Executing your Commands via Motion Diffusion in Latent Space Appendix

https://github.com/chenfengye/motion-latent-diffusion

This appendix provides more qualitative results (Sec. A), several additional experiments (Sec. B) on the components of motion latent diffusion (MLD) models, evaluations of inference time (Sec. C), visualization of latent space (Sec. D), evaluations on hyperparameters (Sec. E), user study (Sec. F), details of motion representations (Sec. G), implementation details of MLD models (Sec. H) and metric definitions (Sec. I).

Video. We have provided supplemental videos in Project Page. In these supplemental videos, we show 1) comparisons of text-based motion generation, 2) comparisons of action-conditional motion generation, and 3) more samples of unconditional generation. We suggest the reader watch this video for dynamic motion results.

Code is available on GitHub Page. We provide the process of the training and evaluation of MLD models, the pre-trained model files, the demo script, and example results.

A. Qualitative Results

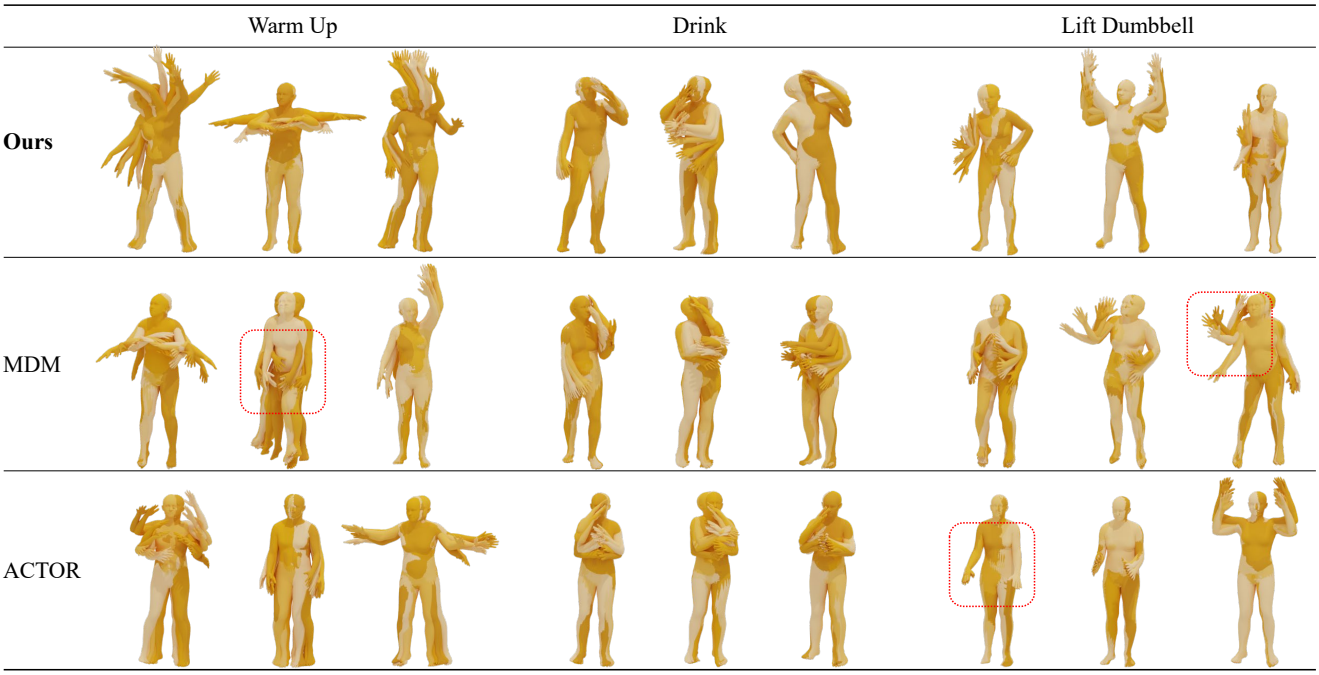

Figure 7. The comparison of the state-of-the-art methods on action-conditional motion synthesis task. All provided methods are under the same training and inference setting on HumanAct12 dataset [10]. We generate three motions for each action label. The results demonstrate that our generations correspond better to the action label and have richer diversity.

