Autoregressive Motion Generation with Gaussian Mixture-Guided Latent Sampling

Linnan Tu[1], Lingwei Meng[2], Zongyi Li[1], Hefei Ling[1]*, Shijuan Huang[1]
[1]Department of Computer Science and Technology, Huazhong University of Science and Technology
[2]The Chinese University of Hong Kong
{lntu, zongyili, lhefei, shijuan_huang}@hust.edu.cn
lmeng@se.cuhk.edu.hk

Abstract

Existing efforts in motion synthesis typically utilize either generative transformers with discrete representations or diffusion models with continuous representations. However, the discretization process in generative transformers can introduce motion errors, while the sampling process in diffusion models tends to be slow. In this paper, we propose a novel text-to-motion synthesis method GMMotion that combines a continuous motion representation with an autoregressive model, using the Gaussian mixture model (GMM) to represent the conditional probability distribution. Unlike prior autoregressive approaches relying on residual vector quantization, our model employs continuous motion representations derived from the VAE's latent space. This choice streamlines both the training and the inference processes while mitigating discretization errors. Specifically, we utilize a causal transformer to learn the distributions of continuous motion representations, which are modeled with a learnable Gaussian mixture model. Extensive experiments demonstrate that our model surpasses existing state-of-the-art models in the motion synthesis task.

1 Introduction

3D human motion synthesis, *i.e.*, generating a vivid action sequence by control conditions, holds promising applications in game development, embodied intelligence, and the animation. Two main paradigms are used today: (1) One paradigm uses generative transformers with discrete motion representation, such as GPT-like (1; 2; 3) or BERT-like (4; 5) models, to synthesize motions based on specific conditions. Typically, these methods require a vector quantization model (6) to convert continuous motion sequences into discrete codebook tokens. Subsequently, a generative transformer is trained either using a teacher-forcing approach to generate discrete motion tokens autoregressively or using a masked filling strategy to generate them non-autoregressively. Finally, a decoder synthesizes the final motion sequence. (2) The other paradigm employs diffusion models with continuous motion representation (7; 8; 9; 10). They first train a continuous autoencoder, primarily using VAE-based (11) models, to create a compressed and semantically rich representation of motion in latent space. The diffusion model then utilizes various sampling strategies (12; 9; 13) and conditional control methods (14; 15) to generate latent vectors of motion that align with the given conditions. However, both paradigms have disadvantages.

The VQ process inevitably disrupts the continuity of the time series, which can lead to errors during the token connection process. Some works employ residual VQ (RVQ) (4; 16), which involves iteratively summing residuals using multiple codebooks to mitigate compression loss. However, models using RVQ often have a structure similar to VALL-E (17), which requires separate processing

*Corresponding Author

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

of tokens for the primary layer and the residual layers, increasing the complexity of the model. Some studies use binary codebooks (18) or model body joints separately (19), but this can increase the codebook size and lead to higher storage costs. Moreover, the highly compressed nature of the motion can lead to less diverse outputs, favoring common patterns found in the training data.

The diffusion models offer better diversity with text conditions and generate high-quality motion (20; 21; 8); however, their inference speed is limited since the sampling process requires multiple iterations. Efforts to address these issues, such as MotionLCM (12) and B2A-HDM (22), have successfully decreased sampling steps through distillation techniques. Nevertheless, they require pre-specifying a maximum sequence length, which limits the scalability of motion generation. Therefore, we are motivated to develop an approach that reduces compression loss in motion representation while enabling continuous generation in multi-modal spaces.

In this paper, we propose a novel framework called Autoregressive Modeling with Gaussian Mixtures (GMMotion), which aims to synthesize motions with continuous GMM latent sapce, demonstrating that vector quantization is not a necessary prerequisite for autoregressive motion modeling. We lead the motion sequence into the Gaussian mixtures' latent space with learnable parameters in the first stage. By constraining the latent representation to be a continuous multi-modal distribution during VAE training and recovering it in the second stage with a continuous autoregressive (AR) model, we can build a AR model that retains all the advantages of Large Language Modeling (prompting, integrated duration modeling, and sampling) without many of the challenges associated with VQ. Additionally, our key advantage lies in eliminating the need to predefine the duration of generated content, enabling the synthesis of longer motion sequences based on the complexity of control conditions.

Our approach includes three major aspects:

- We utilize a learnable Gaussian mixture model to represent motion sequences as multi-modal distributions.
- We introduce an autoregressive causal transformer that learns the distribution of continuous representations and employs Gaussian mixture sampling to generate motion representations.
- We design a straightforward architecture that benefits from single-step Gaussian mixture sampling and AR generation, leading to extrapolatable inference and high-quality motion synthesis.

2 Related Work

2.1 Autoregressive Generation with Continuous Tokens

Autoregressive models (23; 24; 17) typically generate content utilizing quantized representations extracted from raw data (6; 25). However, recent studies (26; 27) find that as long as the per-token distributions are modeled, autoregressive models can be approached without vector quantization. MAR (27) introduced an image-masked autoregressive modeling method that uses an MLP head to perform diffusion sampling on several consecutive tokens at each iteration. LatentLM (28) introduces a multi-modal latent language model with a next-token diffusion approach, achieving good results in both speech synthesis and image generation tasks. However, the sequential iterative diffusion sampling process can lead to extremely slow inference speeds. MELLE (29) proposes an autoregressive text-to-speech synthesis model using uni-modal Gaussian sampling to accelerate the inference process. However, as shown in Figure 1, time series data often follow multi-modal distributions, making it difficult to accurately fit with an unimodal Gaussian distribution. VAE (12) exhibits an unimodal distribution but does not cluster the representations of movements. RVQ (4) captures the slightly chaotic multimodal distribution. Due to sampling errors and the varying movements of different joints, the raw data do not conform to a typical normal distribution, resulting in multi-modal distributions or even more complex distributions.

