EMDM: Efficient Motion Diffusion Model for Fast and High-Quality Motion Generation **Supplementary Material**

Wenyang Zhou[1], Zhiyang Dou[2,3,4,†,⋆], Zeyu Cao[1], Zhouyingcheng Liao[2], Jingbo Wang[5], Wenjia Wang[2], Yuan Liu[2], Taku Komura[2], Wenping Wang[6], and Lingjie Liu[3]

[1] University of Cambridge
[2] The University of Hong Kong
[3] University of Pennsylvania
[4] TransGP
[5] Shanghai AI Laboratory
[6] Texas A&M University
[†] *Project Lead.*

This supplementary material covers: More Qualitative Results (Sec. A); Unconditional Motion Generation (Sec. B); Implementation Details (Sec. C) and More Experimental Results (Sec. D). Please watch our supplementary video for a more thorough review.

A More Qualitative Results

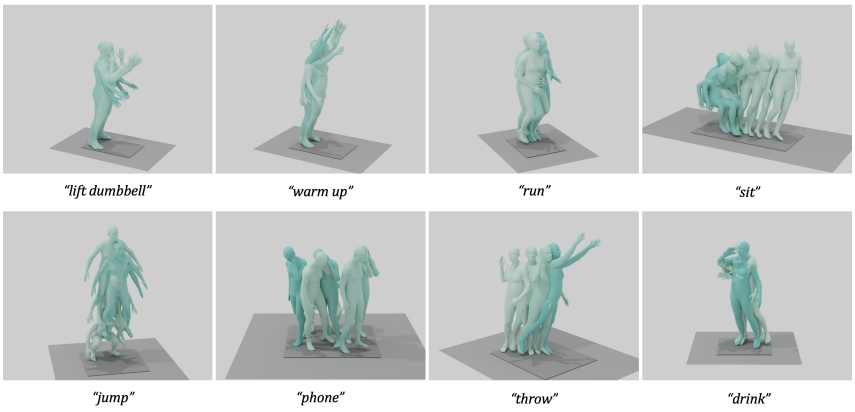


Fig. A1: More qualitative results of EMDM on the task of action-to-motion.

⋆ Collaborating with Tencent Games.

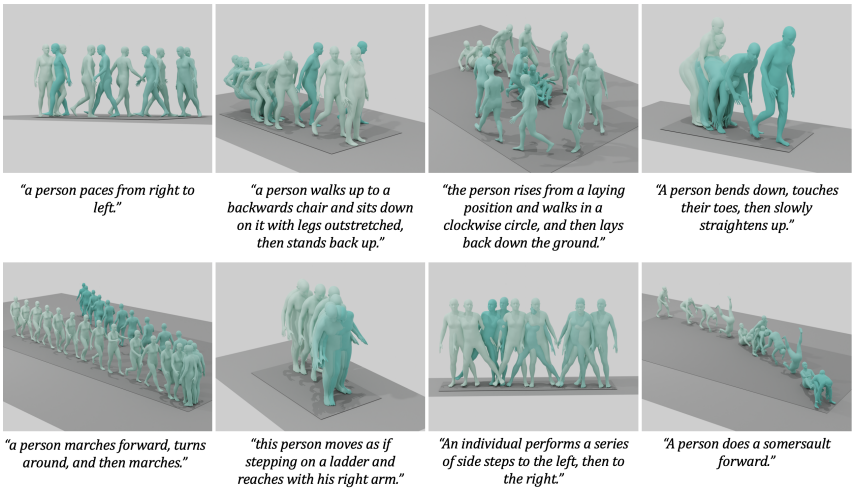


Fig. A2: More qualitative results of EMDM on the task of text-to-motion.

In the following, we provide more qualitative results of action-to-motion and text-to-motion tasks, which are visualized in Fig. A1 and Fig. A2. The model is evaluated on the HumanAct12 [3] dataset and HumanML3D dataset [2], respectively. EMDM produces high-quality human motions that faithfully align with the input conditions. We highly suggest readers watch our supplementary video for a more thorough review.

B Unconditional Motion Generation

Next, we evaluate unconditional motion generation following [1]. As shown in Tab. B1, EMDM exhibits higher motion quality and significantly reduced running time when compared to existing methods.

Table B1: Comparison of unconditional motion generation task on the part of AMASS dataset following [1].