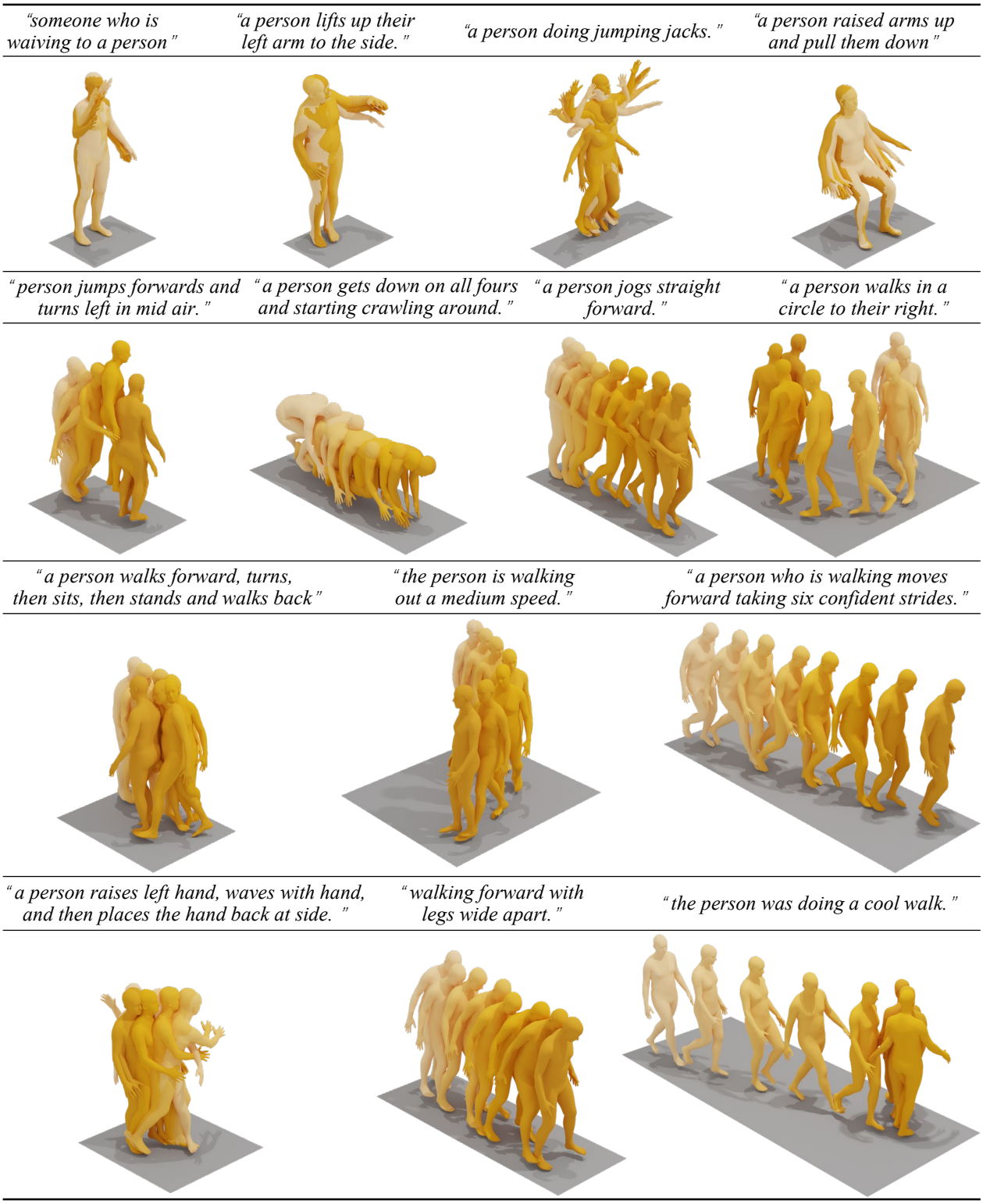

Figure 8. More samples from our best model for text-to-motion synthesis, *MLD-1*, which was trained on the HumanML3D dataset. Samples generated with text prompts of the test set. We recommend the supplemental video to see these motion results.

B. Additional Experiments

We conduct several experiments to continue the evaluations of MLD models. We first study the influence of language models τ_θ^w and the shape of text embedding on motion generations. Then, we evaluate the effectiveness of long skip connections for motion diffusion models. We finally study the importance of regularization on motion latent space.

B.1. Evaluation of Language Models τ_θ^w

We experiment with different language models, CLIP [52] and BERT [10]. Inspired by Stable Diffusion [56], we leverage the hidden state of CLIP to generate word-wise tokens and explore its effects. The comparisons are listed in Tab. 7. CLIP is more suited to our task compared to BERT, and the word-wise text tokens are competitive with the single token, however, lower the computation efficiency of diffusion models. Therefore, we choose CLIP and a single text token for our models.