2.2 Motion Generation Methods

Research on human motion analysis has a long history (30), statistical models (31; 32; 1) have been employed in earlier studies. Some motion synthesis models leverage raw motion data for training generative models (7). However, these models can be affected by measurement errors in the

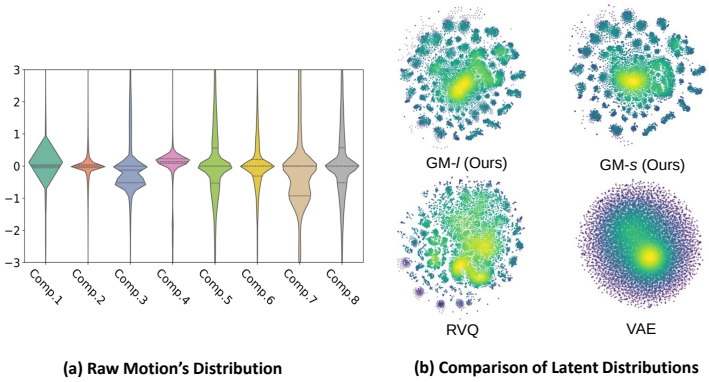

Figure 1: (a) Violin map of the motion's moving components randomly selected from the normalized HumanML3D datasets;(b) Visualization of different representation model's latent space with t-SNE. Yellow indicates dense sample distribution, while blue represents sparse sample distribution.

data, which often result from inaccuracies in the motion capture process (33; 34). Furthermore, the motion vectors corresponding to different joints exhibit specific distributions, making it challenging for these models to learn representations of complex movements. To maintain the continuity of time series, some approaches (3; 13) employ continuous representation learning models such as variational autoencoders (VAEs) (11; 35). In these models, an encoder predicts the mean and variance of motion latent vectors, followed by sampling from a Gaussian distribution to obtain the latent vector representation, which is then decoded to reconstruct the motion. Although VAEs (8) achieved satisfactory reconstruction results, they struggled to differentiate between various motions, thereby increasing the difficulty of learning for the subsequent generative model.

Recent work has leveraged Vector Quantized-Variational Autoencoders (VQ-VAEs) (5; 36; 18) to achieve discrete representation learning for motion. These models use codebook indices as motion tokens, enabling the application of language modeling (37) techniques and resulting in impressive generation outcomes. However, due to the inevitable quantization error introduced by the discrete process, autoregressive or masked generation methods based on VQ models have been constrained (29; 27).

3 Methodology

Given a motion description like "A person rolls forward once, then raises their hand and quickly shoots a ball.", our goal is to create a 3D motion sequence that reflects this description. In this paper, we propose a learnable Gaussian mixture model to represent motion sequences as multi-modal distributions, regularized by the Kolmogorov-Smirnov (KS) distance (Sec. 3.1). We then introduce a masked autoregressive transformer that learns the distribution of continuous representations and employs Gaussian mixture sampling to generate motion representations (Sec. 3.2).

Our proposed model comprises two primary components: a Gaussian Mixture Variational Autoencoder (GM-VAE) and an AR model. The GM-VAE incorporates causal convolutions to preserve temporal consistency in sequential data processing, along with a learnable prior distribution to regularize the latent space. The AR model consists of a text encoder, a causal transformer, a latent sampling module, and a PostNet. The latent sampling module applies Gaussian resampling to the autoregressively generated latent vectors from the transformer to produce coarse reconstructions, which the PostNet then refines into detailed motion outputs.

3.1 Stage 1: Learning Continuous Motion Representation

Motion data, which describes the movement of different parts of the body, can be highly variable due to the unique motion patterns of individual joints. This complexity makes it difficult to represent the data with a single normal distribution. For example, the movement of your feet when walking or running (including speed and rhythm) can differ significantly from the movements of other parts of your body. This motivated us to model multi-modal distributions in a continuous latent space to preserve the temporal integrity of motion data (as shown in Figure 1).

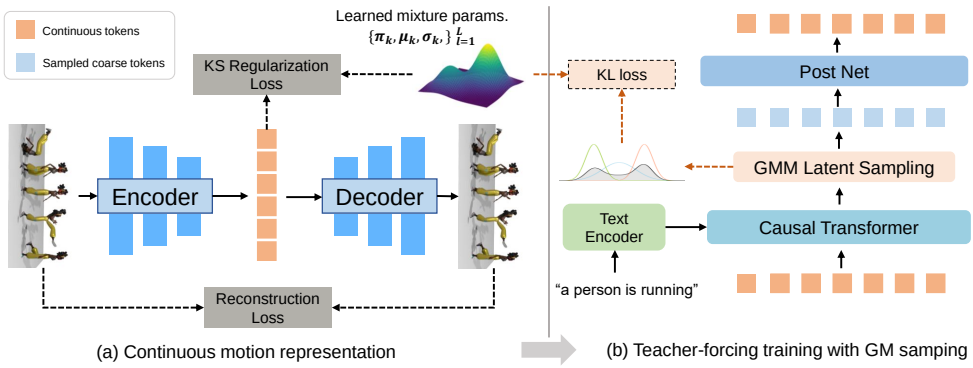

Figure 2: The training stage of the proposed GMMotion model. First, a VAE is trained to reconstruct motion data. The encoder obtains the posterior distribution of motions, which is shaped by a learnable Gaussian mixture, and then the decoder generates reconstructed motions from latent features. Next, a causal transformer generates latent representations, which are regularized by VAE's shared GMM parameters. Text embeddings and latent representations control the transformer to predict Gaussian variables for masked latent. Finally, continuous latent is restored through Gaussian mixture latent sampling.

Learnable GMM as a prior. Some works (38; 39) suggest that structured VAEs can effectively train deep models using a GMM as a prior distribution, replacing the normal distribution typically used in vanilla VAEs. Inspired by (39), an unsupervised clustering model using Gaussian mixture variational autoencoders, we propose the learnable mixture of Gaussians as a prior distribution to replace the single normal distribution. The evidence lower bound is:

$$\text{ELBO} = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})] - \lambda \text{KL}_{gmm}, \tag{1}$$

where λ controls the strength of the regularization, $\mathbb{E}_{q(z|x)}[\log p(x|z)]$ is the log-likelihood of the reconstructed data. Similar to VQ-based models, we obtain the latent vector z from the encoder through a deterministic mapping. However, since the posterior is a deterministic function and the prior is composed of a mixture of Gaussians, directly calculating the difference between these two distributions (known as KL divergence) is not straightforward (40). The reconstruction term, therefore, can be estimated by drawing Monte Carlo samples:

$$\begin{aligned} \text{KL}_{gmm} =& \text{KL}(q(\mathbf{z}|\mathbf{x}) \| \sum_{l=1}^{L} \pi_l \mathcal{N}(\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)) \\ \approx& \frac{1}{M} \sum_{j=1}^{M} \sum_{l=1}^{L} p_\beta^{(j)} \text{KL}(q(\mathbf{z}|\mathbf{x}) \| p(\mathbf{x}|\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l, \mathbf{t}_l = 1)), \end{aligned} \tag{2}$$

where l denotes the mixing exponent index, L is the total number of mixtures, and $\mathbf{t}$ is a one-hot vector sampled from the mixing probability π, which chooses one component from the Gaussian mixture. M is number of samples, $p_\beta^{(j)} = p(\mathbf{t}_l = 1|\mathbf{z}^{(j)})$ is the conditional probability of $\mathbf{t}_l$ being equal to 1 when given the sampled latent $\mathbf{z}^{(j)}$. The gradient can be backpropagated with the standard reparameterization trick (35). The prior term can be calculated analytically.

