Real-time Controllable Motion Generation via Latent Consistency Model

Wenxun Dai, Ling-Hao Chen, Yufei Huo, Jingbo Wang, Jinpeng Liu, Bo Dai, Yansong Tang

Abstract—Existing methods for spatial-temporal control in text-conditioned motion generation suffer from significant runtime inefficiency. To address this issue, we first propose the motion latent consistency model (**MotionLCM**) for motion generation, building on the motion latent diffusion model. By adopting one-step (or few-step) inference, we further improve the runtime efficiency of the motion latent diffusion model for motion generation. To ensure effective controllability, we incorporate a motion ControlNet within the latent space of MotionLCM and enable explicit control signals (*i.e.*, initial motions) in the vanilla motion space to further provide supervision for the training process. However, the suboptimal latent space caused by the single-latent-token compression leads to the generation lacking expressive motion details. To tackle this limitation, we design a latent adapter to directly control the VAE compression rate, thereby providing a more compact latent space for high-performance multi-latent-token consistency distillation (**MotionLCM-M**). Besides, to support more general joint-based control, we propose consistency latent tuning, which leverages the gradients of error from the motion space to iteratively refine the learnable latent noise, enabling MotionLCM-M to effectively handle sparse control signals while preserving the naturalness of the generated motions. We also show our method can be extended to the real-time music-to-dance task by jointly modeling the motion dynamics of the upper and lower body. Experimental results demonstrate the remarkable generation and controlling capabilities of our method while maintaining real-time runtime efficiency. Our codes are available at https://github.com/Dai-Wenxun/MotionLCM.

Index Terms—Text-to-Motion, Real-time Control, Consistency Model, Music-to-Dance.

1 Introduction

Text-to-motion generation (T2M) has attracted increasing attention [2], [5], [9], [13], [14] due to its important roles in many applications [15], [16]. Previous attempts mainly focus on GANs [13], [17], VAEs [2], [18]–[20] and diffusion models [4]–[6], [21]–[24] via pairwise text-motion data [1], [25]–[31] and achieve impressive generation results. Existing approaches [4]–[6] mainly take diffusion models [32]–[35] as a base generative model, owing to their powerful ability to model motion distribution. However, these diffusion fashions inevitably require considerable sampling steps for motion synthesis during inference, even with some sampling acceleration methods [36]. Specifically, MDM [5] and MLD [6] require ∼24s and ∼0.2s to generate a high-quality motion sequence. Such low efficiency blocks the applications of generating high-quality motions in various real-time scenarios.

In addition to the language description itself serving as a coarse control signal, another line of research focuses on controlling the motion generation with spatial-temporal constraints [23], [37], [38]. Although these attempts enjoy impressive controlling ability in the T2M task, there still exists a significant gap towards real-time applications. For example, OmniControl [38] exhibits a relatively long inference time, ∼81s per sequence. Therefore, trading-off between generation quality and efficiency is a challenging problem. As a result, in this paper, we target the real-time controllable motion generation research problem.

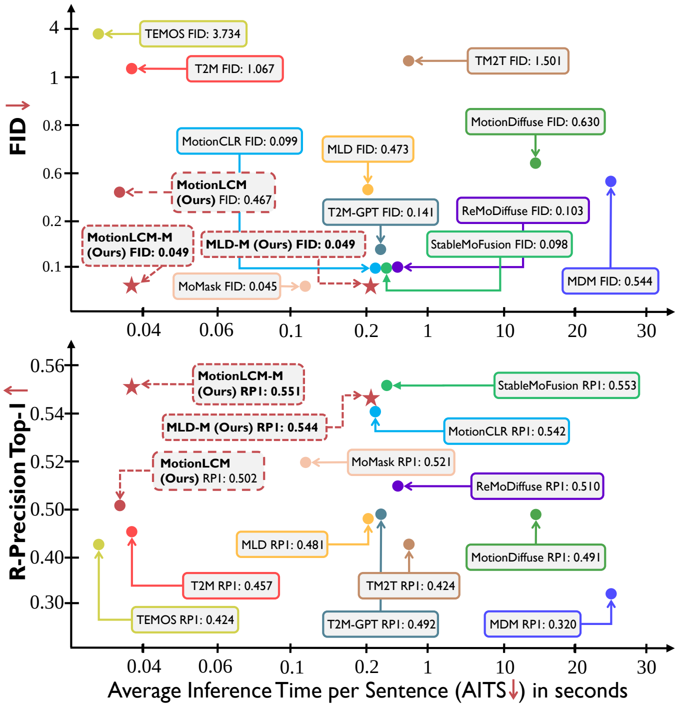
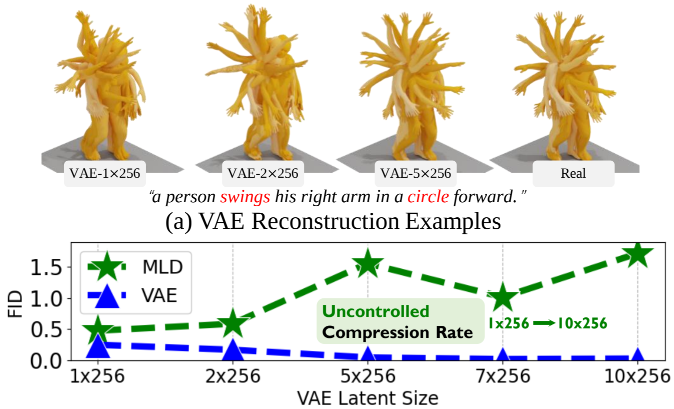


Fig. 1: Comparison of inference time costs on HumanML3D [1]. We evaluate the performance of our methods using AITS, FID, and R-Precision Top-1 metrics, benchmarking them against the state-of-the-arts [1]–[11]. The statistics are sourced from previous works [6], [9]–[12]. Our **MotionLCM** achieves real-time inference speed while ensuring high-quality motion generation. **MLD-M** significantly improves generation performance over its predecessor, MLD. **MotionLCM-M** further advances the state of text-to-motion generation by excelling in inference speed, motion generation quality, and motion-text alignment capability.

- *W. Dai, L.H. Chen, Y. Huo, J. Liu, Y. Tang are with the Tsinghua Shenzhen International Graduate School, Tsinghua University, China 518071. J. Wang is with the Shanghai AI Laboratory. B. Dai is with the University of Hong Kong. E-mail: ({daiwx23, clh22, huoyf24, liujp22}@mails.tsinghua.edu.cn; wangjingbo@pjlab.org.cn; bdai@hku.hk; tang.yansong@sz.tsinghua.edu.cn).*



(a) VAE Reconstruction Examples

(b) VAE and MLD Performance Curve

Fig. 2: (a) The reconstruction visualizations of VAEs with different numbers of latent tokens. A closer match to the real reference indicates better reconstruction precision. (b) Comparison of FID scores across different VAE latent sizes (*i.e.*, compression rates). An uncontrolled compression rate restricts MLD to low-quality single-latent-token diffusion (*i.e.*, 1x256), hindering the potential to achieve a higher upper bound (*e.g.*, 2x256, etc) for motion generation quality. The statistics are sourced from MLD [6].

