Zhang, R. Hanocka, and O. Sorkine-Hornung, "Ganimator: Neural motion synthesis from a single sequence," *ACM Transactions on Graphics (TOG)*, 2022.

[64] H. Ahn, T. Ha, Y. Choi, H. Yoo, and S. Oh, "Text2action: Generative adversarial synthesis from language to action," in *International Conference on Robotics and Automation (ICRA)*, 2018.

[65] Y. Zhi, X. Cun, X. Chen, X. Shen, W. Guo, S. Huang, and S. Gao, "Livelyspeaker: Towards semantic-aware co-speech gesture generation," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023.

[66] N. Athanasiou, M. Petrovich, M. J. Black, and G. Varol, "TEACH: Temporal Action Compositions for 3D Humans," in *International Conference on 3D Vision (3DV)*, 2022.

[67] ——, "SINC: Spatial composition of 3D human motions for simultaneous action generation," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023.

[68] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, 2019.

[69] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[70] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin *et al.*, "Opt: Open pre-trained transformer language models," *arXiv*, 2022.

[71] D. Driess, F. Xia, M. S. M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, W. Huang, Y. Chebotar, P. Sermanet, D. Duckworth, S. Levine, V. Vanhoucke, K. Hausman, M. Toussaint, K. Greff, A. Zeng, I. Mordatch, and P. Florence, "Palm-e: An embodied multimodal language model," *arXiv*, 2023.

[72] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. P. Xing, H. Zhang, J. E. Gonzalez, and I. Stoica, "Judging llm-as-a-judge with mt-bench and chatbot arena," *arXiv*, 2023.

[73] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong *et al.*, "A survey of large language models," *arXiv*, 2023.

[74] A. Razavi, A. Van den Oord, and O. Vinyals, "Generating diverse high-fidelity images with vq-vae-2," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[75] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[76] A. S. Lin, L. Wu, R. Corona, K. Tai, Q. Huang, and R. J. Mooney, "Generating animated videos of human activities from natural language descriptions," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[77] U. Bhattacharya, N. Rewkowski, A. Banerjee, P. Guhan, A. Bera, and D. Manocha, "Text2gestures: A transformer-based network for generating emotive body gestures for virtual agents," in *Virtual Reality and 3D User Interfaces (VR)*, 2021.

[78] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz, "Mocogan: Decomposing motion and content for video generation," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[79] Y. Zhang, D. Huang, B. Liu, S. Tang, Y. Lu, L. Chen, L. Bai, Q. Chu, N. Yu, and W. Ouyang, "Motiongpt: Finetuned llms are general-purpose motion generators," *arXiv*, 2023.

[80] B. Jiang, X. Chen, W. Liu, J. Yu, G. Yu, and T. Chen, "Motiongpt: Human motion as a foreign language," *arXiv*, 2023.

[81] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black, "Amass: Archive of motion capture as surface shapes," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2019.

[82] C. Guo, X. Zuo, S. Wang, S. Zou, Q. Sun, A. Deng, M. Gong, and L. Cheng, "Action2motion: Conditioned generation of 3d human motions," in *Proceedings of the ACM International Conference on Multimedia (ACMMM)*, 2020.

[83] C. Mandery, Ö. Terlemez, M. Do, N. Vahrenkamp, and T. Asfour, "The kit whole-body human motion database," in *International Conference on Robotics and Automation (ICRA)*, 2015.

[84] "Cmu graphics lab motion capture database," http://mocap.cs.cmu.edu/, accessed: 2022-11-11.

[85] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[86] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations (ICLR)*, 2019.

[87] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "Smpl: A skinned multi-person linear model," *ACM transactions on graphics (TOG)*, 2015.

APPENDIX A
VQ-VAE ARCHITECTURE

We illustrate the detailed architecture of VQ-VAE in Table 7. The dimensions of the HumanML3D [7] and KIT-ML [29] datasets feature D_{in} are 263 and 259 respectively.

TABLE 7: **Architecture of our Motion VQ-VAE.**

Components	Architecture
VQ-VAE Encoder	(0): Conv1D(D_{in}, 512, kernel_size=(3,), stride=(1,), padding=(1,)) (1): ReLU() (2): 2 $\times$ Sequential((0): Conv1D(512, 512, kernel_size=(4,), stride=(2,), padding=(1,)) (1): Resnet1D((0): ResConv1DBlock((activation1): ReLU() (conv1): Conv1D(512, 512, kernel_size=(3,), stride=(1,), padding=(9,), dilation=(9,)) (activation2): ReLU() (conv2): Conv1D(512, 512, kernel_size=(1,), stride=(1,))) (1): ResConv1DBlock((activation1): ReLU() (conv1): Conv1D(512, 512, kernel_size=(3,), stride=(1,), padding=(3,), dilation=(3,)) (activation2): ReLU() (conv2): Conv1D(512, 512, kernel_size=(1,), stride=(1,))) (2): ResConv1DBlock((activation1): ReLU() (conv1): Conv1D(512, 512, kernel_size=(3,), stride=(1,), padding=(1,)) (activation2): ReLU() (conv2): Conv1D(512, 512, kernel_size=(1,), stride=(1,)))))
Codebook	nn.Parameter((512, 512), requires_grad=False)
VQ-VAE Decoder	(0): 2 $\times$ Sequential((0): Conv1D(512, 512, kernel_size=(3,), stride=(1,), padding=(1,)) (1): Resnet1D((0): ResConv1DBlock((activation1): ReLU() (conv1): Conv1D(512, 512, kernel_size=(3,), stride=(1,), padding=(9,), dilation=(9,)) (activation2): ReLU() (conv2): Conv1D(512, 512, kernel_size=(1,), stride=(1,))) (1): ResConv1DBlock((activation1): ReLU() (conv1): Conv1D(512, 512, kernel_size=(3,), stride=(1,), padding=(3,), dilation=(3,)) (activation2): ReLU() (conv2): Conv1D(512, 512, kernel_size=(1,), stride=(1,))) (2): ResConv1DBlock((activation1): ReLU() (conv1): Conv1D(512, 512, kernel_size=(3,), stride=(1,), padding=(1,)) (activation2): ReLU() (conv2): Conv1D(512, 512, kernel_size=(1,), stride=(1,)))) (2): Upsample(scale_factor=2.0, mode=nearest) (3): Conv1D(512, 512, kernel_size=(3,), stride=(1,), padding=(1,)) (1): ReLU() (2): Conv1D(512, D_{in}, kernel_size=(3,), stride=(1,), padding=(1,))