KS distance for latent regularization. Our loss definition is based on the Kolmogorov-Smirnov (KS) test (41) for equality of one-dimensional probability distributions. The KS test serves as a statistical tool to determine whether a set of N one-dimensional data points is drawn from a specified reference distribution. This determination is made by comparing the cumulative distribution function (CDF) of the reference distribution with the empirical CDF $\overline{F}_N$, which is derived from the observed samples.

For each mode $l \leq L$ in the GMM, let u_l represent the mean, Σ_l represent the covariance matrix, and π_l represent the weight of that specific mode. The CDFs for univariate Gaussian distributions can be defined as:

$$F_{\text{GMM},j}(z) = \sum_{l=1}^{L} \pi_l \Phi\left(\frac{z - [\mu_l]_j}{[\Sigma]_{j,j}}\right), \tag{3}$$

and the covariance matrix of the GMM can be computed as:

$$\Sigma_{\text{GMM}} = \sum_{l=1}^{L} p_l \Sigma_l + \sum_{l=1}^{L} p_l (\mu_l - \bar{\mu})(\mu_l - \bar{\mu})^T, \tag{4}$$

where $\bar{\mu} = \frac{1}{L}\sum_{l=1}^{L} \mu_l$, applying our proposed regularization method to multi-modal GMMs is a simple extension. Given d-dimensional latent samples $z_1, ..., z_N$, the empirical marginal CDF in dimension j is given by:

$$\bar{F}_j^{(N)}(z) = \frac{1}{n} \sum_{n=1}^{N} \mathbb{I}_{[z_n]_j \leq z}, \tag{5}$$

where $\mathbb{I}$ is an indicator function, the primary term in our loss function is specified as:

$$\mathcal{L}_{\text{KS},L}(\mathbf{z}_1, \ldots, \mathbf{z}_N) = \frac{1}{d} \sum_{j=1}^{d} \text{MSE}\left(\bar{F}_j^{(N)}(\mathbf{z}_j), F_{\text{GMM},j}(\mathbf{z}_j)\right). \tag{6}$$

Based on a motion-VAE (42), we use an L2 loss between the ground truth poses $\mathbf{x}$ and predictions $\hat{\mathbf{x}}$. We use an L2 loss between the root-centered vertices of the SMPL mesh (43) $\mathbf{v}$ and predictions $\hat{\mathbf{v}}$:

$$\mathcal{L}_{rec}(\hat{\mathbf{x}}, \mathbf{x}) = \|\hat{\mathbf{x}} - \mathbf{x}\|_2^2 + \|\hat{\mathbf{v}} - \mathbf{v}\|_2^2. \tag{7}$$

The final loss function includes motion reconstruction and latent space distribution constraints:

$$\mathcal{L}_{loss} = \lambda_{\text{KS}} \mathcal{L}_{\text{KS},L} + \mathcal{L}_{\text{rec}}. \tag{8}$$

3.2 Stage 2: Learning Autoregressive Latent Sampling with GMM

Considering that the motion representation in the first stage is constrained to be continuous and follows a GM distribution, in the second stage, we directly generate this distribution from a variational inference perspective, thereby avoiding the multi-step iterations of diffusion sampling (44).

The model uses a causal transformer architecture like T2M-GPT (36). Additionally, we designed a Residual-MLP composed of three-layer Multilayer Perceptrons to reorganize the sampled latent representations. Then, the coarse latent representations are refined through a PostNet with residual connections to reconstruct finer latent representations. In the case of Gaussian mixture generative modeling, we no longer need an embedding layer or a softmax layer. Since there are no mask/padding token IDs in the continuous case, we design learnable mask latent and padding latent to replace them.

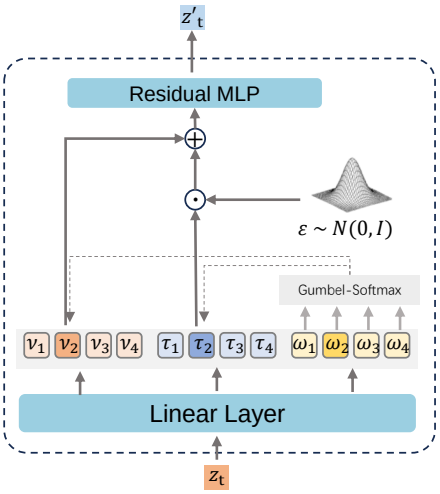

Figure 3: GMM latent sampling process.

Gaussian mixture latent sampling. We aim for the transformer to predict the parameters of a mixture of Gaussian distributions, from which we can sample to obtain latent vectors(Figure 3). We define an autoregressive model for the continuous random variable $z_t \in \mathbb{R}^D$, where the conditional probabilities are represented as a mixture of Gaussian distributions:

$$p(\mathbf{z}_t^{'} \mid \mathbf{z}_{t-1}^{'}, \ldots, \mathbf{z}_1^{'}, Y) = \sum_{l=1}^{L} \omega_l^t \mathcal{N}\left(\mathbf{z}_t; \boldsymbol{\nu}_l^t, (\boldsymbol{\tau}_l^t)^2\right), \tag{9}$$

where ω_n^t is the n-th GM's weights, $\boldsymbol{\nu}_n^t$ and $\boldsymbol{\tau}_n^t$ represent the n-th GM's mean and diagonal variances for generating step t. The mixture parameters are generated by a neural network $f()$, which takes the previous inputs and conditional information as its inputs:

$$[\omega_{1:L}^t, \boldsymbol{\nu}_{1:L}^t, \boldsymbol{\tau}_{1:L}^t] = f(\mathbf{z}_{t-1}, \ldots, \mathbf{z}_1, C), \tag{10}$$

where C is the condition embeddings, z is the AR transformer latent representations. We use one Linear layer to obtain the weights, means, and diagonal variances. We use the negative log-likelihood loss function as:

$$\mathcal{L}_{\text{NLL}} = \sum_{i=1}^{M} -\log p\left(\mathbf{z}_i \mid \{\omega_l^i, \nu_l^i, \tau_l^i\}_{l=1}^{L}\right) + \text{KL}(\sum_{l=1}^{L} \omega_l^{1:M} \mathcal{N}\left(\boldsymbol{\nu}_l^{1:M}, \boldsymbol{\tau}_l^{1:M}\right)) \| \sum_{l=1}^{L} \pi_l \mathcal{N}(\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)). \tag{11}$$

To ensure the mixture weights and variances are valid, we apply softmax to normalize the ω^t values and softplus to the τ^t values in the network output.

Head pre-padding and learnable rotary position encoding. Motion sequences in the same batch vary in length. Previous AR motion generation models pad shorter sequences with padding tokens at the tail to maintain uniform length and apply absolute position encoding to indicate positional relationships. Inspired by large language models (45), we fill padding tokens at the head of shorter sequences and employ learnable relative position encoding (46) to preserve positional relationships. Our goal is to enhance the scalability of the autoregressive model, enabling it to synthesize longer motion videos even when trained on shorter sequences from the HumanML3D dataset.