Recently, the concept of consistency models [39], [40] has been introduced in image generation, resulting in significant progress by enabling efficient and high-fidelity image synthesis with a minimal number of sampling steps (*e.g.*, 4 steps *vs.* 50 steps). These properties perfectly align with our goal of accelerating motion generation without compromising generation quality. Therefore, we propose **MotionLCM** (Motion Latent Consistency Model) distilled from the motion latent diffusion model, MLD [6], to tackle the low-efficiency problem in diffusion sampling. To the best of our knowledge, we introduce consistency distillation into the motion generation area *for the first time* and accelerate motion generation to a real-time level (∼30ms per motion sequence) via latent consistency distillation [40].

As the foundation for latent consistency distillation, the VAE leverages the learnable Gaussian distribution parameters to conduct motion compression and then samples the latent tokens for the next stage of latent diffusion. This compression process simplifies the complex motion distribution into a target distribution that is easier for diffusion learning. However, as shown in Figure 2a, decreasing the number of latent tokens (*i.e.*, the dimension of the latent space) reduces the capture of expressive motion details, negatively impacting both text matching and motion quality. However, Figure 2b shows that simply increasing the number of latent tokens results in decline in motion generation performance as indicated by the green dashed curve. We attribute this to the increased complexity of the target distribution caused by a higher number of latent tokens, which makes diffusion training more difficult. These issues limit the original MLD and MotionLCM to *single-latent-token* learning (*i.e.*, 1x256), leading to a lower upper bound on motion generation quality compared to using VAEs with *multi-latent-token* compression (*e.g.*, 2x256, etc.). Therefore, we further propose **MotionLCM-M** based on **MLD-M**, which employs a latent adapter to directly control the size of the latent space (*i.e.*, compression rate). This elegant design (Section 3.2) enables us to leverage the strong compression capability of multi-latent tokens, providing a more compact latent space for subsequent diffusion and consistency distillation. As shown in Figure 1, MLD-M significantly outperforms the vanilla MLD in both motion-text alignment capability and motion generation quality. MotionLCM-M achieves an approximately **10**× improvement in FID (0.049), with an R-Precision Top-1 score of 0.551, while maintaining real-time inference speed (∼38ms per motion sequence).

Here, in MotionLCM, we are facing another challenge on how to control motions with spatial-temporal signals (*i.e.*, initial motions) in the latent space. Previous methods [23], [38] model human motions in the vanilla motion space and can manipulate the motion directly in the denoising process. However, for our latent-diffusion-based MotionLCM, it is non-trivial to feed the control signals into the latent space. This is because the latent has no explicit motion semantics, which cannot be manipulated directly by the control signals. Inspired by the notable success of [41] in controllable image generation [35], we introduce a motion ControlNet to control motion generation in the latent space. However, the naïve motion ControlNet is not sufficient to provide supervision for the control signals. The main reason is the lack of explicit supervision in the motion space. Therefore, during the training phase, we decode the predicted latent through the frozen VAE [42] decoder into the vanilla motion space to provide explicit control supervision on the generated motion. Thanks to the powerful one-step inference capability of MotionLCM, the latent generated by MotionLCM can significantly facilitate control supervision both in the latent space and motion space for training the motion ControlNet compared to MLD [6].

In contrast to initial motions as control signals, which offer dense control across both temporal and spatial dimensions, controlling motion using sparse signals is more challenging, *e.g.*, controlling a hand joint at a specific keyframe. Therefore, to support more general joint-based control, we propose a consistency latent tuning (Section 3.5) method that leverages the gradients of error from the motion space to iteratively refine the learnable latent noise, ensuring alignment with the imposed sparse spatial constraints. Specifically, we first sample learnable latent noise from standard Gaussian distribution and then use the frozen MotionLCM and VAE decoder to perform one-step inference and generate the full motion sequence. This motion sequence is used to align with the control signals to directly fine-tune the learnable latent noise. However, relying solely on the control loss from the vanilla motion space can cause the generated motion to seek to strictly match the control signals, resulting in unrealistic motion. Inspired by the previous work [43], we introduce a latent decorrelation loss across the latent tokens, which regularizes the latent noise by decorrelating each latent dimension and significantly reduces the issue of unnatural motion. The consistency latent tuning method enables our MotionLCM-M to effectively handle sparse control signals while preserving the naturalness of the generated motion sequences.

Besides, we explore the potential application of motion latent consistency distillation in the music-to-dance task (Section 3.6). Compared to traditional text-to-motion generation, dance motion requires more complex and expressive movements from different body parts. Therefore, we employ two independent VQ-VAEs [44] to separately encode the upper and lower body, achieving part-based decoupling and enhancing expressiveness. Additionally, we propose a VQ-based motion latent diffusion model (VQ-MLD) that jointly denoises the concatenated upper and lower body features conditioned on music input. These denoised dual-part features are then fed into their respective VQ-VAE decoders to generate a coherent and natural dance motion sequence. By performing latent consistency distillation on the pre-trained VQ-MLD, we achieve real-time music-to-dance generation.

This paper is an extended version of our ECCV'24 conference paper [12] with the following new contributions:

1) We further propose MotionLCM-M, which incorporates a latent adapter to directly control the VAE compression rate, which effectively addresses the lack of expressive motion details caused by the sub-optimal latent space and thereby enabling a more compact latent space for high-performance multi-latent-token consistency distillation.
2) We introduce consistency latent tuning, which leverages the gradients of error derived from the motion space to iteratively refine the learnable latent noise. This tuning paradigm enables MotionLCM-M to effectively handle sparse control signals while preserving the naturalness of the generated motions.
3) We extend our method to the music-to-dance task by distilling the newly designed VQ-based motion latent diffusion model (VQ-MLD), which jointly models the motion dynamics of the upper and lower body, achieving a new state-of-the-art performance while maintaining real-time inference speed.

2 Related Work

Generating human motions can be divided into three main fashions according to inputs: motion synthesis (1) without any condition [5], [45]–[47], (2) with some given multi-modal conditions, such as action labels [18], [19], [48]–[51], textual description [1]–[7], [9]–[11], [13]–[17], [20]–[24], [37], [52]–[69], audio or music [70]–[78], (3) with user-defined trajectories [23], [37], [38], [43], [59], [79]–[84]. In this section, we present an overview of the related works from the following three aspects: (1) text-to-motion, (2) controllable motion generation, and (3) music-to-dance.

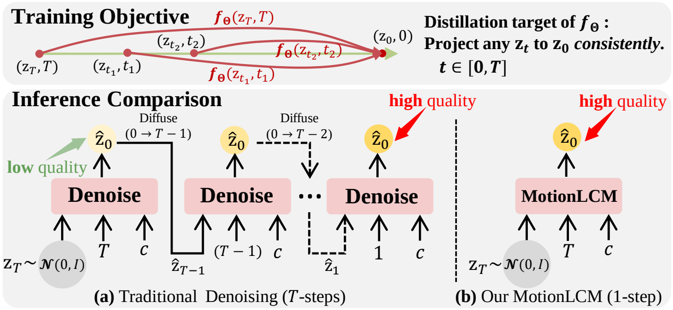


Fig. 3: The training objective of consistency distillation is to learn a consistency function f_Θ, initialized with the parameters of a pre-trained diffusion model (*e.g.*, MLD [6]). This function f_Θ should projects any points (*i.e.*, $\mathbf{z}_t$) on the ODE trajectory to its solution (*i.e.*, $\mathbf{z}_0$). Once the pre-trained model [6] is distilled, unlike the traditional denoising model [4], [5] that requires considerable sampling steps, our MotionLCM can generate high-quality motion sequences with one-step sampling and further improve the generation quality through multi-step inference.