Models	Text Encoder τ_θ^w	Embeddings Shape	R Precision Top 3↑	FID↓	MM Dist↓	Diversity→	MModality↑
Real	-	-	$0.797^{\pm.002}$	$0.002^{\pm.000}$	$2.974^{\pm.008}$	$9.503^{\pm.065}$	-
MLD-1	BERT [10]	1×256	$0.725^{\pm.002}$	$0.553^{\pm.020}$	$3.530^{\pm.011}$	$\mathbf{9.697}^{\pm.080}$	$\mathbf{3.360}^{\pm.118}$
MLD-1	CLIP [52]	1×256	$\mathbf{0.769}^{\pm.002}$	$0.446^{\pm.011}$	$\mathbf{3.227}^{\pm.010}$	$9.772^{\pm.071}$	$2.413^{\pm.072}$
MLD-1	CLIP [52]	77×256	$0.737^{\pm.002}$	$\mathbf{0.422}^{\pm.012}$	$3.436^{\pm.010}$	$9.840^{\pm.082}$	$2.799^{\pm.107}$

Table 7. Quantitative comparison of the employed language models. Here we set batch size to 500 and only change the text encoder τ_θ^w.

B.2. Effectiveness of Long Skip Connection

We have demonstrated the effectiveness of skip connection, especially on diffusion models in Tab. 5. Here we analyze its influence on the training of diffusion stage. As shown in Fig. 9, the model with long skip connection not only achieves higher performance but also provides faster convergence compared to the other one. The results suggest leveraging long skip connections for iterative motion diffusion models.

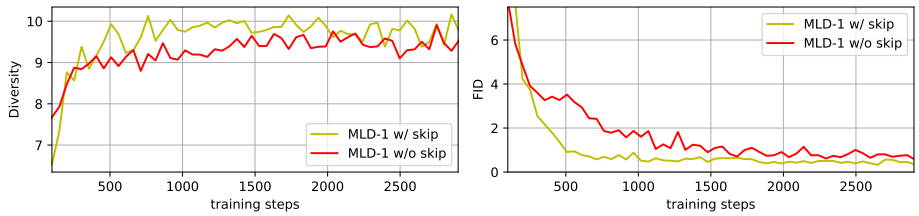

Figure 9. The evaluation on long skip connection on diffusion training stage. Two sub-figures are under the same training process and evaluated on the test set of HumanML3D. Training steps indicate the epoch number.

B.3. Diffusion on Autoencoder or VAE

We study the importance of regularization on motion latent space. The regularized latent space provides stronger generation ability and supports the latent diffusion models as demonstrated:

Method	Reconstruction			Generation	
	MPJPE ↓	PAMPJPE↓	ACCL↓	FID↓	DIV→
Autoencoder	38.5	28.2	5.8	0.156	9.628
VAE	**14.7**	**8.9**	**5.1**	**0.017**	**9.554**

Method	R Precision Top 3↑	FID↓	MM Dist↓	Diversity→	MModality↑
MLD w/ Autoencoder	$0.581^{\pm.003}$	$5.033^{\pm.061}$	$4.600^{\pm.018}$	$7.953^{\pm.083}$	$\mathbf{3.754}^{\pm.111}$
MLD w/ VAE	$\mathbf{0.772}^{\pm.002}$	$\mathbf{0.473}^{\pm.013}$	$\mathbf{3.196}^{\pm.010}$	$\mathbf{9.724}^{\pm.082}$	$2.413^{\pm.079}$

Table 8. Evaluation of autoencoder (without Kullback-Leibler regularization) and VAE model on motion generations.

B.4. Prediction of Denoising

We compare predicting the denoised latent vector z_0 directly instead of ϵ in the denoising process. Tab. 9 shows that the latter performs better, which verifies the proposal from DDPM [23].

Methods	R Precision ↑			FID↓	MM Dist↓	Diversity→	MModality↑
	Top 1	Top 2	Top 3				
MLD-1 (z_0)	$0.447^{\pm.002}$	$0.633^{\pm.002}$	$0.734^{\pm.002}$	$0.513^{\pm.011}$	$3.384^{\pm.008}$	$9.181^{\pm.065}$	$0.735^{\pm.055}$
MLD-1 (ϵ)	$\mathbf{0.481}^{\pm.003}$	$\mathbf{0.673}^{\pm.003}$	$\mathbf{0.772}^{\pm.002}$	$\mathbf{0.473}^{\pm.013}$	$\mathbf{3.196}^{\pm.010}$	$\mathbf{9.724}^{\pm.082}$	$\mathbf{2.413}^{\pm.079}$

Table 9. Comparison of text-to-motion. (cf. Tab. 1 for details.)

C. Inference time

We provide a detailed ablation study with DDIM below. In Tab. 10, MLD reduces the computational cost of diffusion models, which is the main reason for faster inference. The iterations of diffusion further widen the gap in computational cost. Please note the bad FID of MDM with DDIM is mentioned in their GitHub issues #76.