On the other hand, we adopt the same text-conditioning injection method as SALAD (47), where a cross-attention module with residual connections is added after each transformer block to inject text embeddings. We use the same text encoder as LAMP (16). Compared to embedding text conditions as the first token in the AR iteration process, this approach allows for more flexible control and better compatibility with our head pre-filling method.

4 Experimental Results

4.1 Datasets and evaluation metrics

Datasets. To fairly and accurately compare our method with the baseline, we used two main motion-language benchmarks: KITML (34) and HumanML3D (33). The KITML dataset comprises 3,911 actions from KIT motion data, with each action accompanied by one to four text notes (a total of 6,278 notes). The KITML motions are set at 12.5 frames per second (FPS). HumanML3D includes 14,616 actions from the AMASS (48) and HumanAct12 (49) datasets. Each action is described by three text scripts (a total of 44,970 notes). The HumanML3D motions are set at 20 FPS and last up to 10 seconds. We augmented the data by flipping motions and split both datasets into training, testing, and validation sets.

Evaluation metrics. We evaluate the generated motions in three aspects: (1) **Quality of the generated motions.** We use the Frechet Inception Distance (FID) to measure how close the generated motion patterns are to the real ones. (2) **Text-motion alignment.** We use the Matching Score to measure how well the texts match the generated motions. Additionally, we apply R-Precision(N) to assess how accurately motions can be retrieved based on their corresponding texts within a set of N motion-text pairs. (3) **Motion disversity.** MultiModality (MModality) measures the generation diversity conditioned on the same text and Diversity calculates variance through features (33).

4.2 Experimental setup

We use the same CNN-based encoder and decoder as Momask (4). We introduce a linear layer after the encoder (the same as (50)) and replace the vector quantization step with a learnable Gaussian mixture distribution. To maintain training stability, we make the mean learnable, initialize the weights with a uniform distribution, and fix the variance to be the identity matrix. The dimension of the 8-layer Causal Transformer is set to 512, with 8 heads and a dropout rate of 0.1, using the GELU activation function. Learnable RoPE embeddings are applied. The diagonal covariance matrices are set to be diagonal.

Methods	R-Precision ↑ Top-1	Top-2	Top-3	FID ↓	MM-Dist ↓	Diversity →	MultiModality ↑
Real motion	$0.511^{\pm.003}$	$0.703^{\pm.003}$	$0.797^{\pm.002}$	$0.002^{\pm.000}$	$2.974^{\pm.008}$	$9.503^{\pm.065}$	-
T2M-GPT (36)	$0.492^{\pm.003}$	$0.679^{\pm.002}$	$0.775^{\pm.002}$	$0.141^{\pm.005}$	$3.121^{\pm.009}$	$9.722^{\pm.082}$	$1.831^{\pm.048}$
AttT2M (51)	$0.499^{\pm.003}$	$0.690^{\pm.002}$	$0.786^{\pm.002}$	$0.112^{\pm.006}$	$3.038^{\pm.007}$	$9.700^{\pm.090}$	$2.452^{\pm.051}$
ParCo (52)	$0.515^{\pm.003}$	$0.706^{\pm.003}$	$0.801^{\pm.002}$	$0.109^{\pm.005}$	$2.927^{\pm.008}$	$9.576^{\pm.088}$	$1.382^{\pm.060}$
MoMask (4)	$0.521^{\pm.002}$	$0.713^{\pm.002}$	$0.807^{\pm.002}$	$0.045^{\pm.002}$	$2.958^{\pm.008}$	-	$1.241^{\pm.040}$
MoGenTS (19)	$0.529^{\pm.003}$	$0.719^{\pm.002}$	$0.812^{\pm.002}$	$0.033^{\pm.001}$	$2.867^{\pm.006}$	$9.570^{\pm.077}$	-
LaMP (16)	$0.557^{\pm.003}$	$0.751^{\pm.002}$	$0.843^{\pm.001}$	$0.032^{\pm.002}$	$2.759^{\pm.007}$	$9.571^{\pm.069}$	-
DiverseMotion (53)	$0.515^{\pm.003}$	$0.706^{\pm.002}$	$0.802^{\pm.002}$	$0.072^{\pm.004}$	$2.941^{\pm.007}$	$9.683^{\pm.102}$	$1.869^{\pm.089}$
MDM (7)	$0.320^{\pm.005}$	$0.498^{\pm.004}$	$0.611^{\pm.007}$	$0.544^{\pm.044}$	$5.566^{\pm.027}$	$9.559^{\pm.086}$	$2.799^{\pm.072}$
MLD (8)	$0.481^{\pm.003}$	$0.673^{\pm.003}$	$0.772^{\pm.002}$	$0.473^{\pm.013}$	$3.196^{\pm.010}$	$9.724^{\pm.082}$	$2.413^{\pm.079}$
MotionDiffuse (54)	$0.491^{\pm.001}$	$0.681^{\pm.001}$	$0.782^{\pm.001}$	$0.630^{\pm.001}$	$3.113^{\pm.001}$	$9.410^{\pm.049}$	$1.553^{\pm.042}$
ReMoDiffuse (55)	$0.510^{\pm.005}$	$0.698^{\pm.006}$	$0.795^{\pm.004}$	$0.103^{\pm.004}$	$2.974^{\pm.016}$	$9.018^{\pm.075}$	$1.795^{\pm.043}$
Fg-T2M++ (56)	$0.513^{\pm.002}$	$0.702^{\pm.002}$	$0.801^{\pm.003}$	$0.089^{\pm.004}$	$2.925^{\pm.007}$	$9.223^{\pm.114}$	$2.625^{\pm.084}$
GMMotion (Ours)	$0.572^{\pm.003}$	$0.761^{\pm.003}$	$0.852^{\pm.001}$	$0.086^{\pm.003}$	$2.743^{\pm.008}$	$9.792^{\pm.085}$	$2.033^{\pm.058}$
Real motion	$0.424^{\pm.005}$	$0.649^{\pm.006}$	$0.779^{\pm.006}$	$0.031^{\pm.004}$	$2.788^{\pm.012}$	$11.08^{\pm.097}$	-
T2M-GPT (36)	$0.416^{\pm.006}$	$0.627^{\pm.006}$	$0.745^{\pm.006}$	$0.514^{\pm.029}$	$3.007^{\pm.023}$	$10.92^{\pm.108}$	$1.570^{\pm.039}$
AttT2M (51)	$0.413^{\pm.006}$	$0.632^{\pm.006}$	$0.751^{\pm.006}$	$0.870^{\pm.039}$	$3.039^{\pm.021}$	$10.96^{\pm.123}$	$2.281^{\pm.047}$
ParCo (52)	$0.430^{\pm.004}$	$0.649^{\pm.007}$	$0.772^{\pm.006}$	$0.453^{\pm.027}$	$2.820^{\pm.028}$	$10.95^{\pm.094}$	$1.245^{\pm.022}$
MoMask (4)	$0.433^{\pm.007}$	$0.656^{\pm.005}$	$0.781^{\pm.005}$	$0.204^{\pm.011}$	$2.779^{\pm.022}$	-	$1.131^{\pm.043}$
DiverseMotion (53)	$0.416^{\pm.005}$	$0.637^{\pm.008}$	$0.760^{\pm.011}$	$0.468^{\pm.098}$	$2.892^{\pm.041}$	$10.87^{\pm.101}$	$2.062^{\pm.079}$
MoGenTS (19)	$0.445^{\pm.006}$	$0.671^{\pm.006}$	$0.797^{\pm.005}$	$0.143^{\pm.004}$	$2.711^{\pm.024}$	$10.92^{\pm.090}$	-
LaMP (16)	$0.479^{\pm.006}$	$0.691^{\pm.005}$	$0.826^{\pm.005}$	$0.141^{\pm.013}$	$2.704^{\pm.018}$	$10.93^{\pm.101}$	-
MDM (7)	$0.164^{\pm.004}$	$0.291^{\pm.004}$	$0.396^{\pm.004}$	$0.497^{\pm.021}$	$9.191^{\pm.022}$	$10.85^{\pm.109}$	$1.907^{\pm.214}$
MLD (8)	$0.390^{\pm.008}$	$0.609^{\pm.008}$	$0.734^{\pm.007}$	$0.404^{\pm.027}$	$3.204^{\pm.027}$	$10.80^{\pm.117}$	$2.192^{\pm.071}$
MotionDiffuse (54)	$0.417^{\pm.004}$	$0.621^{\pm.004}$	$0.739^{\pm.004}$	$1.954^{\pm.062}$	$2.958^{\pm.005}$	$11.10^{\pm.143}$	$0.730^{\pm.013}$
ReMoDiffuse (55)	$0.427^{\pm.014}$	$0.641^{\pm.004}$	$0.765^{\pm.055}$	$0.155^{\pm.006}$	$2.814^{\pm.012}$	$10.80^{\pm.105}$	$1.239^{\pm.028}$
Fg-T2M++ (56)	$0.442^{\pm.006}$	$0.657^{\pm.005}$	$0.781^{\pm.004}$	$0.135^{\pm.004}$	$2.696^{\pm.011}$	$10.99^{\pm.105}$	$1.255^{\pm.078}$
GMMotion (Ours)	$0.481^{\pm.005}$	$0.703^{\pm.006}$	$0.819^{\pm.004}$	$0.198^{\pm.012}$	$2.604^{\pm.023}$	$11.12^{\pm.095}$	$1.457^{\pm.039}$