Text-to-Motion. JL2P [53] jointly embeds text and motion using curriculum learning to prioritize shorter sequences before longer, more complex ones. MotionCLIP [56] leverages the knowledge embedded in CLIP for text-conditioned motion generation. TEMOS [2] designs a variational autoencoder (VAE) to model motion sequences, generating diverse motions. MDM [5] presents a motion diffusion model that operates on raw motion data, offering both high-quality generation and versatile conditioning, establishing a strong baseline for new motion generation tasks. MLD [6] introduces a latent diffusion model that enhances generation quality while reducing computational resource demands. ReMoDiffuse [8] integrates a retrieval mechanism to refine the motion denoising process, enhancing the generalizability and diversity of text-driven motion generation. PhysDiff [22] proposes a physics-guided motion diffusion model that generates physically plausible motions. TMR [57] incorporates a contrastive framework for retrieval-based motion synthesis under text conditions. T2M-GPT [7] utilizes a GPT-like model to generate high-quality human motions from textual descriptions with discrete representations. MotionGPT [58] employs motion-language pre-training and prompt learning to build a unified and versatile model for diverse motion-related tasks, such as text-driven motion generation. HumanTOMATO [14] presents a text-driven framework for whole-body motion generation, applicable to holistic motion synthesis. Momask [9] employs a hierarchical quantization scheme for motion tokenization and designs a masked transformer and residual transformer to predict motion tokens. StableMoFusion [10] thoroughly investigates network architectures, training strategies, and inference processes in motion diffusion models. ScaMo [69] introduces an auto-regressive motion generation framework to explore scaling laws in the text-to-motion task.

Controllable Motion Generation. Shafir *et al.* [23] propose PriorMDM, which achieves accurate control by blending models with different control signals. GMD [37] introduces a guided motion diffusion model that incorporates spatial constraints into the motion generation process. OmniControl [38] integrates flexible spatial-temporal control signals across different joints by combining analytic spatial guidance with realism guidance. InterControl [79] presents a novel controllable motion generation method designed to maintain the desired distance between joint pairs in the synthesized motions. TLControl [59] incorporates both trajectory control and language semantics control through the integration of neural-based and optimization-based techniques. A-MDM [83] introduces an auto-regressive motion diffusion model and incorporates interactive controls, enabling efficient adaptation to a variety of new downstream tasks. The AAMDM [84] framework enhances interactive control in motion synthesis by integrating denoising diffusion GANs with auto-regressive diffusion models in a lower-dimensional embedded space. ControlMM [80] proposes masked consistency modeling for high-fidelity motion generation and uses inference-time logit editing to adjust the predicted motion distribution, ensuring the generated motion adheres to the input control signals.

Music-to-Dance. Li *et al.* [70] present a novel two-stream motion transformer generative model that synthesizes dance motion sequences with high flexibility. DanceNet [71] introduces an autoregressive generative model that uses the style, rhythm, and melody of music as control signals to generate 3D dance motions characterized by both realism and diversity. DanceRevolution [72] proposes a novel Seq2Seq architecture for the music-conditioned dance generation task. FACT [73] incorporates a deep cross-modal transformer block with full attention, trained to predict future motion sequences. DanceFormer [74] reformulates the music-to-dance task as a two-stage framework, involving key pose generation followed by the prediction of parametric motion curves for intermediate frames. Bailando [75] employs an actor-critic GPT model to compose dance units into a coherent and fluent dance synchronized with the music. EDGE [76] introduces an editable dance generation framework capable of producing realistic, physically plausible dances that remain faithful to the input music. FineDance [77] addresses the issue of monotonous and unnatural hand movements in prior methods by proposing a full-body dance generation network. Lodge [78] presents a two-stage coarse-to-fine diffusion architecture and introduces expressive dance primitives as intermediate representations between the two diffusion models.

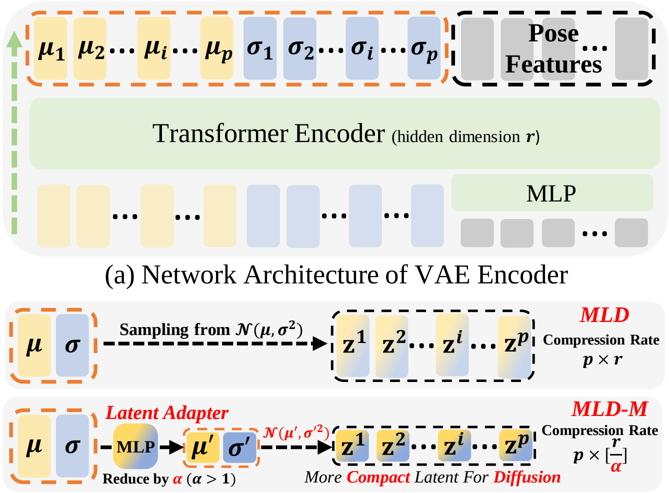
3 Method

In this section, we first briefly introduce preliminaries about latent consistency models in Section 3.1. Next, we illustrate the technical design details of multi-latent-token diffusion models in Section 3.2. Then, we describe how to conduct latent consistency distillation for motion generation in Section 3.3, followed by our implementation of motion control in latent space in Section 3.4. Next, we present the consistency latent tuning method in Section 3.5. Finally, we introduce our VQ-based motion latent diffusion framework (VQ-MLD) for the music-to-dance task in Section 3.6. The overall pipeline is illustrated in Figure 5.

3.1 Preliminaries

The Consistency Model (CM) [39] introduces a kind of efficient generative model designed for efficient one-step or few-step generation. Given a Probability Flow ODE (*a.k.a.* PF-ODE) that smoothly converts data to noise, the CM is to learn the function $\boldsymbol{f}(\cdot,\cdot)$ that maps any points on the ODE trajectory to its origin distribution (*i.e.*, the solution of the PF-ODE). The consistency function is formally defined as $\boldsymbol{f} : (\mathbf{x}_t, t) \longmapsto \mathbf{x}_\epsilon$, where $t \in [\epsilon, T]$, $T > 0$ is a fixed constant and ϵ is a small positive number to avoid numerical instability. According to [39], the consistency function should satisfy the *self-consistency property*:

$$\boldsymbol{f}(\mathbf{x}_t, t) = \boldsymbol{f}(\mathbf{x}_{t'}, t'), \forall t, t' \in [\epsilon, T]. \tag{1}$$


μ_1 μ_2 … μ_i … μ_p σ_1 σ_2 … σ_i … σ_p
Pose Features …
Transformer Encoder (hidden dimension r)
MLP