Methods	FID ↓	Diversity→	Running Time (per frame; ms)↓
Real	0.002	9.503	-
VPoser-t [6]	36.65	3.259	-
ACTOR [7]	14.14	5.123	$0.523^{\pm.009}$
MDM [9]	8.84	6.429	$62.505^{\pm.071}$
MLD [1] †	**1.4**	**8.577**	**0.886**$^{\pm.007}$
EMDM (Ours)	**3.46**	**8.759**	**0.280**$^{\pm.002}$

Blue and **orange** indicate the best and the second best result.
† **Two-stage and non end-to-end approach.**

C Implementation Details

In the following, we present the network structures and training details of EMDM. During the training stage, we noise a ground-truth image $\mathbf{x}_0$ to $\mathbf{x}_{t-1}$ and $\mathbf{x}_t$ given a time step t. We use the $\mathbf{x}_t$, as well as conditions (text/action $\mathbf{c}$, time step t) and latent variable $\mathbf{z}$ to generate $\hat{\mathbf{x}}_0$ which is then used to sample $\hat{\mathbf{x}}_{t-1}$. The fake $\hat{\mathbf{x}}_{t-1}$ or real $\mathbf{x}_{t-1}$, together with conditions (text/action $\mathbf{c}$, time step t, and the real $\mathbf{x}_t$), are fed to the conditional discriminator. During inference, conditions (including text/action $\mathbf{c}$, time step t, and $\mathbf{x}_t$) and a latent variable $\mathbf{z}$ are fed to our generator. The denoised output is the generated motion.

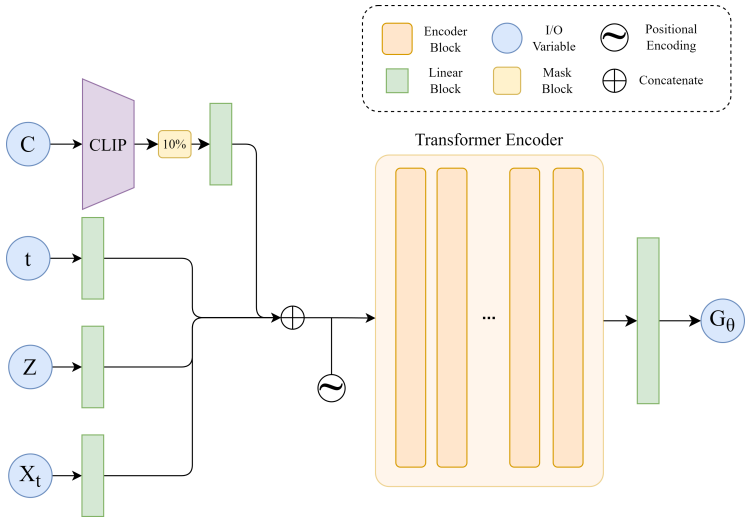


Fig. C3: The generator architecture for the text-to-motion tasks. For the action-to-motion task, the *CLIP* module, masking module, and the corresponding linear layer are replaced with a single linear layer for action label embedding. The linear layer for $\mathbf{z}$ consists of 5 layers.

C.1 Conditional Generator Structure

In this paper, we employ a conditional generator for synthesizing motion conditioned on text or action labels, time step t and human motion $\mathbf{x}_t$ at t-th time step. The model can be written as $G_\theta(\mathbf{x}_t, \mathbf{z}, \mathbf{c}, t)$, where $\mathbf{x}_t$ is the motion to be denoised, $\mathbf{z} \in \mathbb{R}^{64}$ is the latent variable for GAN, and $\mathbf{c}$, either a string of text or an action number $\in \mathbb{R}^1$, is the input control signal. The network structure of G_θ is shown as in Fig. C3.

T2M Architecture t and $\mathbf{z}$ are mapped to $\mathbb{R}^{1024}$ by 1 and 5 linear layers respectively, while $\mathbf{c}$ is encoded by *CLIP* [8] to $\mathbb{R}^{512}$, randomly masked 10% of the

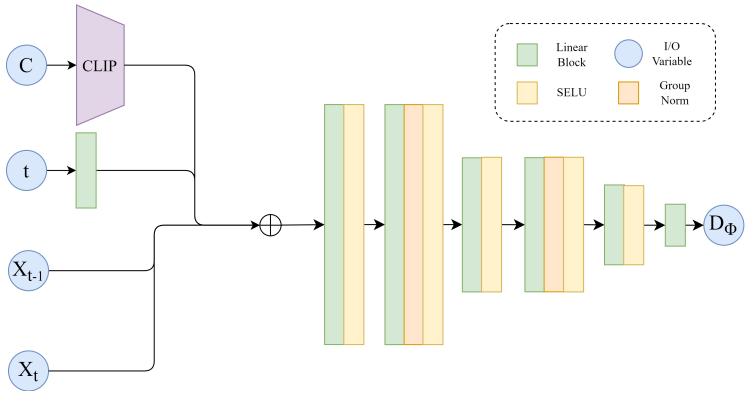


Fig. C4: The discriminator architecture for the text-to-motion task. We replace the *CLIP* module with one-hot encoding for the action-to-motion task.

values and embedded to $\mathbb{R}^{1024}$ by a linear layer. $\mathbf{x}_t$ is mapped to $\mathbb{R}^{seq\times 1024}$ by a linear layer, where *seq* is the length of the motion. All the aforementioned values are concatenated and fed to the encoder. We discard the first 3 tokens of the output and map it back to a motion using a linear layer.