Table 1: Quantitative evaluation results on the test sets of HumanML3D (top) and KIT-ML (bottom). ↑ and ↓ denote that higher and lower values are better, respectively, while → denotes that the values closer to the real motion are better. Red and blue colors indicate the best and the second best results.

4.3 Motion representation performance

In Table 2, we compared our motion GM-VAE with other motion tokenizers, such as RVQ-VAE (4), Transformer-VAE (8), and VQ-VAE (36), and found that our model demonstrates superior results in motion reconstruction.

Methods	FID ↓	MPJPE ↓	R-Precision ↑ Top 1	Top 2	Top 3
VQ-VAE	$0.081^{\pm.001}$	$72.6^{\pm.001}$	$0.483^{\pm.003}$	$0.680^{\pm.003}$	$0.780^{\pm.002}$
RVQ-VAE	$0.029^{\pm.001}$	$31.5^{\pm.001}$	$0.497^{\pm.002}$	$0.693^{\pm.003}$	$0.791^{\pm.002}$
Trans-VAE	$0.023^{\pm.001}$	$13.7^{\pm.001}$	$0.499^{\pm.002}$	$0.695^{\pm.003}$	$0.791^{\pm.003}$
GM-VAE-*s* (Ours)	$\mathbf{0.004}^{\pm.001}$	$\mathbf{9.4}^{\pm.001}$	$0.514^{\pm.002}$	$0.703^{\pm.002}$	$\mathbf{0.819}^{\pm.003}$
GM-VAE-*l* (Ours)	$0.008^{\pm.001}$	$10.2^{\pm.001}$	$\mathbf{0.518}^{\pm.002}$	$\mathbf{0.710}^{\pm.002}$	$0.811^{\pm.002}$

Table 2: **Reconstruction results** of latent encoders in our method vs baseline methods(VQ-VAE (36), RVQ (4) and VAE (12)) on HumanML3D (33) data. *s* and *l* mean 128 dims and 512 dims in GMM latent spaces.

We also analyzed how our latent space representation compares with others, as shown in the t-SNE plot (see Figure 1). Constrained by the standard normal distribution, VAE (12) exhibits an unimodal distribution but does not cluster the representations of movements. Meanwhile, due to uneven utilization of the codebook, RVQ (4) captures representations of high-frequency movements but is less sensitive to low-frequency movements. Our model not only represents multi-modal motion distributions more effectively but also achieves better clustering of motion data. This is facilitated by the use of Gaussian mixture distributions, which allows the model to capture detailed motion characteristics through unsupervised clustering.

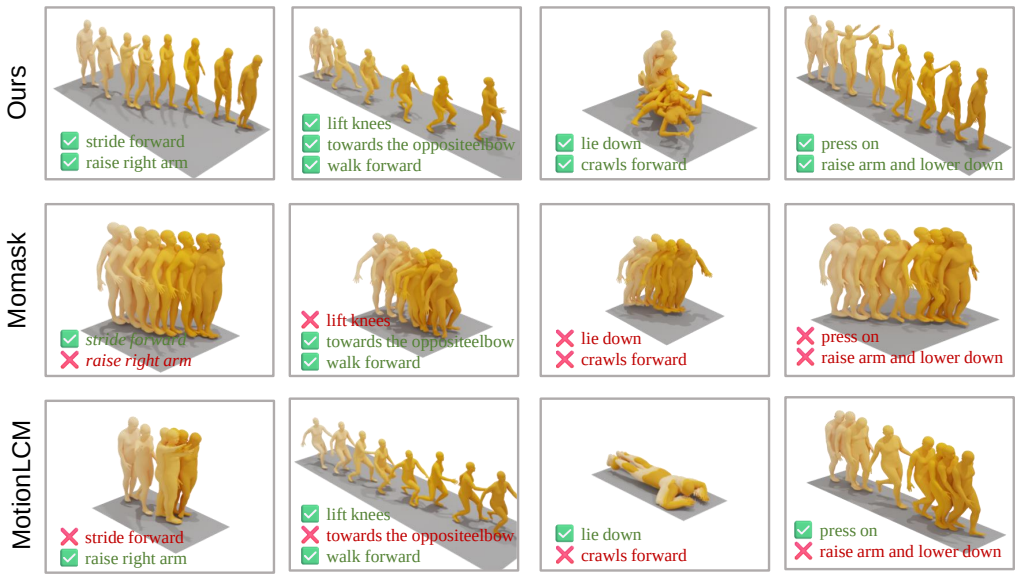

Figure 5: Visualization of qualitative results vs. diffusion based model (12) and VQ based model (4). The color from light yellow to dark yellow indicates the motion sequence order.