(a) Network Architecture of VAE Encoder

μ σ
Sampling from $\mathcal{N}(\mu, \sigma^2)$
z^1 z^2 … z^i … z^p
MLD
Compression Rate $p \times r$
Latent Adapter
MLP
μ' σ'
Reduce by α ($\alpha > 1$)
$\mathcal{N}(\mu', \sigma'^2)$
MLD-M
Compression Rate $p \times [\frac{r}{\alpha}]$
More Compact Latent For Diffusion

(b) Diffusion Latent Sampling in MLD and MLD-M

Fig. 4: Comparison of latent space construction between MLD and MLD-M. (a) The VAE encoder compresses the motion sequence using the learnable Gaussian distribution parameters (*i.e.*, μ_i and σ_i) to fuse the projected pose features. (b) In the original MLD, the diffusion latent tokens $\mathbf{z}^i$ are directly sampled from the Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$, resulting in the final compression rate of $p \times r$ (*i.e.*, the size of the latent space). In MLD-M, a latent adapter is adopted to control the compression rate (*i.e.*, $p \times \frac{r}{\alpha}$), thereby avoiding a continuous decrease in the compression rate as the number of tokens p increases and providing a more compact latent space for the next-stage latent diffusion.

As shown in Equation (1), the self-consistency property indicates that for arbitrary pairs of $(\mathbf{x}_t, t)$ on the same PF-ODE trajectory, the outputs of the model should be consistent. The goal of a parameterized consistency model $\boldsymbol{f}_{\boldsymbol{\Theta}}$ is to learn a consistency function from data by enforcing the self-consistency property in Equation (1). To ensure that $\boldsymbol{f}_{\boldsymbol{\Theta}}(\mathbf{x}, \epsilon) = \mathbf{x}$, the consistency model $\boldsymbol{f}_{\boldsymbol{\Theta}}$ is parameterized as,

$$\boldsymbol{f}_{\boldsymbol{\Theta}}(\mathbf{x}, t) = c_{\text{skip}}(t)\mathbf{x} + c_{\text{out}}(t)\boldsymbol{F}_{\boldsymbol{\Theta}}(\mathbf{x}, t), \tag{2}$$

where $c_{\text{skip}}(t)$ and $c_{\text{out}}(t)$ are differentiable functions with $c_{\text{skip}}(\epsilon) = 1$ and $c_{\text{out}}(\epsilon) = 0$, and $\boldsymbol{F}_{\boldsymbol{\Theta}}(\cdot,\cdot)$ is a deep neural network to learn the self-consistency. The CM trained from distilling the knowledge of pre-trained diffusion models is called *Consistency Distillation*. The consistency loss is defined as follows,

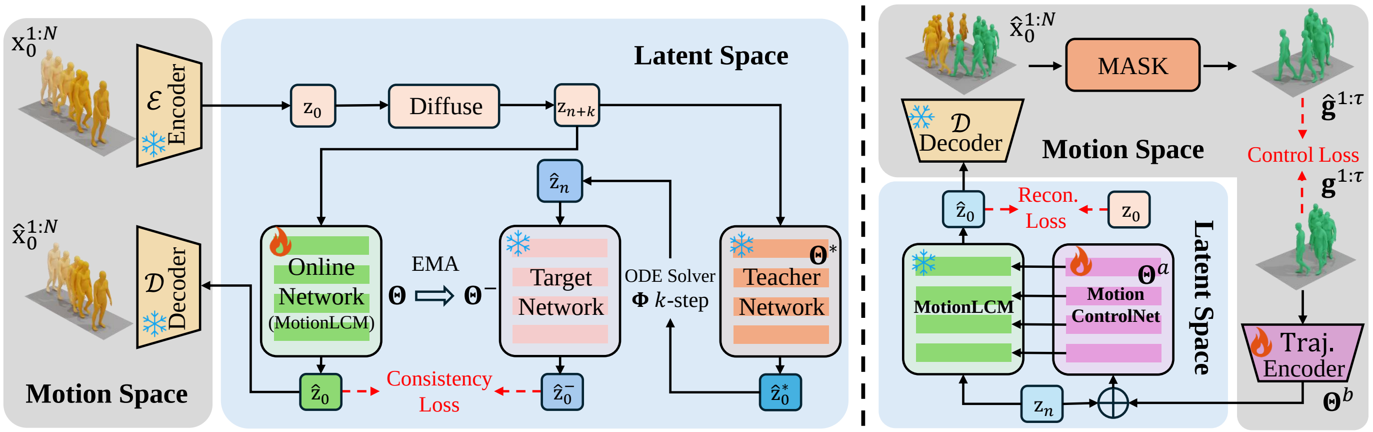
$$\mathcal{L}(\boldsymbol{\Theta}, \boldsymbol{\Theta}^-; \boldsymbol{\Phi}) = \mathbb{E}\left[d\left(\boldsymbol{f}_{\boldsymbol{\Theta}}(\mathbf{x}_{t_{n+1}}, t_{n+1}), \boldsymbol{f}_{\boldsymbol{\Theta}^-}(\hat{\mathbf{x}}^{\boldsymbol{\Phi}}_{t_n}, t_n)\right)\right], \tag{3}$$

where $d(\cdot,\cdot)$ is a chosen metric function for measuring the distance between two samples. $\boldsymbol{f}_{\boldsymbol{\Theta}}(\cdot,\cdot)$ and $\boldsymbol{f}_{\boldsymbol{\Theta}^-}(\cdot,\cdot)$ are referred to as "online network" and "target network" according to [39]. Besides, $\boldsymbol{\Theta}^-$ is updated with the exponential moving average (EMA) of the parameters of $\boldsymbol{\Theta}$ [1]. In Equation (3), $\hat{\mathbf{x}}^{\boldsymbol{\Phi}}_{t_n}$ is a one-step estimation of $\mathbf{x}_{t_n}$ from $\mathbf{x}_{t_{n+1}}$, which is formulated as,

$$\hat{\mathbf{x}}^{\boldsymbol{\Phi}}_{t_n} \leftarrow \mathbf{x}_{t_{n+1}} + (t_n - t_{n+1})\boldsymbol{\Phi}(\mathbf{x}_{t_{n+1}}, t_{n+1}, \emptyset), \tag{4}$$

where $\boldsymbol{\Phi}$ is a one-step ODE solver applied to PF-ODE.

[1] EMA operation: $\boldsymbol{\Theta}^- \leftarrow \mathtt{sg}(\mu\boldsymbol{\Theta}^- + (1-\mu)\boldsymbol{\Theta})$, where $\mathtt{sg}(\cdot)$ denotes the stopgrad operation and μ satisfies $0 \leq \mu < 1$.