We use the PyTorch implementation for Transformers. The model has 12 transformer layers and 32 attention heads. The feed-forward size and latent dimension are both set to be 1024. The dropout rate is 0.1. We employ *selu* as the activation function.

A2M Architecture The overall architecture is the same. The only difference for action-to-motion tasks is that instead of using *CLIP* + masking + linear layer to map a text to $\mathbb{R}^{1024}$, we use a linear layer to map the action number directly to $\mathbb{R}^{1024}$.

C.2 Conditional Discriminator Structure

In EMDM, we employ a conditional discriminator for assessing the authenticity of motions, which can be written as $D_\phi(\mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{c}, t)$, where $\mathbf{x}_{t-1}$ is the motion to be assessed. The input control signals $\mathbf{c}$ (either a string of text or an action number $\in \mathbb{R}^1$), time step t and $\mathbf{x}_t$ serve as the conditions for D_ϕ. The network structure is shown in Figure C4.

After training, the discriminator would give a positive value for real motions $\mathbf{x}_{t-1}$ and a negative value for the fake ones $\hat{\mathbf{x}}_{t-1}$.

T2M Architecture The Discriminator consists of 7 linear layers, each followed by a *selu* layer. Group normalization is applied after two of the linear layers as well. t is embedded to $\mathbb{R}^{128}$ with sinusoidal positional embeddings as similar to [10]. $\mathbf{c}$ is embedded to $\mathbb{R}^{512}$ using $CLIP$. We then concatenate $\mathbf{x}_t$, $\mathbf{x}_{t-1}$, embedded t and embedded $\mathbf{c}$ and pass the result to the linear layers.

A2M Architecture The overall architecture is the same. The only difference for action-to-motion tasks is that instead of using $CLIP$ to embed the text, we use one-hot encoding to transform the action number from $\mathbb{R}^1$ to $\mathbb{R}^A$, where A is the number of possible action labels.

C.3 Training Details

During network training, we adopt the scheduling scheme following [10]. During each iteration, we first train the discriminator with objective

$$\min_{\phi} \sum_{t\geq 1} (\mathbb{E}_{q(\mathbf{x}_0)q(\mathbf{x}_{t-1}|\mathbf{x}_0)q(\mathbf{x}_t|\mathbf{x}_{t-1})}[\mathrm{F}(-D_\phi(\mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{c}, t))] + \mathbb{E}_{q(\mathbf{x}_t)}\mathbb{E}_{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}[\mathrm{F}(D_\phi(\mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{c}, t))]). \tag{1}$$

Then we train the generator with objective

$$\min_{\theta} \sum_{t\geq 1} (\mathbb{E}_{q(\mathbf{x}_t)}\mathbb{E}_{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}[\mathrm{F}(-D_\phi(\mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{c}, t))] + R \cdot \mathcal{L}_{\text{geo}}), \tag{2}$$

where $\mathrm{F}(\cdot)$ denotes the softplus$(\cdot)$ function and $\mathcal{L}_{\text{geo}} = \mathcal{L}_{\text{recon}} + \lambda(\mathcal{L}_{\text{pos}} + \mathcal{L}_{\text{vel}} + \mathcal{L}_{\text{foot}})$, as stated in the main paper.

Similar to [10] we add an R_1 regularization term [5] to the loss term of the discriminator:

$$\frac{\gamma}{2}\mathbb{E}_{q(\mathbf{x}_0)q(\mathbf{x}_{t-1}|\mathbf{x}_0)q(\mathbf{x}_t|\mathbf{x}_{t-1})}[\|\nabla_{x_{t-1}} D_\phi(\mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{c}, t)\|^2]. \tag{3}$$

In this paper, we use $\gamma = 0.02$ for all tasks.

We train our model using the Adam optimizer [4] with cosine learning rate decay [10]. The exponential moving average (EMA) is used during the training of the generator. The batch size is 64 for all tasks.