4.4 Text to motion Generation

Quantitative results. We compared our model with other state-of-the-art methods (5; 4; 36; 12; 7), which can be broadly divided into two types: VQ-based models and diffusion-based models. The results indicate that GMMotion achieves favorable outcomes in both text alignment and motion quality. In the qualitative analysis, we evaluated the quality of motion synthesis, the alignment between text and motion, and the diversity of motions. For the quantitative analysis, we compared our model's motion synthesis results with those of other baselines using the same text instructions.

Additionally, to demonstrate the feasibility of applying our method, we compare average inference time results in Appendix Table **??**.

Qualitative results. We also visualize our qualitative results in Figure 5. GMMotion demonstrates more accurate and natural-looking motions compared to the other models. For instance, in the action of striding forward with a raised arm, our model captures the movement fluidly, whereas Momask and MotionLCM exhibit some blurriness and less precise limb positioning. Similarly, in the knee-lifting motion, our model shows clearer and more realistic leg movements, while the other models struggle with the finer details. Overall, our model outperforms Momask and MotionLCM in generating coherent and lifelike human movements across different actions.

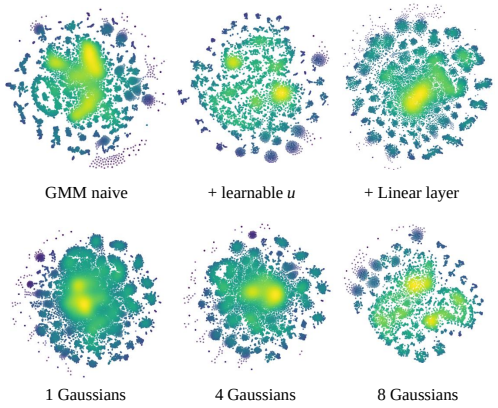

Figure 4: Visualization of different GMM settings with t-SNE.

4.5 Ablation Study

We focused on two aspects: (1) The effectiveness of the GMM. We explored its reconstruction performance under various conditions, including the number of components and regularization weights. (2) The continuous motion Transformer structure. We investigated how the transformer module, designed for continuous sampling, influences generation outcomes. We also discuss the

effectiveness of the generation methods in the Appendix. We examined the combined effects of different sampling and training approaches.

Ablation of Gaussian mixtures. In Table 3, we examine how the number of components and the loss weight λ influence the generative performance. It is evident that the reconstruction quality improves significantly when the number of components exceeds one, suggesting that a Gaussian mixture distribution has better fitting capabilities than a single Gaussian. We also adjusted the weight λ of the regularization constraint in the GM-VAE, and the results indicate that the model performs best when the weight is set to 1. In Figure 4, we find that adding a linear layer in the vanilla VAE and setting the GMM parameters to be learnable can improve the latent space representation.

No.Gaus	KS weight λ	FID↓	Matching score↓	R-Pre.↑
1	0.1	$0.121^{\pm.004}$	$2.939^{\pm.008}$	$0.815^{\pm.006}$
	1.0	$0.145^{\pm.005}$	$2.942^{\pm.007}$	$0.805^{\pm.005}$
	10.0	$0.165^{\pm.004}$	$3.080^{\pm.006}$	$0.799^{\pm.004}$
4	0.1	$0.088^{\pm.003}$	$2.781^{\pm.006}$	$0.841^{\pm.004}$
	1.0	$0.086^{\pm.003}$	$2.743^{\pm.008}$	$0.852^{\pm.001}$
	10.0	$0.142^{\pm.004}$	$2.892^{\pm.007}$	$0.827^{\pm.005}$
8	0.1	$0.101^{\pm.002}$	$3.011^{\pm.011}$	$0.801^{\pm.004}$
	1.0	$0.196^{\pm.009}$	$3.110^{\pm.009}$	$0.782^{\pm.004}$
	10.0	$0.659^{\pm.016}$	$3.556^{\pm.010}$	$0.679^{\pm.006}$

Table 3: Text-to-motion results with different Gaussian components and KS weights.

Ablation of AR architecture. In Table 4, we present the results of ablation studies on the structure of the AR model. The findings indicate that the head pre-padding and RoPE modules improve the quality of motion, while the motion reorganization module has a significant impact on the synthesis effect. We believe the motion reorganization module plays a crucial role in refining the rough latent representations after sampling. When we removed Gaussian sampling, reverting the model to deterministic sampling, there was a notable decline in performance. This suggests that the stochastic nature of Gaussian sampling is essential for maintaining high-quality motion synthesis.

Components	FID↓	Matching score↓	R-Pre.↑
Full	$0.086^{\pm.003}$	$2.743^{\pm.008}$	$0.852^{\pm.001}$
w/o RoPE	$0.133^{\pm.006}$	$2.851^{\pm.010}$	$0.823^{\pm.007}$
w/o Res	$0.945^{\pm.028}$	$3.423^{\pm.018}$	$0.752^{\pm.009}$
w/o GM Samp.	$0.485^{\pm.013}$	$3.241^{\pm.014}$	$0.791^{\pm.004}$
w/o Post	$0.141^{\pm.004}$	$2.939^{\pm.010}$	$0.813^{\pm.005}$

Table 4: Results of AR. Where *Res* is the residual net, *GM Samp.* is the Gaussian mixture sampling stage, and *Post* is the PostNet.

User study. To evaluate the perceptual quality of text-driven motion generation, we conducted a user study with 19 participants comparing our method against two baselines: LaMP (16), which generates motions by discrete AR models; and MotionLCM (50), which generates motions by continuous diffusion models. For each method, participants were presented with 15 video examples and asked to evaluate them based on two criteria: visual quality and text-motion alignment. All ratings were collected using a 5-point Likert scale ranging from 1 (poorest) to 5 (best). The results demonstrate that our model exhibits advantages in terms of visual quality and text-motion alignment.

Methods	Visual Quality	Text-motion Alignment
LaMP (16)	$3.962^{\pm.171}$	$3.775^{\pm.186}$
MotionLCM (50)	$3.418^{\pm.163}$	$3.219^{\pm.196}$
GMMotion (Ours)	$4.392^{\pm.093}$	$4.121^{\pm.098}$

Table 5: User study results.