(a) Motion Latent Consistency Distillation

(b) Motion Control in Latent Space

Fig. 5: **The overview of MotionLCM.** (a) *Motion Latent Consistency Distillation (Section 3.3).* Given a raw motion sequence $\mathbf{x}_0^{1:N}$, a pre-trained VAE [42] encoder first compresses it into the latent space, then a forward diffusion operation is performed to add $n+k$ steps of noise. Then, the noisy $\mathbf{z}_{n+k}$ is fed into the online network and teacher network to predict the clean latent. The target network takes the k-step estimation results of the teacher output to predict the clean latent. To learn self-consistency, a loss is applied to enforce the output of the online network and target network to be consistent. (b) *Motion Control in Latent Space (Section 3.4).* With the powerful MotionLCM trained in the first stage, we incorporate a motion ControlNet into the MotionLCM to achieve controllable motion generation. Furthermore, we leverage the decoded motion to explicitly supervise the spatial-temporal control signals (*i.e.*, initial poses $\mathbf{g}^{1:\tau}$).

The Latent Consistency Model (LCM) [40] learns the self-consistency property in the latent space $D_{\mathbf{z}} = \{(\mathbf{z}, \mathbf{c}) | \mathbf{z} = \mathcal{E}(\mathbf{x}), (\mathbf{x}, \mathbf{c}) \in D\}$, where D denotes the dataset, $\mathbf{c}$ is the given condition, and $\mathcal{E}$ is the pre-trained encoder. Compared to CMs [39] using the numerical continuous PF-ODE solver [85], LCMs [40] adopt the discrete-time schedule [36], [86], [87] to adapt to Stable Diffusion [35]. Instead of ensuring consistency between adjacent time steps $t_{n+1} \to t_n$, LCMs [40] are designed to ensure consistency between the current time step and k-step away, *i.e.*, $t_{n+k} \to t_n$, thereby significantly reducing convergence time costs. As classifier-free guidance (CFG) [88] plays a crucial role in synthesizing high-quality text-aligned images, LCMs integrate CFG into the distillation as follows,

$$\hat{\mathbf{z}}_{t_n}^{\mathbf{\Phi},w} \leftarrow \mathbf{z}_{t_{n+k}} + (1+w)\mathbf{\Phi}(\mathbf{z}_{t_{n+k}}, t_{n+k}, t_n, \mathbf{c}) \\ - w\mathbf{\Phi}(\mathbf{z}_{t_{n+k}}, t_{n+k}, t_n, \emptyset), \tag{5}$$

where w denotes the CFG scale which is uniformly sampled from $[w_{\text{min}}, w_{\text{max}}]$ and k is the skipping interval. To efficiently perform the above k-step guided distillation, LCMs augment the consistency function to $\boldsymbol{f} : (\mathbf{z}_t, t, w, \mathbf{c}) \longmapsto \mathbf{z}_0$, which is also the form adopted by our MotionLCM.

3.2 Multi-Latent-Token Diffusion Models

The success of the motion latent diffusion paradigm [6] fundamentally relies on the achieved **perceptual compression** of the first-stage VAE, *i.e.*, removing high-frequency motion details while preserving essential semantic information. This enables the second-stage MLD to focus on learning the semantic and conceptual composition of the motion data, *i.e.*, **semantic compression**. This principle has also been validated in Stable Diffusion [35]. Therefore, the key to improving the motion generation performance of MLD lies in obtaining an optimal latent space. Additionally, the latent space must carefully balance a suitable compression rate (*i.e.*, the size of the latent space) with the preservation of critical semantic information, allowing the MLD to utilize semantically rich latent representations for high-quality motion generation. As shown in Figure 4, in the vanilla MLD [6], the VAE encoder performs motion compression using the learnable Gaussian distribution parameters (*i.e.*, μ_i and σ_i) to fuse the projected pose features. It then samples latent tokens $\mathbf{z}^i$ from the Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ for the next stage of latent diffusion. The hidden dimension of the VAE encoder is denoted as r and the number of latent tokens is p. This leads to the final compression rate of $p \times r$. However, we observe that increasing the number of latent tokens p improves motion reconstruction precision, but it leads to an unstable decline in motion generation capability as shown in Figure 2b. We attribute this to the uncontrolled compression rate, where increasing the number of latent tokens directly leads to a continuous decline in the compression rate (*i.e.*, 1x256→10x256). This is because the original MLD samples latent tokens directly from the Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ encoded by the VAE encoder. The uncontrolled compression rate results in leaving most of the perceptual compression to the diffusion model, thus hindering the ability to generate high-quality motions. These issues result in the original MLD being limited to *single-latent-token* learning (*i.e.*, 1x256) for diffusion training, leading to a lower upper bound on motion generation quality compared to using VAEs with multi-latent tokens (*e.g.*, 2x256, etc). Therefore, to enable *multi-latent-token* learning for high-performance diffusion, as shown in Figure 4b, in MLD-M, we add a linear layer as the latent adapter to adapt the dimension of the embedded distribution parameters to directly control the size of the latent space $\mathcal{N}(\mu^{'}, \sigma^{'2})$. This elegant design enables us to harness the strong compression capability of multi-latent tokens while maintaining control over the compression rate,

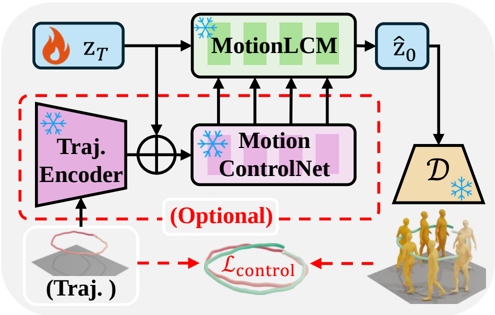


Fig. 6: **Consistency Latent Tuning**. We first sample the latent noise $\mathbf{z}_T$ from $\mathcal{N}(0, I)$ and predict the motion sequence through one-step inference with the frozen MotionLCM and VAE decoder $\mathcal{D}$. Next, we extract the global absolute locations (in green) of the control joint (*i.e.*, root) to align them with the predefined trajectory (in red) using the control loss $\mathcal{L}_{\text{control}}$. The introduction of the motion ControlNet is optional as the MotionLCM itself can serve as a powerful motion prior to our consistency latent tuning.

thereby providing a more compact latent space for the subsequent diffusion stage. Extensive experiments show that by controlling the compression rate using the latent adapter, we enable high-performance multi-latent-token diffusion.

3.3 MotionLCM: Motion Latent Consistency Model

Motion compression into the latent space. Motivated by [39], [40], we propose MotionLCM (Motion Latent Consistency Model) to tackle the low-efficiency problem in motion diffusion models [4], [5], unleashing the potential of LCM in the motion generation task. Similar to MLD [6], our MotionLCM adopts a consistency model in the motion latent space. We choose MLD [6] as the underlying diffusion model to distill from. We aim to achieve few-step (2∼4) and even one-step inference without compromising motion quality. In MLD, an autoencoder ($\mathcal{E}$, $\mathcal{D}$) is first trained to compress a high dimensional motion into low dimensional latent vectors $\mathbf{z} = \mathcal{E}(\mathbf{x})$, which are then decoded to reconstruct the motion as $\hat{\mathbf{x}} = \mathcal{D}(\mathbf{z})$. Training diffusion models in the motion latent space greatly reduces the computational resources compared to the vanilla diffusion models trained on raw motion sequences (*i.e.*, motion space) and speeds up the inference process. Thus, we effectively leverage the motion latent space for consistency distillation.

