
Autoregressive Motion Generation with Gaussian Mixture-Guided Latent Sampling

Linnan Tu¹, Lingwei Meng², Zongyi Li¹, Hefei Ling^{1*}, Shijuan Huang¹

¹Department of Computer Science and Technology, Huazhong University of Science and Technology

²The Chinese University of Hong Kong

{lntu, zongyili, lhefei, shijuan_huang}@hust.edu.cn

lmeng@se.cuhk.edu.hk

A Preliminaries: Gaussian Mixture Model

Demonstrations can be found at (<https://gmmotion.github.io/>). A Gaussian Mixture Model (GMM) is a probabilistic model that assumes all the data points are generated from a mixture of several Gaussian distributions with unknown parameters. The general form of a GMM can be written as:

$$p(x|\theta) = \sum_{l=1}^L \pi_l \mathcal{N}(x|\mu_l, \Sigma_l), \quad (1)$$

where π_l represents the mixing coefficient for the l -th component and satisfies $\sum_{l=1}^L \pi_l = 1$, ensuring that the distribution integrates to one. Each component $\mathcal{N}(x|\mu_l, \Sigma_l)$ is a Gaussian distribution defined by its mean vector μ_l and covariance matrix Σ_l . Specifically, the probability density function of a multivariate Gaussian distribution is given by:

$$\mathcal{N}(x|\mu_l, \Sigma_l) = \frac{1}{(2\pi)^{D/2} |\Sigma_l|^{1/2}} \exp \left(-\frac{1}{2} \delta_l^T \Sigma_l^{-1} \delta_l \right), \quad (2)$$

where D is the dimensionality of the data points x , and $|\Sigma_l|$ denotes the determinant of the covariance matrix Σ_l , δ_l denotes $x - \mu_l$. The parameters of a GMM include the means $\{\mu_l\}_{l=1}^L$, covariances $\{\Sigma_l\}_{l=1}^L$, and mixing coefficients $\{\pi_l\}_{l=1}^L$. Estimating these parameters from data typically involves using an iterative algorithm such as the Expectation-Maximization (EM) algorithm.

B Metric Definitions

Time costs: To assess the inference efficiency of models, we follow to report the Average Inference Time per Sentence (AITS) measured in seconds. We calculate AITS on the test set of HumanML3D, set the batch size to 1, and exclude the time cost for model and dataset loading.

Frechet Inception Distance: Frechet Inception Distance (FID) measures the distributional difference between the generated and real motions, calculated using the feature extractor associated with a specific dataset:

$$\text{FID} = \|\mu_{\text{gen}} - \mu_{\text{gt}}\|^2 - \text{Tr}(\text{Cov}_{\text{gen}} + \text{Cov}_{\text{gt}} - 2(\text{Cov}_{\text{gen}} \text{Cov}_{\text{gt}})^{\frac{1}{2}}), \quad (3)$$

where μ_{gen} and μ_{gt} are the mean of \mathbf{f}_{gen} and \mathbf{f}_{gt} , Cov denotes the covariance matrix, and Tr denotes the trace of a matrix.

R-precision: R-precision (Top-1, Top-2, Top-3) measures the matching between the motion and the text. For each generated motion, its ground-truth text description and 31 randomly selected

*Corresponding Author

mismatched descriptions from the test set form a description pool. Then this metric is calculated by computing and ranking the Euclidean distances between the motion feature and the text feature of each description in the pool. Then the average accuracy is counted at Top-1, Top-2, and Top-3 places. The ground truth entry falling into the top-k candidates is treated as successful retrieval.

MultiModality Distance: Different from Diversity, MultiModality Distance (MM-Dist) measures how much the generated motions diversify within each text description. Specifically, a set of text descriptions with size C is randomly sampled from all descriptions. Then we randomly sample two subsets with the same size I from all generated motions conditioned by c -th text description, with extracted feature vectors $\{\mathbf{v}_{c,1}, \dots, \mathbf{v}_{c,I}\}$ and $\{\mathbf{v}'_{c,1}, \dots, \mathbf{v}'_{c,I}\}$. MModality is formalized as follows,

$$\text{MModality} = \frac{1}{C \times I} \sum_{c=1}^C \sum_{i=1}^I \|\mathbf{v}_{c,i} - \mathbf{v}'_{c,i}\|_2. \quad (4)$$

CLIP-Score: The CLIP-Score measures the alignment between an image I and its corresponding text description T . This score utilizes the CLIP model to embed both the image and text into a common feature space, subsequently calculating their cosine similarity as follows:

$$\text{CLIP-Score}(I, T) = \cos(\mathbf{f}_I, \mathbf{f}_T) = \frac{\mathbf{f}_I \cdot \mathbf{f}_T}{\|\mathbf{f}_I\| \|\mathbf{f}_T\|} \quad (5)$$

where \mathbf{f}_I represents the feature vector of the image I , \mathbf{f}_T denotes the feature vector of the text T , and \cdot indicates the dot product operation. The numerator computes the dot product between the two feature vectors, while the denominator normalizes this product by the magnitudes of the feature vectors, ensuring that the resulting cosine similarity falls within the range $[-1, 1]$. Higher CLIP-Scores indicate greater alignment between the image and text, with scores closer to 1 suggesting nearly identical representations in the feature space.