5 Conclusion

We introduced GMMotion, a novel text-to-motion synthesis framework that employs continuous motion representation and GMM to capture multimodal human motions. GMMotion streamlines training and inference by avoiding vector quantization, instead sampling from learnable GMMs in the latent space. Our two-stage model—first modeling motion sequences into multimodal distributions with GMM, then using a causal transformer for efficient generation—outperforms existing models in quality and alignment.

Limitations. While GMMotion achieves efficient motion synthesis through autoregressive generation, the sequential nature of the process may occasionally lead to minor error accumulation over long sequences. Early prediction inaccuracies (e.g., subtle joint angle deviations) could propagate temporally, potentially affecting the smoothness of extended motions.

Acknowledgments

This work was supported in part by the Natural Science Foundation of China under Grant 62372203 and 62302186, in part by the Major Scientific and Technological Project of Shenzhen (202316021), in part by the National key research and development program of China(2022YFB2601802), in part by the Major Scientific and Technological Project of Hubei Province (2022BAA046, 2022BAA042).

References

[1] L. Siyao, 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 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 11 050–11 059.

[2] Z. Zhang, A. Liu, I. Reid, R. Hartley, B. Zhuang, and H. Tang, "Motion mamba: Efficient and long sequence motion generation with hierarchical and bidirectional selective ssm," *arXiv preprint arXiv:2403.07487*, 2024.

[3] H. Kong, K. Gong, D. Lian, M. B. Mi, and X. Wang, "Priority-centric human motion generation in discrete latent space," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 14 806–14 816.

[4] C. Guo, Y. Mu, M. G. Javed, S. Wang, and L. Cheng, "Momask: Generative masked modeling of 3d human motions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 1900–1910.

[5] E. Pinyoanuntapong, P. Wang, M. Lee, and C. Chen, "Mmm: Generative masked motion model," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[6] A. Van Den Oord, O. Vinyals *et al.*, "Neural discrete representation learning," *Advances in neural information processing systems*, vol. 30, 2017.

[7] G. Tevet, S. Raab, B. Gordon, Y. Shafir, D. Cohen-or, and A. H. Bermano, "Human motion diffusion model," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=SJ1kSyO2jwu

[8] X. Chen, B. Jiang, W. Liu, Z. Huang, B. Fu, T. Chen, and G. Yu, "Executing your commands via motion diffusion in latent space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 18 000–18 010.

[9] H. Liang, J. Bao, R. Zhang, S. Ren, Y. Xu, S. Yang, X. Chen, J. Yu, and L. Xu, "Omg: Towards open-vocabulary motion generation via mixture of controllers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 482–493.

[10] K. Karunratanakul, K. Preechakul, E. Aksan, T. Beeler, S. Suwajanakorn, and S. Tang, "Optimizing diffusion noise can serve as universal motion priors," in *arxiv:2312.11994*, 2023.

[11] C. Felce, S. Liorsdóttir, and L. Pachter, "The virial theorem and the price equation," 2024. [Online]. Available: https://arxiv.org/abs/2312.06114

[12] W. Dai, L.-H. Chen, J. Wang, J. Liu, B. Dai, and Y. Tang, "Motionlcm: Real-time controllable motion generation via latent consistency model," *arXiv preprint arXiv:2404.19759*, 2024.

[13] G. Barquero, S. Escalera, and C. Palmero, "Seamless human motion composition with blended positional encodings," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[14] Q. Zhang, J. Song, X. Huang, Y. Chen, and M.-Y. Liu, "Diffcollage: Parallel generation of large content with diffusion models," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2023, pp. 10 188–10 198.

[15] Y. Shafir, G. Tevet, R. Kapon, and A. H. Bermano, "Human motion diffusion as a generative prior," *arXiv preprint arXiv:2303.01418*, 2023.

[16] Z. Li, W. Yuan, Y. He, L. Qiu, S. Zhu, X. Gu, W. Shen, Y. Dong, Z. Dong, and L. T. Yang, "Lamp: Language-motion pretraining for motion generation, retrieval, and captioning," 2025. [Online]. Available: https://arxiv.org/abs/2410.07093

[17] S. Chen, S. Liu, L. Zhou, Y. Liu, X. Tan, J. Li, S. Zhao, Y. Qian, and F. Wei, "Vall-e 2: Neural codec language models are human parity zero-shot text to speech synthesizers," 2024. [Online]. Available: https://arxiv.org/abs/2406.05370

[18] S. Lu, J. Wang, Z. Lu, L.-H. Chen, W. Dai, J. Dong, Z. Dou, B. Dai, and R. Zhang, "Scamo: Exploring the scaling law in autoregressive motion generation model," 2024. [Online]. Available: https://arxiv.org/abs/2412.14559

[19] W. Yuan, W. Shen, Y. He, Y. Dong, X. Gu, Z. Dong, L. Bo, and Q. Huang, "Mogents: Motion generation based on spatial-temporal joint modeling," 2024. [Online]. Available: https://arxiv.org/abs/2409.17686

[20] K. Karunratanakul, K. Preechakul, S. Suwajanakorn, and S. Tang, "Guided motion diffusion for controllable human motion synthesis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 2151–2162.

[21] O. Bar-Tal, L. Yariv, Y. Lipman, and T. Dekel, "Multidiffusion: Fusing diffusion paths for controlled image generation," *Proceedings of Machine Learning Research*, vol. 202, pp. 1737–1752, 2023.

[22] Z. Xie, Y. Wu, X. Gao, Z. Sun, W. Yang, and X. Liang, "Towards detailed text-to-motion synthesis via basic-to-advanced hierarchical diffusion model," 2023. [Online]. Available: https://arxiv.org/abs/2312.10960

[23] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019.

[24] L. Yu, J. Lezama, N. B. Gundavarapu, L. Versari, K. Sohn, D. Minnen, Y. Cheng, V. Birodkar, A. Gupta, X. Gu, A. G. Hauptmann, B. Gong, M.-H. Yang, I. Essa, D. A. Ross, and L. Jiang, "Language model beats diffusion – tokenizer is key to visual generation," 2024. [Online]. Available: https://arxiv.org/abs/2310.05737

[25] P. Esser, R. Rombach, and B. Ommer, "Taming transformers for high-resolution image synthesis," 2021. [Online]. Available: https://arxiv.org/abs/2012.09841

[26] L. Fan, T. Li, S. Qin, Y. Li, C. Sun, M. Rubinstein, D. Sun, K. He, and Y. Tian, "Fluid: Scaling autoregressive text-to-image generative models with continuous tokens," 2024. [Online]. Available: https://arxiv.org/abs/2410.13863

[27] T. Li, Y. Tian, H. Li, M. Deng, and K. He, "Autoregressive image generation without vector quantization," 2024. [Online]. Available: https://arxiv.org/abs/2406.11838

[28] Y. Sun, H. Bao, W. Wang, Z. Peng, L. Dong, S. Huang, J. Wang, and F. Wei, "Multimodal latent language modeling with next-token diffusion," 2024. [Online]. Available: https://arxiv.org/abs/2412.08635

[29] L. Meng, L. Zhou, S. Liu, S. Chen, B. Han, S. Hu, Y. Liu, J. Li, S. Zhao, X. Wu, H. Meng, and F. Wei, "Autoregressive speech synthesis without vector quantization," 2024. [Online]. Available: https://arxiv.org/abs/2407.08551

[30] V. M. Spitzer and D. G. Whitlock, "The visible human dataset: the anatomical platform for human simulation," *The Anatomical Record: An Official Publication of the American Association of Anatomists*, vol. 253, no. 2, pp. 49–57, 1998.