Motion latent consistency distillation. An overview of our motion latent consistency distillation is described in Figure 5a. A raw motion sequence $\mathbf{x}_0^{1:N} = \{\mathbf{x}^i\}_{i=1}^N$ is a sequence of human poses, where N is the number of frames. We follow [1] to use the redundant motion representation for our experiments, which is widely used in previous work [4]–[6]. As shown in Figure 5a, given a raw motion sequence $\mathbf{x}_0^{1:N}$, a pre-trained VAE [42] encoder first compresses it into the latent space, $\mathbf{z}_0 = \mathcal{E}(\mathbf{x}_0)$. Then, a forward diffusion operation with $n + k$ steps is conducted to add noise on $\mathbf{z}_0$, where k is the skipping interval illustrated in Section 3.1. The noisy $\mathbf{z}_{n+k}$ is fed to the frozen teacher network and trainable online network to predict the clean $\hat{\mathbf{z}}_0^*$, and $\hat{\mathbf{z}}_0$. The target network uses the cleaner $\hat{\mathbf{z}}_n$ obtained by a k-step ODE solver $\mathbf{\Phi}$, such as DDIM [36] to predict the $\hat{\mathbf{z}}_0^-$. Since the classifier-free guidance (CFG) [88] is essential for condition alignment in diffusion models [5], [6], [35], we integrate CFG into the distillation,

$$\begin{aligned}\hat{\mathbf{z}}_n \leftarrow \mathbf{z}_{n+k} &+ (1+w)\mathbf{\Phi}(\mathbf{z}_{n+k}, t_{n+k}, t_n, \mathbf{c}) \\ &- w\mathbf{\Phi}(\mathbf{z}_{n+k}, t_{n+k}, t_n, \emptyset),\end{aligned} \tag{6}$$

where $\mathbf{c}$ is the text condition and w denotes the guidance scale. To ensure the self-consistency property defined in Equation (1), the latent consistency distillation loss $\mathcal{L}_{\text{LCD}}$ is designed as follows,

$$\begin{aligned}\mathcal{L}_{\text{LCD}}(\mathbf{\Theta}, \mathbf{\Theta}^-) = \mathbb{E}\big[d\big(&\boldsymbol{f}_{\mathbf{\Theta}}(\mathbf{z}_{n+k}, t_{n+k}, w, \mathbf{c}), \\ &\boldsymbol{f}_{\mathbf{\Theta}^-}(\hat{\mathbf{z}}_n, t_n, w, \mathbf{c})\big)\big],\end{aligned} \tag{7}$$

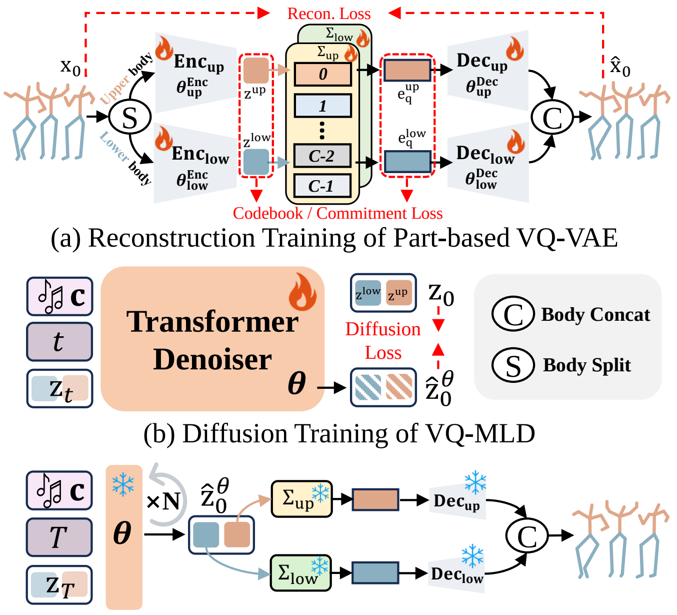
where $d(\cdot, \cdot)$ is a distance measuring function, such as L2 loss or Huber loss [89]. As discussed in Section 3.1, the target network $\mathbf{\Theta}^-$ is updated with the exponential moving average (EMA) of the trainable parameters of the online network $\mathbf{\Theta}$. Here we define the teacher network $\mathbf{\Theta}^*$ as the pre-trained motion latent diffusion model, *i.e.*, MLD [6]. According to [40], the online network and target network are initialized with the parameters of the teacher network. During the inference phase, as shown in Figure 8, our MotionLCM can generate high-quality motions with one-step sampling and achieve the fastest runtime (**∼30ms per motion sequence**) compared to other motion diffusion models [5], [6]. Based on multi-latent-token learning, MotionLCM-M is capable of generating motion with richer details while maintaining real-time runtime efficiency.

3.4 Controllable Motion Generation in Latent Space

After addressing the low-efficiency issue in the motion latent diffusion model [6], we delve into another exploration of real-time motion control. Inspired by the great success of ControlNet [41] in controllable image generation [35], we introduce a motion ControlNet $\mathbf{\Theta}^a$ in the latent space of MotionLCM and initialize the motion ControlNet with a trainable copy of MotionLCM. Specifically, each layer in the motion ControlNet is appended with a zero-initialized linear layer for eliminating random noise in the initial training steps. To achieve an autoregressive motion generation paradigm, we define the motion control task as generating motions given the initial τ poses and textual description. As shown in Figure 5b, the initial τ poses are defined by the trajectories of K control joints, $\mathbf{g}^{1:\tau} = \{\mathbf{g}^i\}_{i=1}^{\tau}$, where $\mathbf{g}^i \in \mathbb{R}^{K \times 3}$ denotes the global absolute locations of each control joint. In our motion control pipeline, we design a Trajectory Encoder $\mathbf{\Theta}^b$ consisting of stacked transformer [90] layers to encode the trajectory signals. We append a global token (*i.e.*, [CLS]) before the start of the trajectory sequence as the output feature of the encoder, which is added to the noisy $\mathbf{z}_n$ and fed into the trainable motion ControlNet $\mathbf{\Theta}^a$. Under the guidance of motion ControlNet, MotionLCM predicts the denoised $\hat{\mathbf{z}}_0$ through the consistency function $\boldsymbol{f}_{\mathbf{\Theta}^s}$, where $\mathbf{\Theta}^s$ is the combination of $\mathbf{\Theta}^a$, $\mathbf{\Theta}^b$ and $\mathbf{\Theta}$. The following reconstruction loss $\mathcal{L}_{\text{recon}}$ optimizes the motion ControlNet $\mathbf{\Theta}^a$ and Trajectory Encoder $\mathbf{\Theta}^b$,

$$\mathcal{L}_{\text{recon}}(\mathbf{\Theta}^a, \mathbf{\Theta}^b) = \mathbb{E}\left[d\left(\boldsymbol{f}_{\mathbf{\Theta}^s}\left(\mathbf{z}_n, t_n, w, \mathbf{c}^*\right), \mathbf{z}_0\right)\right], \tag{8}$$

where $\mathbf{c}^*$ includes the text condition and control guidance from the Trajectory Encoder and the motion ControlNet.