Methods	Total Inference Time (s) ↓				FLOPs (G) ↓				Parameter	FID ↓			
	DDIM			DDPM	DDIM			DDPM		DDIM			DDPM
	50	100	200	1000	50	100	200	1000		50	100	200	1000
MDM	225.28	456.70	911.36	4546.23	597.97	1195.94	2391.89	11959.44	$x \in \mathbb{R}^{196\times512}$	7.334	5.990	5.936	0.544
MLD	**10.24**	**16.38**	**28.67**	**148.97**	**29.86**	**33.12**	**39.61**	**91.60**	$z \in \mathbb{R}^{1\times256}$	0.473	0.426	0.432	0.568

Table 10. Evaluation of inference time costs on text-to-motion: we evaluate the total inference time to generate 2048 motion clips with different diffusion schedules, floating point operations (FLOPs) counted by THOP library, the size of diffusion input, and FID.

D. Latent space visualization

We provide the visualizations of the t-SNE results on the latent space in Fig. 10 to demonstrate how latent space evolves during the diffusion process with different actions. From left to right, it shows the evolved latent codes during the inference of diffusion models.

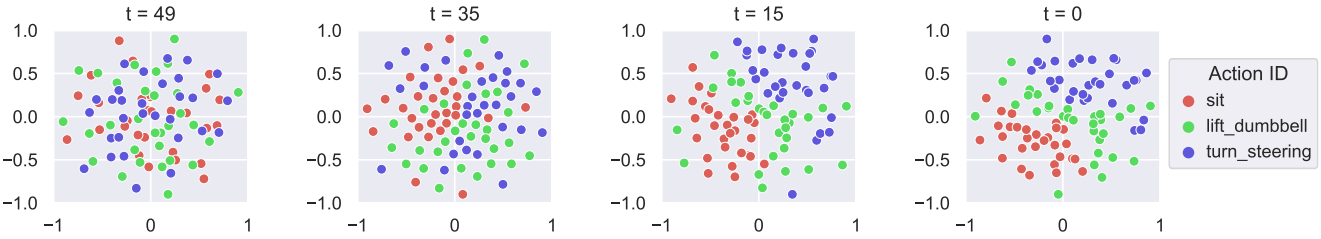

Figure 10. Visualization of the t-SNE results on evolved latent codes $\hat{z}_t$ during the reverse diffusion process (inference) on action-to-motion task. t is the diffusion step but ordered in the forward diffusion trajectory. $\hat{z}_{t=49}$ is the initial random noise. $\hat{z}_{t=0}$ is our prediction. We sample 30 motions for each action label.

E. Evaluation of Hyperparameters

Here, we present two different experiments of text-to-motion on HumanML3D [17]. The first experiment is to change the dropout p and scale in classifier-free diffusion guidance [24]. In Tab. 11, we find that by changing dropout p from 0.1 to 0.25, the text correspondences (R Precision) become worse but the motion quality (FID) gets better. It is the same as changing scale s range from 7.5 to 2.5. Besides, some settings like $(0.25, 7.5)$ achieve the best FID of 0.229, but we still suggest $(0.1, 7.5)$ as dropout and scale (p, s) for MLD models (Sec. 4) overall metrics.

Next, in Tab. 12, we experiment with batch sizes of 32, 64, 128, 256 and 512 under 8 Tesla V100 each with 32 GPU memory. We set it to 64 in our other experiments.