The learning rate of the conditional discriminator is 1.25×10^{-4}. For the generator, we use a learning rate of 3×10^{-5} and 2×10^{-5} for action-to-motion and text-to-motion tasks, respectively.

D More Experiments

D.1 Comparisons with DDIM Sampling Methods

Moreover, in Tab. D2, we compare EMDM with other few-step sampling diffusion models for motion generation [1,9,12]. To be specific, we show that accelerating sampling by naively reducing the sampling step size using DDIM (10 steps) leads to quality degradation due to the inaccurate approximation of complex data distributions as analyzed in Sec. 3 of the main paper. This holds true for both motion diffusion models [9,12] or the motion latent diffusion models [1].

Table D2: Comparison with motion diffusion models with few-step sampling (10 sampling steps) on Text-to-motion. We test on HumanML3D.

Methods	R Precision ↑			FID↓	MM Dist↓	Diversity→	MModality↑	Running Time (per frame; ms)↓
	Top 1	Top 2	Top 3					
Real	$0.511^{\pm.003}$	$0.703^{\pm.003}$	$0.797^{\pm.002}$	$0.002^{\pm.000}$	$2.974^{\pm.008}$	$9.503^{\pm.065}$	-	-
MotionDiffuse	$0.040^{\pm.005}$	$0.074^{\pm.006}$	$0.108^{\pm.008}$	$100.780^{\pm.619}$	$12.434^{\pm.052}$	$10.943^{\pm.106}$	$6.650^{\pm.273}$	$1.426^{\pm.030}$
MDM	$0.076^{\pm.062}$	$0.139^{\pm.004}$	$0.194^{\pm.007}$	$33.232^{\pm.308}$	$7.165^{\pm.048}$	$3.440^{\pm.060}$	**$2.325^{\pm.023}$**	$0.673^{\pm.001}$
MLD†	**$0.480^{\pm.003}$**	**$0.670^{\pm.003}$**	**$0.769^{\pm.003}$**	**$0.397^{\pm.009}$**	**$3.199^{\pm.010}$**	**$9.923^{\pm.075}$**	**$2.488^{\pm.094}$**	**$0.359^{\pm.002}$**
EMDM (Ours)	**$0.498^{\pm.007}$**	**$0.684^{\pm.006}$**	**$0.786^{\pm.006}$**	**$0.112^{\pm.019}$**	**$3.110^{\pm.027}$**	**$9.551^{\pm.078}$**	$1.641^{\pm.078}$	**$0.280^{\pm.002}$**

Blue and **orange** indicate the best and the second best result.
† **Two-stage and non end-to-end approach.**

Table D3: EMDM v.s. DDGAN on HumanML3D.

Methods	R Precision ↑			FID↓	MM Dist↓	Diversity→	MModality↑
	Top 1	Top 2	Top 3				
Real	$0.511^{\pm.003}$	$0.703^{\pm.003}$	$0.797^{\pm.002}$	$0.002^{\pm.000}$	$2.974^{\pm.008}$	$9.503^{\pm.065}$	-
Naive DDGAN	$0.072^{\pm.003}$	$0.140^{\pm.004}$	$0.207^{\pm.006}$	$31.085^{\pm.256}$	$7.389^{\pm.034}$	$5.060^{\pm.059}$	**$3.155^{\pm.092}$**
EMDM (Ours)	**$0.498^{\pm.007}$**	**$0.684^{\pm.006}$**	**$0.786^{\pm.006}$**	**$0.112^{\pm.019}$**	**$3.110^{\pm.027}$**	**$9.551^{\pm.078}$**	$1.641^{\pm.078}$

Blue indicates the best result.

D.2 Comparisons with DDGAN [10]

In addition to the DDIM approach, the recent work DDGAN [10] proposes another implementation of a few-step sampling for efficient image generation. Next, we compare EMDM with a baseline model that directly combines DDGAN [10] and a representative motion diffusion model MDM [9]. Specifically, the baseline approach trains without a condition passed to the discriminator with the weights of geometric loss set to be 0 ($R = 0$). The experiment is conducted using HumanML3D datasets for the text-to-motion task. As shown in Tab. D3, Naive DDGAN produces poor performance in terms of generated motion quality, which is because motion generation typically requires more specific constraints for each frame of the movement.

D.3 Ablation Study on Conditioning with Geometric Loss.

As shown in Tab. D4, without providing conditions to the discriminator, the performance in motion quality is slightly worse. This proves the necessity of providing text/action conditions to the discriminator, which is different from naive DDGAN [10].

D.4 Physical Plausibility.