Recon. Loss
x_0
Upper body
Lower body
S
$\mathrm{Enc}_{\mathrm{up}}$
$\theta^{\mathrm{Enc}}_{\mathrm{up}}$
$\mathrm{Enc}_{\mathrm{low}}$
$\theta^{\mathrm{Enc}}_{\mathrm{low}}$
z^{up}
$\mathrm{z}^{\mathrm{low}}$
Σ_{low}
Σ_{up}
0
1
C-2
C-1
$\mathrm{e}_{\mathrm{q}}^{\mathrm{up}}$
$\mathrm{e}_{\mathrm{q}}^{\mathrm{low}}$
$\mathrm{Dec}_{\mathrm{up}}$
$\theta^{\mathrm{Dec}}_{\mathrm{up}}$
$\mathrm{Dec}_{\mathrm{low}}$
$\theta^{\mathrm{Dec}}_{\mathrm{low}}$
C
$\hat{\mathrm{x}}_0$
Codebook / Commitment Loss

(a) Reconstruction Training of Part-based VQ-VAE

c
t
z_t
Transformer Denoiser
θ
$\mathrm{z}^{\mathrm{low}}$ z^{up}
z_0
Diffusion Loss
$\hat{\mathrm{z}}_0^{\theta}$
C Body Concat
S Body Split

(b) Diffusion Training of VQ-MLD

c
T
z_T
θ
×N
$\hat{\mathrm{z}}_0^{\theta}$
Σ_{up}
Σ_{low}
$\mathrm{Dec}_{\mathrm{up}}$
$\mathrm{Dec}_{\mathrm{low}}$
C

(c) Music-to-Dance with VQ-MLD

Fig. 7: (a) We employ two independent VQ-VAEs [44] to separately model the upper and lower body parts, providing more expressive latent representations for the next stage of diffusion training. (b) We design a transformer-based denoiser that jointly denoises the dual-part body features conditioned on music input. (c) During inference, the music acts as the control signal, guiding the denoiser in generating dual-part body features through multiple rounds of denoising. These features are then divided into upper-body and lower-body components, which are subsequently quantized and decoded to produce the final dance sequence.

However, during training, the sole reconstruction supervision in the latent space is insufficient. We argue this is because the controllable motion generation requires more detailed constraints, which cannot be effectively provided solely by the reconstruction loss in the latent space. Unlike previous methods like OmniControl [38], which directly diffuse in the motion space, allowing explicit supervision of control signals, effectively supervising control signals in the latent space is non-trivial. Therefore, we utilize the frozen VAE [42] decoder $\mathcal{D}$ to decode the latent $\hat{\mathbf{z}}_0$ into the motion space, obtaining the predicted motion $\hat{\mathbf{x}}_0$, thereby introducing the control loss $\mathcal{L}_{\text{control}}$ as follows,

$$\mathcal{L}_{\text{control}}(\mathbf{\Theta}^a,\mathbf{\Theta}^b)=\mathbb{E}\left[\frac{\sum_i\sum_j m_{ij}||R(\hat{\mathbf{x}}_0)_{ij}-R(\mathbf{x}_0)_{ij}||_2^2}{\sum_i\sum_j m_{ij}}\right], \tag{9}$$

where $R(\cdot)$ converts the joint local positions to global absolute locations and $m_{ij}\in\{0,1\}$ is the binary joint mask at frame i for the joint j. Then we optimize the motion ControlNet $\mathbf{\Theta}^a$ and Trajectory Encoder $\mathbf{\Theta}^b$ with the overall objective,

$$\mathbf{\Theta}^a,\mathbf{\Theta}^b=\underset{\mathbf{\Theta}^a,\mathbf{\Theta}^b}{\arg\min}(\mathcal{L}_{\text{recon}}+\lambda\mathcal{L}_{\text{control}}), \tag{10}$$

where λ is the weight to balance the two losses. This design enables explicit control signals in the vanilla motion space to further provide supervision for the generation process. Extensive experiments demonstrate that the introduced supervision is beneficial in improving motion control performance, which will be introduced in the following section.

3.5 Consistency Latent Tuning

Compared to the initial poses in Figure 5b as control signals, which provide dense control across both temporal and spatial dimensions, controlling motion using sparse signals is more challenging, such as controlling a hand joint at a specific keyframe. To address this issue, we propose a consistency latent tuning method. As shown in Figure 6, we first sample a leanable latent noise $\mathbf{z}_T$ from the standard Gaussian distribution $\mathcal{N}(0,I)$, and then use the frozen MotionLCM and VAE decoder $\mathcal{D}$ to perform one-step inference to generate the full motion sequence. Taking the root joint control as an example (in red), we extract the global absolute locations of the root joint from the generated motion (in green) and fine-tune the latent noise $\mathbf{z}_T$ using the control loss $\mathcal{L}_{\text{control}}$ defined in Equation (9). Notably, the introduction of the motion ControlNet is optional, which means that MotionLCM itself can serve as a powerful motion prior for our consistency latent tuning. However, relying solely on the control loss $\mathcal{L}_{\text{control}}$ can lead to unrealistic motion, such as foot sliding. This happens because the generated motion seeks to strictly match the control signals. Inspired by the previous work [43], we introduce a latent decorrelation loss across the latent tokens, which regularizes the latent noise by decorrelating each latent dimension and significantly reduces the issue of unnatural motion. The latent decorrelation loss $\mathcal{L}^m_{\text{decorr}}$ is defined as follows,

$$\mathcal{L}^m_{\text{decorr}}=\frac{1}{mr}\sum_{i=1}^{m}\mathbf{z}_T^m(i)^\top\mathbf{z}_T^m(i+1), \tag{11}$$

where r is the dimension size of the latent tokens. We employ this loss at several scales across the number of latent tokens $m\in\{M,M/2,M/4,...,2\}$. Specifically, assuming the initial number of latent tokens is M, we progressively downsample the number of tokens by half using average pooling on two consecutive tokens. Then we fine-tune the latent noise $\mathbf{z}_T$ with the following objective,

$$\mathbf{z}_T=\underset{\mathbf{z}_T}{\arg\min}(\mathcal{L}_{\text{control}}+\lambda_{\text{decorr}}\sum_m\mathcal{L}^m_{\text{decorr}}), \tag{12}$$

where λ_{decorr} is the weight to control the latent decorrelation loss. Based on the powerful MotionLCM motion prior, our consistency latent tuning enables precise control over any joint at any time, while achieving high-quality motion generation and strong motion-text alignment capability.