Models	Classifier-free		R Precision	FID↓	MM Dist↓	Diversity→	MModality↑
	Dropout	Scale	Top 3↑				
Real	-	-	$0.797^{\pm.002}$	$0.002^{\pm.000}$	$2.974^{\pm.008}$	$9.503^{\pm.065}$	-
MLD-1	$p = 0.05$	$s = 7.5$	$0.766^{\pm.002}$	$0.574^{\pm.013}$	$3.237^{\pm.007}$	$9.664^{\pm.069}$	$2.433^{\pm.074}$
MLD-1	$p = 0.10$	$s = 7.5$	$\mathbf{0.772}^{\pm.002}$	$0.473^{\pm.013}$	$\mathbf{3.196}^{\pm.010}$	$9.724^{\pm.082}$	$2.413^{\pm.079}$
MLD-1	$p = 0.15$	$s = 7.5$	$0.765^{\pm.002}$	$0.311^{\pm.009}$	$3.209^{\pm.007}$	$9.649^{\pm.065}$	$2.525^{\pm.070}$
MLD-1	$p = 0.20$	$s = 7.5$	$0.761^{\pm.002}$	$0.279^{\pm.011}$	$3.243^{\pm.009}$	$\mathbf{9.632}^{\pm.082}$	$2.651^{\pm.080}$
MLD-1	$p = 0.25$	$s = 7.5$	$0.757^{\pm.002}$	$\mathbf{0.229}^{\pm.010}$	$3.260^{\pm.008}$	$9.649^{\pm.069}$	$\mathbf{2.685}^{\pm.084}$
MLD-1	$p = 0.30$	$s = 7.5$	$0.759^{\pm.002}$	$0.289^{\pm.010}$	$3.249^{\pm.008}$	$9.670^{\pm.073}$	$2.650^{\pm.082}$
MLD-1	$p = 0.10$	$s = 1.5$	$0.648^{\pm.002}$	$0.401^{\pm.019}$	$3.857^{\pm.009}$	$9.263^{\pm.056}$	$\mathbf{3.914}^{\pm.115}$
MLD-1	$p = 0.10$	$s = 2.5$	$0.720^{\pm.002}$	$\mathbf{0.350}^{\pm.013}$	$3.441^{\pm.010}$	$\mathbf{9.549}^{\pm.058}$	$3.201^{\pm.098}$
MLD-1	$p = 0.10$	$s = 3.5$	$0.745^{\pm.002}$	$0.358^{\pm.011}$	$3.299^{\pm.009}$	$9.639^{\pm.065}$	$2.890^{\pm.087}$
MLD-1	$p = 0.10$	$s = 4.5$	$0.758^{\pm.002}$	$0.375^{\pm.011}$	$3.232^{\pm.009}$	$9.676^{\pm.069}$	$2.701^{\pm.078}$
MLD-1	$p = 0.10$	$s = 5.5$	$0.764^{\pm.002}$	$0.396^{\pm.011}$	$3.202^{\pm.010}$	$9.681^{\pm.072}$	$2.577^{\pm.076}$
MLD-1	$p = 0.10$	$s = 6.5$	$0.767^{\pm.002}$	$0.424^{\pm.011}$	$\mathbf{3.191}^{\pm.009}$	$9.658^{\pm.072}$	$2.498^{\pm.074}$
MLD-1	$p = 0.10$	$s = 7.5$	$\mathbf{0.772}^{\pm.002}$	$0.473^{\pm.013}$	$3.196^{\pm.010}$	$9.724^{\pm.082}$	$2.413^{\pm.079}$
MLD-1	$p = 0.10$	$s = 8.5$	$0.768^{\pm.002}$	$0.504^{\pm.012}$	$3.207^{\pm.009}$	$9.604^{\pm.073}$	$2.413^{\pm.072}$
MLD-1	$p = 0.10$	$s = 9.5$	$0.766^{\pm.001}$	$0.555^{\pm.012}$	$3.227^{\pm.010}$	$9.567^{\pm.072}$	$2.394^{\pm.069}$

Table 11. **Classifier-free Diffusion Guidance:** We study the influence of its hyperparameters, dropout p and scale s on text-to-motion.

Models	Batch Size	R Precision Top 3↑	FID↓	MM Dist↓	Diversity→	MModality↑
Real	-	$0.797^{\pm.002}$	$0.002^{\pm.000}$	$2.974^{\pm.008}$	$9.503^{\pm.065}$	-
MLD-1	32	$0.761^{\pm.003}$	$0.445^{\pm.012}$	$3.243^{\pm.010}$	$9.751^{\pm.086}$	$\mathbf{2.581}^{\pm.070}$
MLD-1	64	$\mathbf{0.772}^{\pm.002}$	$0.473^{\pm.013}$	$\mathbf{3.196}^{\pm.010}$	$9.724^{\pm.082}$	$2.413^{\pm.079}$
MLD-1	128	$0.771^{\pm.002}$	$\mathbf{0.421}^{\pm.013}$	$3.187^{\pm.008}$	$\mathbf{9.691}^{\pm.080}$	$2.370^{\pm.078}$
MLD-1	256	$0.770^{\pm.002}$	$0.423^{\pm.010}$	$3.211^{\pm.007}$	$9.800^{\pm.070}$	$2.401^{\pm.074}$
MLD-1	512	$0.769^{\pm.002}$	$0.446^{\pm.011}$	$3.227^{\pm.010}$	$9.772^{\pm.071}$	$2.413^{\pm.072}$

Table 12. **Batch Size:** We explore the evaluation of the batch size. We find the results are close and suggest 64 and 128 in this task.