[31] R. Bowden, "Learning statistical models of human motion," in *IEEE Workshop on Human Modeling, Analysis and Synthesis, CVPR*, vol. 2000, 2000.

[32] A. Galata, N. Johnson, and D. Hogg, "Learning variable-length markov models of behavior," *Computer Vision and Image Understanding*, vol. 81, no. 3, pp. 398–413, 2001.

[33] 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 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5152–5161.

[34] M. Plappert, C. Mandery, and T. Asfour, "The kit motion-language dataset," *Big data*, vol. 4, no. 4, pp. 236–252, 2016.

[35] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2014. [Online]. Available: https://arxiv.org/abs/1312.6114

[36] J. Zhang, Y. Zhang, X. Cun, S. Huang, Y. Zhang, H. Zhao, H. Lu, and X. Shen, "T2m-gpt: Generating human motion from textual descriptions with discrete representations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[37] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *ICML*, 2021.

[38] I. Higgins, L. Matthey, X. Glorot, A. Pal, B. Uria, C. Blundell, S. Mohamed, and A. Lerchner, "Early visual concept learning with unsupervised deep learning," 2016. [Online]. Available: https://arxiv.org/abs/1606.05579

[39] N. Dilokthanakul, P. A. M. Mediano, M. Garnelo, M. C. H. Lee, H. Salimbeni, K. Arulkumaran, and M. Shanahan, "Deep unsupervised clustering with gaussian mixture variational autoencoders," 2017. [Online]. Available: https://arxiv.org/abs/1611.02648

[40] A. Saseendran, K. Skubch, S. Falkner, and M. Keuper, "Shape your space: A gaussian mixture regularization approach to deterministic autoencoders," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 7319–7332. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/3c057cb2b41f22c0e740974d7a428918-Paper.pdf

[41] R. H. Lopes, I. Reid, and P. R. Hobson, "The two-dimensional kolmogorov-smirnov test," 2007.

[42] M. Petrovich, M. J. Black, and G. Varol, "Action-conditioned 3d human motion synthesis with transformer vae," 2021. [Online]. Available: https://arxiv.org/abs/2104.05670

[43] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "Smpl," *ACM Transactions on Graphics*, p. 1–16, Nov 2015. [Online]. Available: http://dx.doi.org/10.1145/2816795.2818013

[44] Z. Meng, Y. Xie, X. Peng, Z. Han, and H. Jiang, "Rethinking diffusion for text-driven human motion generation," 2024. [Online]. Available: https://arxiv.org/abs/2411.16575

[45] J. Xu, Z. Guo, J. He, H. Hu, T. He, S. Bai, K. Chen, J. Wang, Y. Fan, K. Dang, B. Zhang, X. Wang, Y. Chu, and J. Lin, "Qwen2.5-omni technical report," 2025. [Online]. Available: https://arxiv.org/abs/2503.20215

[46] J. Su, Y. Lu, S. Pan, A. Murtadha, B. Wen, and Y. Liu, "Roformer: Enhanced transformer with rotary position embedding," 2023. [Online]. Available: https://arxiv.org/abs/2104.09864

[47] S. Bae, H. Kim, Y. Choi, and J.-H. Lee, "Salad: Improving robustness and generalization through contrastive learning with structure-aware and llm-driven augmented data," 2025. [Online]. Available: https://arxiv.org/abs/2504.12185

[48] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. Black, "Amass: Archive of motion capture as surface shapes," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct 2019. [Online]. Available: http://dx.doi.org/10.1109/iccv.2019.00554

[49] 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 28th ACM International Conference on Multimedia*, Oct 2020. [Online]. Available: http://dx.doi.org/10.1145/3394171.3413635

[50] W. Dai, L.-H. Chen, Y. Huo, J. Wang, J. Liu, B. Dai, and Y. Tang, "Motionlcm-v2: Improved compression rate for multi-latent-token diffusion," December 2024. [Online]. Available: https://huggingface.co/blog/wxDai/motionlcm-v2

[51] C. Zhong, L. Hu, Z. Zhang, and S. Xia, "Attt2m: Text-driven human motion generation with multi-perspective attention mechanism," 2023. [Online]. Available: https://arxiv.org/abs/2309.00796

[52] Q. Zou, S. Yuan, S. Du, Y. Wang, C. Liu, Y. Xu, J. Chen, and X. Ji, "Parco: Part-coordinating text-to-motion synthesis," 2024. [Online]. Available: https://arxiv.org/abs/2403.18512

[53] Y. Lou, L. Zhu, Y. Wang, X. Wang, and Y. Yang, "Diversemotion: Towards diverse human motion generation via discrete diffusion," 2023. [Online]. Available: https://arxiv.org/abs/2309.01372

[54] 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 preprint arXiv:2208.15001*, 2022.

[55] M. Zhang, X. Guo, L. Pan, Z. Cai, F. Hong, H. Li, L. Yang, and Z. Liu, "Remodiffuse: Retrieval-augmented motion diffusion model," *arXiv preprint arXiv:2304.01116*, 2023.

[56] Y. Wang, M. Li, J. Liu, Z. Leng, F. W. B. Li, Z. Zhang, and X. Liang, "Fg-t2m++: Llms-augmented fine-grained text driven human motion generation," 2025. [Online]. Available: https://arxiv.org/abs/2502.05534

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: We demonstrate our novel GMM-based design.

   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: We discussed the impact of the error accumulation process.

   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: [Yes]

Justification: We have provided a thorough theoretical explanation for GMM sampling.

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: The code will be made publicly available after the review process.

   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: The code will be made publicly available after the review process.

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: We repeated the description of the experiment.

   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: [Yes]

   Justification: We presented the results in detail.

   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: We have provided the relevant content in the appendix.

   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: We have included the relevant content.

   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: We discussed the content related to this issue.

    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: [Yes]

    Justification: We discussed the content related to this issue.

    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: We discussed the content related to this issue.

    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: [Yes]

Justification: we communicate the details of the model as part of submissions via structured templates.

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: [Yes]

Justification: We discussed the content related to this issue.

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: [Yes]

Justification: We discussed the content related to this issue.

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.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: We used LLMs in accordance with the conference guidelines.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.