As discussed in the limitation section, kinematics-based motion generation methods currently focus more on motion semantics and typically suffer from physical implausibility. We report penetration and skate metrics following [11] in Tab. D5,

Table D4: Influence of condition on discriminator. Both models are trained to the same number of epochs.

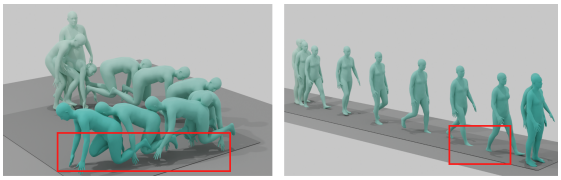
Diffusion Steps	R Precision ↑			FID↓	MM Dist↓	Diversity→	MModality↑
	Top 1	Top 2	Top 3				
Real	$0.424^{\pm.005}$	$0.649^{\pm.006}$	$0.779^{\pm.006}$	$0.031^{\pm.004}$	$2.788^{\pm.012}$	$11.08^{\pm.097}$	-
Without	$0.467^{\pm.006}$	$0.666^{\pm.006}$	$0.771^{\pm.006}$	$0.510^{\pm.037}$	$3.209^{\pm.021}$	$10.01^{\pm.072}$	$2.221^{\pm.021}$
With (Ours)	$0.476^{\pm.005}$	$0.674^{\pm.004}$	$0.779^{\pm.004}$	$0.506^{\pm.031}$	$3.187^{\pm.017}$	$10.03^{\pm.075}$	$2.235^{\pm.039}$

Table D5: Comparison on Physical Plausibility.

Method	Penetration	Skate
EMDM	0.094	1.083
MDM	0.064	0.878
T2M-GPT	0.356	2.618

evaluated on 200 motions, where our motion quality is better than T2M-GPT and comparable with MDM. We agree that injecting physics information can be a promising future direction.

E Limitations and Future Works



"a person got down and is crawling across the floor." (a)

"a person walks forward and stops." (b)

Fig. E5: Motion artifacts: (a) floating and (b) ground penetration in the generated human motion.

While EMDM demonstrates promising performance in efficient human motion generation, it lacks physical considerations, leading to issues such as floating and ground penetration; See Fig. E5. Integrating physics-based characters shows potential for future improvements. Additionally, although EMDM currently primarily accepts textual inputs, it has the potential to incorporate visual inputs or music sources for online motion synthesis, offering exciting research directions.

References

1. Chen, X., Jiang, B., Liu, W., Huang, Z., Fu, B., Chen, T., Yu, G.: Executing your commands via motion diffusion in latent space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18000–18010 (2023) 2, 5
2. Guo, C., Zou, S., Zuo, X., Wang, S., Ji, W., Li, X., Cheng, L.: Generating diverse and natural 3d human motions from text. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5152–5161 (June 2022) 2
3. Guo, C., Zuo, X., Wang, S., Zou, S., Sun, Q., Deng, A., Gong, M., Cheng, L.: Action2motion: Conditioned generation of 3d human motions. In: Proceedings of the 28th ACM International Conference on Multimedia. pp. 2021–2029 (2020) 2
4. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 5
5. Mescheder, L., Geiger, A., Nowozin, S.: Which training methods for gans do actually converge? In: International conference on machine learning. pp. 3481–3490. PMLR (2018) 5
6. Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A.A.A., Tzionas, D., Black, M.J.: Expressive body capture: 3d hands, face, and body from a single image. In: Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2019) 2
7. Petrovich, M., Black, M.J., Varol, G.: Action-conditioned 3D human motion synthesis with transformer VAE. In: International Conference on Computer Vision (ICCV) (2021) 2
8. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning. pp. 8748–8763. PMLR (2021) 3
9. Tevet, G., Raab, S., Gordon, B., Shafir, Y., Bermano, A.H., Cohen-Or, D.: Human motion diffusion model. arXiv preprint arXiv:2209.14916 (2022) 2, 5, 6
10. Xiao, Z., Kreis, K., Vahdat, A.: Tackling the generative learning trilemma with denoising diffusion gans. arXiv preprint arXiv:2112.07804 (2021) 4, 5, 6
11. Yuan, Y., Song, J., Iqbal, U., Vahdat, A., Kautz, J.: Physdiff: Physics-guided human motion diffusion model. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16010–16021 (2023) 6
12. Zhang, M., Cai, Z., Pan, L., Hong, F., Guo, X., Yang, L., Liu, Z.: Motiondiffuse: Text-driven human motion generation with diffusion model. arXiv preprint arXiv:2208.15001 (2022) 5