F. User Study

For the pairwise comparisons of the user study presented in Fig. 11, we use the force-choice paradigm to ask "Which of the two motions is more realistic?" and "which of the two motions corresponds better to the text prompt?". The provided motions are generated from 30 text descriptions, which are randomly generated from the test set of HumanML3D [17] dataset. We invite 20 users and provide three comparisons, ours and MDM [69], ours and T2M [16], ours and real motions from the dataset. Our MLD was preferred over the other state-of-the-art methods and even competitive to the ground troth motions.

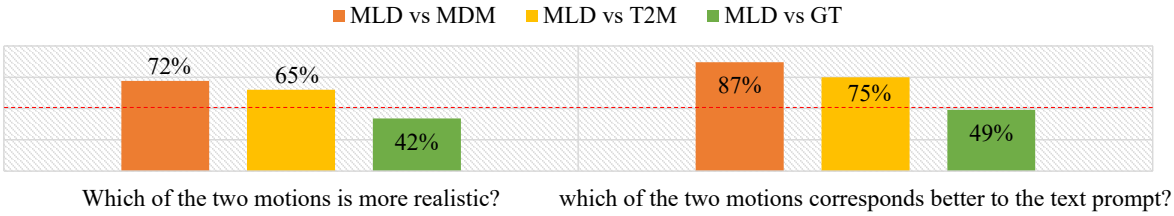

Figure 11. **User Study:** Each bar indicates the preference rate of MLD over other methods. The red line indicates the 50%. Please refer to the supplemental video for the comparisons of dynamic motion results.

G. Motion Representations

Four relevant motion representations are summarized:

HumanML3D Format [17] proposes a motion representation $x^{1:L}$ inspired by motion features in character control [66, 45, 65]. This redundant representation is quite suited to neural models, particularly variational autoencoders. Specifically, the i-th pose x^i is defined by a tuple of root angular velocity $\dot{r}^a \in \mathbb{R}$ along Y-axis, root linear velocities $(\dot{r}^x, \dot{r}^z \in \mathbb{R})$ on XZ-plane, root height $r^y \in \mathbb{R}$, local joints positions $\mathbf{j}^p \in \mathbb{R}^{3N_j}$, velocities $\mathbf{j}^v \in \mathbb{R}^{3N_j}$ and rotations $\mathbf{j}^r \in \mathbb{R}^{6N_j}$ in root space, and binary foot-ground contact features $\mathbf{c}^f \in \mathbb{R}^4$ by thresholding the heel and toe joint velocities, where N_j denotes the joint number, giving:

$$x^i = \{\dot{r}^a, \dot{r}^x, \dot{r}^z, r^y, \mathbf{j}^p, \mathbf{j}^v, \mathbf{j}^r, \mathbf{c}^f\}. \tag{5}$$

SMPL-based Format [40]. The most popular parametric human model, SMPL [40] and its variants [57, 44] propose motion parameters θ and shape parameters β. $\theta \in \mathbb{R}^{3\times 23+3}$ is rotation vectors for 23 joints and a root, and β are the weights for linear blended shapes. This representation is popular in markerless motion capture [21, 8, 31]. By involving the global translation r, the representation is formulated as:

$$x^i = \{r, \theta, \beta\}. \tag{6}$$

MMM Format [67]. Master Motor Map (MMM) representations propose joints angle parameters by adopting a uniform skeleton structure with 50 DoFs. And most recent methods [2, 13, 47] on text-to-motion task followed preprocess procedure in [25] which transform joint rotation angles to $J = 21$ joints XYZ coordinates, giving $p_m \in \mathbb{R}^{3J}$, and global trajectory t_{root} for the root joint. The preprocessed representation can be formulated as

$$x^i = \{p_m, t_{root}\}. \tag{7}$$

Latent Format [40]. Latent representations are widely used in neural models [46, 47, 19, 9]. We recognize it as motion representation in latent space. By leveraging VAE models, latent vectors can represent plausible motions as:

$$\hat{x}^{1:L} = \mathcal{D}(z), z = \mathcal{E}(x^{1:L}) \tag{8}$$

H. Details on Motion Latent Diffusion Models

H.1. Details Information on Variational Autoencoder Models

We take HumanML3D [17] and its motion representation (Sec. G) as an example here to explain our loss details of Variational Autoencoder Models $\mathcal{V}$. The motion $x^{1:L}$ includes joint features and is supervised with data term by mean squared error:

$$\mathcal{L}_{data} = \left\| x^{1:L} - \mathcal{D}(\mathcal{E}(x^{1:L})) \right\|^2. \tag{9}$$