C More Details

C.1 Training Details

During training, we use the AdamW optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.99$. Following prior works (1; 2; 3), the batch size is set to 256 and 512 for training the AutoEncoder on the HumanML3D and KITML datasets, respectively, with each sample containing 64 frames. For training the generation branch, the batch size is set to 64 for HumanML3D and 16 for KIT-ML, with a maximum sequence length of 196 frames. The learning rate is set at 2×10^{-4} with a linear warmup of 2000 steps. All experiments are trained on 4 NVIDIA RTX 4090 GPUs.

C.2 AR Transformer Structure

AR transformer: Following (2), the model is implemented using a 4-layer causal transformer architecture. The dimension of the Motion Transformer is set to 512, with 8 heads and a dropout rate of 0.1, using the GELU activation function.

Sampling layer: We use a simple Linear layer to predict GMM parameters means, covariance, and weights. The variance type is set to diagonal. We employ the reparameterization trick, where a single linear layer is directly used and then partitioned to obtain different parameters.

ResidualNet: The input motion latents are projected to the model’s embedding dimensionality using a 3-layer perceptron with a 0.5 dropout rate enabled during both training and inference.

PostNet: Following previous model (4), we employ multiple convolutional blocks as the post-net to produce a residual that is added to coarse latent, resulting in the refined latent representation. The post-net consists of 5 convolutional blocks with a kernel size of 5 and 256 intermediate channels.

D More results

D.1 Ablation of Generation method.

In Table 1, we compared the synthesis effects of GMMotion under different generation approaches, where AR denotes autoregressive generation, and NAR stands for non-autoregressive (bidirectional masked) generation. MAR denotes masked autoregressive generation (5).

Gene. Method	latent dims	FID↓	Matching score↓	CLIP score↑	R-Pre.↑
AR	32	$0.560 \pm .008$	$3.468 \pm .011$	$0.611 \pm .006$	$0.734 \pm .008$
	128	$0.354 \pm .009$	$3.398 \pm .010$	$0.623 \pm .004$	$0.766 \pm .005$
	512	$0.131 \pm .004$	$3.315 \pm .006$	$0.634 \pm .004$	$0.778 \pm .006$
NAR	32	$0.311 \pm .006$	$3.360 \pm .004$	$0.629 \pm .004$	$0.769 \pm .003$
	128	$0.149 \pm .006$	$3.296 \pm .007$	$0.632 \pm .004$	$0.790 \pm .003$
	512	$0.140 \pm .004$	$3.345 \pm .007$	$0.640 \pm .005$	$0.784 \pm .004$
MAR	32	$0.210 \pm .004$	$3.349 \pm .003$	$0.631 \pm .003$	$0.782 \pm .007$
	128	$0.159 \pm .009$	$3.299 \pm .009$	$0.629 \pm .003$	$0.792 \pm .004$
	512	$0.136 \pm .005$	$3.340 \pm .008$	$0.642 \pm .004$	$0.786 \pm .003$

Table 1: Text-to-motion results of generation approaches. AR means autoregressive generation, NAR means non-autoregressive/masked generation and MAR means masked autoregressive generation like MARDM. Results are evaluated on *MARDM-HumanML3D benchmark*.

The results indicate that the AR method yields superior synthesis performance. Additionally, we examined the impact of latent space dimensions on the synthesis effect. The findings reveal that as the dimensionality increases, there is a noticeable improvement in the generation quality of the model; however, text alignment decreases to some extent.

D.2 Ablation of Pose Velocity and Acceleration Loss.

In the first stage of the reconstruction process, we introduced velocity loss and acceleration loss for guiding motion poses, each with its own weight. Furthermore, we conducted ablation experiments, which showed that the best motion reconstruction results are achieved when the variable values are set to 1.