To regularize latent space as a standard variational autoencoder [30], we employ a Kullback-Leibler term between $q(z|x^{1:L}) = \mathcal{N}(z; \mathcal{E}_\mu, \mathcal{E}_{\sigma^2})$ and a standard Gaussion distribution $\mathcal{N}(z; 0, 1)$. The full training loss of the VAE model $\mathcal{V}$ follows:

$$\mathcal{L}_{\mathcal{V}} = \mathcal{L}_{data}(x^{1:L}, \mathcal{D}(\mathcal{E}(x^{1:L}))) + \lambda_{reg}\mathcal{L}_{reg}(x^{1:L}; \mathcal{E}, \mathcal{D}), \tag{10}$$

where λ_{reg} is a low weight to control the regularization. The KIT [48], HumanAct12 [19] and UESTC [26] dataset processed by [47, 46] also supports SMPL-based [40] motion representation. Here we list the loss terms for this representation. The data term formulates as followed:

$$\mathcal{L}_{data} = \sum_{i=1}^{L} \left\| r^i - \hat{r}^i \right\|_2 + \sum_{i=1}^{L} \left\| \theta^i - \hat{\theta}^i \right\|_2 + \left\| \beta - \hat{\beta} \right\|_2. \tag{11}$$

Here the motion is $x^{1:L} = \{r^i, \theta^i, \beta\}_{i=1}^L$, which includes global translation r^i, pose parameter θ^i and shape parameter β of the i-th frame. To enhance the full-body supervision, the reconstruction term on the SMPL vertices follows:

$$\mathcal{L}_{\text{mesh}} = \sum_{i=1}^{L} \left\| V_i - M(\hat{r}^i, \hat{\theta}^i, \hat{\beta}^i) \right\|^2, \tag{12}$$

where the body reconstruction function $M(\cdot)$ is from the differentiable SMPL layer, while the vertices V_i are calculated with the ground truth motion parameters using the same layer. The reconstruction loss builds global supervision on almost all predicted parameters $\{r_t, \theta_t, \beta\}$ and shows a reliable supervision [46] for motion generation. The full objective on SMPL-based motion representation reads:

$$\mathcal{L}_{\mathcal{V}} = \mathcal{L}_{data} + \lambda_{mesh}\mathcal{L}_{mesh} + \lambda_{reg}\mathcal{L}_{reg}. \tag{13}$$

where λ_{mesh} is the weight to enhance the supervision on the full-body vertices. Besides, the regularization term is the same as the Kullback-Leibler term in Eq. (11). In practice, the shape parameters, as part of global motion features, increase the complexity of motion generation and influence joint positions. We finally utilize the objective of Eq. (11) to train our text-based models and Eq. (13) to train action-based models in comparisons and evaluations.

H.2. Network Architectures

The details of network architecture are shown as Fig. 12, our MLD comprises three main components, motion encoder $\mathcal{E}$, motion decoder $\mathcal{D}$ and latent denoiser ϵ_θ. Please refer to the following figure and Tab. 13 for more details.

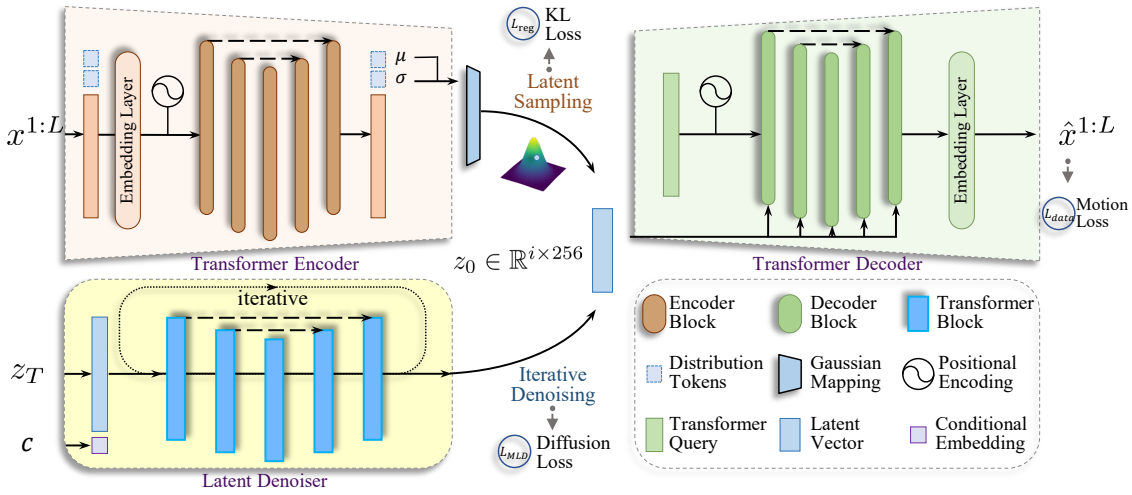

Figure 12. Network architecture of our conditional MLD. We explain each component in its right bottom part and the loss terms in Sec. H.1.

H.3. Implementation Details

For the experiments on text-to-motion, action-to-motion, and unconditional motion synthesis, we implement MLDs with various latent shapes as follows. Specifically, MLD-7 works best in evaluating VAE models (Tab. 4), and MLD-1 wins these generation tasks (Tabs. 1, 2, 3 and 6). In other words, MLD-7 wins the first training stage for the VAE part, while MLD-1 wins the second for the diffusion part. We thought MLD-7 should perform better than MLD-1 in several tasks, but the results differ. The main reason for this downgrade of a larger latent size, we believe, is the small amount of training data. HumanML3D only includes 15k motion sequences, much smaller than billions of images in image generation. MLD-7 could work better when the motion data amount reaches the million level.

	MLD-1	MLD-2	MLD-5	MLD-7	MLD-10
z-shape	1×256	2×256	5×256	7×256	10×256
Training Diffusion steps	1000	1000	1000	1000	1000
Inference Diffusion steps	50	50	50	50	50
Noise Schedule	scaled linear	scaled linear	scaled linear	scaled linear	scaled linear
Denoiser Heads Number	4	4	4	4	4
Denoiser Transformer Layers	9	9	9	9	9
Conditioning	concat	concat	concat	concat	concat
Embedding Dimension	256	256	256	256	256
VAE Heads Number	4	4	4	4	4
VAE Transformer Layers	9	9	9	9	9
Model Size (w/o clip)	26.9M	26.9M	26.9M	26.9M	26.9M
Diffusino Batch Size	64	64	64	64	64
Diffusion Epochs	2000	2200	2400	2600	2800
VAE Batch Size	128	128	128	128	128
VAE Epochs	4000	4500	5000	5500	6000
Learning Rate	1e-4	1e-4	1e-4	1e-4	1e-4

Table 13. Hyperparameters for the conditional MLDs in experiments. We train these models on 8 Tesla V100. The smaller latent shape lowers the computational requirements and provides faster inference.

I. Metric Definitions

We provide more details of evaluation metrics in Sec. 4.1 as follows.

Motion Quality. Frechet Inception Distance (FID) is our principal metric to evaluate the distribution similarity between generated and real motions, calculated with the suitable feature extractor [19, 46, 16] for each dataset. Besides, to evaluate the motion reconstruction error of VAEs, we use popular metrics in motion capture [31, 8, 71], MPJPE, and PAMPJPE [14] for global and local errors in millimeter, Acceleration Error (ACCL) for the quality on temporal.

Generation Diversity. Following [19, 18], we use Diversity (DIV) and MultiModality (MM) to measure the motion variance across the whole set and the generated motion diversity within each text input separately. Here we take the text-to-motion task as an example to explain the calculation steps and for other tasks the operations are similar. To evaluate Diversity, all generated motions are randomly sampled to two subsets of the same size X_d with motion feature vectors $\{x_1, .., x_{X_d}\}$ and $\{x_1^{'}, .., x_{X_d}^{'}\}$ respectively. Then diversity is formalized as:

$$\text{DIV} = \frac{1}{X_d} \sum_{i=1}^{X_d} ||x_i - x_i^{'}||.$$

To evaluate MultiModality, a set of text descriptions with size J_m is randomly sampled from all descriptions. Then two subsets of the same size X_m are randomly sampled from all motions generated by j-th text descriptions, with motion feature vectors $\{x_{j,1}, .., x_{j,X_m}\}$ and $\{x_{j,1}^{'}, .., x_{j,X_m}^{'}\}$ respectively. The multimodality is calculated as:

$$\text{MM} = \frac{1}{J_m \times X_m} \sum_{j=1}^{J_m} \sum_{i=1}^{X_m} ||x_{j,i} - x_{j,i}^{'}||.$$

Condition Matching. For the text-to-motion task, [16] provides motion/text feature extractors to produce geometrically closed features for matched text-motion pairs, and vice versa. Under this feature space, motion-retrieval precision (R Precision) first mix generated motion with 31 mismatched motions and then calculates the text-motion top-1/2/3 matching accuracy, and Multi-modal Distance (MM Dist) that calculates the distance between generated motions and text. For action-to-motion, for each dataset a pretrained recognition model [19, 46] is used to calculate the average motion Accuracy (ACC) for action categories.

Time Costs. To evaluate the computing efficiency of diffusion models, especially the inference efficiency, we propose Average Inference Time per Sentence (AITS) measured in seconds. In our case, we calculate AITS (cf. Fig. 6) on the test set of HumanML3D [17], set the batch size to one, and ignore the time cost for model and dataset loading parts.