Methods	FID ↓	MPIPE ↓	R-Precision ↑		
			Top 1	Top 2	Top 3
$\alpha_1 = 1, \alpha_2 = 0$	$0.010 \pm .001$	$10.5 \pm .002$	$0.499 \pm .002$	$0.696 \pm .002$	$0.796 \pm .002$
$\alpha_1 = 0, \alpha_2 = 1$	$0.019 \pm .001$	$14.6 \pm .002$	$0.494 \pm .001$	$0.698 \pm .002$	$0.792 \pm .003$
$\alpha_1 = 0, \alpha_2 = 0$	$0.007 \pm .001$	$11.2 \pm .002$	$0.498 \pm .002$	$0.689 \pm .003$	$0.796 \pm .004$
$\alpha_1 = 1, \alpha_2 = 1$	$0.008 \pm .001$	$9.4 \pm .001$	$0.504 \pm .002$	$0.703 \pm .002$	$0.802 \pm .003$

Table 2: **Reconstruction results** of pose velocity and acceleration Loss with different weights.

D.3 Average Inference Time Results Comparison

To provide a clear comparison, in Table 3, we report the efficiency of motion generation in terms of average inference time per sentence (AITS) over 100 samples on a single Nvidia 4090 device. It can be seen that compared with diffusion models, we have a significant speed advantage. When compared with VQ-based models, we offer better synthesis quality under similar inference speeds.

D.4 Comparison of VQ-based AR transformer with GMMotion

Our highlight is the masked autoregressive synthesis using continuous motion representations. Unlike the masking strategy employed by MoMask (1), we apply masking and pseudo-reordering to the

Methods	MDM	MLD	T2M-GPT	MARDM	MoMask	MotionLCM-v2	GMMotion (ours)
Framework	Diffusion	Diffusion	VQ + AR	Diffusion + MAR	VQ + NAR	VAE + Diffusion	GMM + AR
AITS ↓	14.7	0.241	0.319	2.30	0.046	0.045	0.033

Table 3: Average Inference Time Results Comparison between our method and baseline methods.

input latent vectors during the training phase. To further test its effectiveness, we introduced a VQ-based MAR synthesis method. We used MMM (3) as a baseline, changed to masked autoregressive synthesis, and kept the first stage VQ-VAE unchanged. The results are shown in Table 4.

Methods	Framework	R-Precision↑			FID↓	Matching↓	MModality↑	CLIP-score↑
		Top 1	Top 2	Top 3				
T2M-GPT (2)	VQ + NAR	0.470 \pm .003	0.659 \pm .002	0.758 \pm .002	0.335 \pm .003	3.505 \pm .017	2.018 \pm .053	0.607 \pm .005
MMM (3)		0.487 \pm .003	0.683 \pm .002	0.782 \pm .001	0.132 \pm .004	3.359 \pm .009	1.241 \pm .073	0.635 \pm .003
MoMask (1)		0.490 \pm .004	0.687 \pm .003	0.786 \pm .003	0.116 \pm .006	3.353 \pm .010	1.263 \pm .079	0.637 \pm .003
MMM + MAR	VQ + MAR	0.496 \pm .004	0.681 \pm .002	0.784 \pm .002	0.149 \pm .003	3.363 \pm .008	1.251 \pm .082	0.631 \pm .003
GMMotion-s	VAE + MAR	0.508 \pm .002	0.698 \pm .003	0.799 \pm .003	0.123 \pm .004	3.259 \pm .008	2.121 \pm .065	0.643 \pm .002
GMMotion-l		0.504 \pm .004	0.701 \pm .003	0.796 \pm .003	0.103 \pm .006	3.302 \pm .009	2.427 \pm .065	0.653 \pm .002
GMMotion-l w/o GM-VAE	MAR	0.403 \pm .002	0.592 \pm .004	0.679 \pm .002	1.211 \pm .016	3.908 \pm .014	2.633 \pm .054	0.584 \pm .003

Table 4: Quantitative evaluation on *MARDM-HumanML3D benchmark*. We tested the impact of replacing the Gaussian mixture sampling with the vector quantization process in MAR generation.

D.5 Single Stage Model: GMMotion Without GM-VAE

We also attempted to construct a single-stage model, similar to MELLE (6), which is an excellent single-stage text-to-speech synthesis model. Despite verifying that the original motion data follows a multi-modal distribution, we tried direct masked autoregressive generation in the original motion space. However, the final results were unsatisfactory, as shown in the Table 4: **GMMotion-l** w/o GM-VAE. We believe that without employing motion representation modeling with VAE, the motion synthesis results are extremely unstable, leading to noticeable jittering. The uncertainty in the GMM sampling process may cause this instability.

References

- [1] 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.
- [2] 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.
- [3] E. Pinyoanunpong, 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.
- [4] N. Li, S. Liu, Y. Liu, S. Zhao, M. Liu, and M. Zhou, “Neural speech synthesis with transformer network,” 2019. [Online]. Available: <https://arxiv.org/abs/1809.08895>
- [5] 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>
- [6] 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>