3.6 MotionLCM for Music-to-Dance

Compared to traditional text-to-motion generation, dance motion requires more complex and expressive movements from different body parts. Therefore, we first adopt two independent VQ-VAEs [44] to separately encode the upper and lower body, achieving part-based decoupling into the discrete latent space. Additionally, we introduce our VQ-based motion latent diffusion model (VQ-MLD) to jointly denoise the concatenated upper and lower body features conditioned on music input. During the inference stage, the

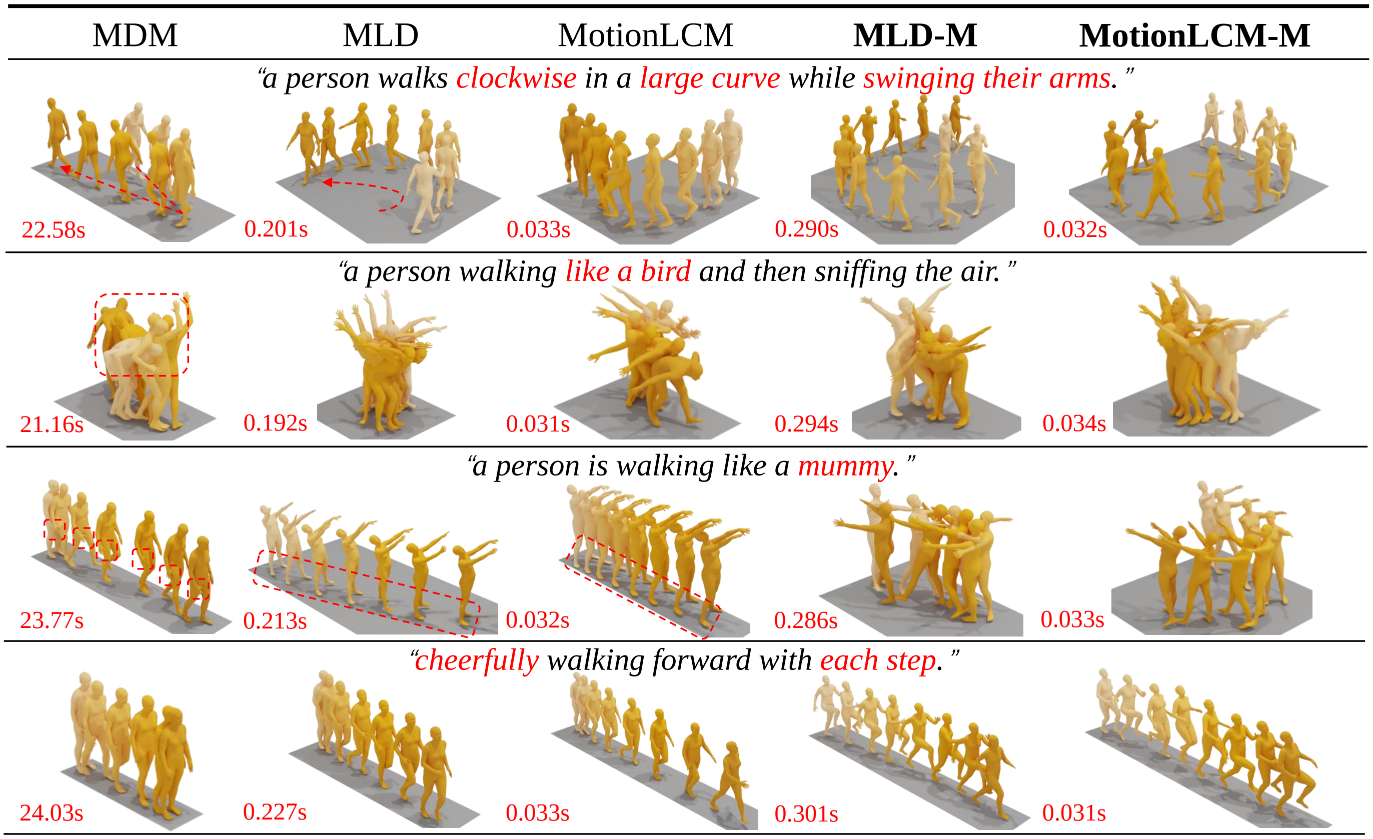

MDM
MLD
MotionLCM
MLD-M
MotionLCM-M
"a person walks clockwise in a large curve while swinging their arms."
22.58s
0.201s
0.033s
0.290s
0.032s
"a person walking like a bird and then sniffing the air."
21.16s
0.192s
0.031s
0.294s
0.034s
"a person is walking like a mummy."
23.77s
0.213s
0.032s
0.286s
0.033s
"cheerfully walking forward with each step."
24.03s
0.227s
0.033s
0.301s
0.031s

Fig. 8: Qualitative comparison of the state-of-the-art methods in the text-to-motion task. With only one-step inference, MotionLCM achieves the fastest motion generation while producing high-quality movements that closely match the textual descriptions. MLD-M and MotionLCM-M exhibit superior motion-text alignment capability and greater expertise in generating motions with rich details, which validates the effectiveness of multi-latent-token learning. Moreover, the inference time for MotionLCM-M (1-step) remains nearly equivalent to that of its predecessor (∼31ms per motion sequence).

denoised dual-part body features are split and fed into their respective VQ-VAE codebooks and decoders to generate a coherent and natural dance motion sequence. The overall framework is illustrated in Figure 7.

Part-based decoupling into the discrete latent space. As shown in Figure 7a, the reconstruction training process of the part-based VQ-VAE is specifically designed to independently encode and decode the motion sequence for the upper and lower body. The input motion sequence $\mathbf{x}_0$ is first split into upper and lower body parts, which are then encoded by the encoders, $\theta_{\mathrm{up}}^{\mathrm{Enc}}$ and $\theta_{\mathrm{low}}^{\mathrm{Enc}}$. These encoders generate the dual-part body features $\mathbf{z}^{\mathrm{up}}$ and $\mathbf{z}^{\mathrm{low}}$, which are then fed into the codebooks for quantization as follows,

$$\mathbf{e}_{\mathrm{q}}^{\mathrm{up}}(i) = \mathop{\arg\min}_{\mathbf{e}^{\mathrm{up}}(j)\in\mathbf{\Sigma}_{\mathrm{up}}} \|\mathbf{z}^{\mathrm{up}}(i)-\mathbf{e}^{\mathrm{up}}(j)\|_2, \tag{13}$$

$$\mathbf{e}_{\mathrm{q}}^{\mathrm{low}}(i) = \mathop{\arg\min}_{\mathbf{e}^{\mathrm{low}}(j)\in\mathbf{\Sigma}_{\mathrm{low}}} \|\mathbf{z}^{\mathrm{low}}(i)-\mathbf{e}^{\mathrm{low}}(j)\|_2, \tag{14}$$

where $\mathbf{\Sigma}_{\mathrm{up}} = \{\mathbf{e}^{\mathrm{up}}(i)\}_{i=1}^{C}$ and $\mathbf{\Sigma}_{\mathrm{low}} = \{\mathbf{e}^{\mathrm{low}}(i)\}_{i=1}^{C}$ are the codebooks with size of C. The quantized features are subsequently fed into the decoders, $\theta_{\mathrm{up}}^{\mathrm{Dec}}$ and $\theta_{\mathrm{low}}^{\mathrm{Dec}}$, to reconstruct the upper and lower body parts, then concatenated to form the complete dance motion sequence $\hat{\mathbf{x}}_0$. We use the following VQ loss $\mathcal{L}_{\mathrm{VQ}}$ to train the part-based VQ-VAE,

$$\mathcal{L}_{\mathrm{VQ}} = \|\hat{\mathbf{x}}_0-\mathbf{x}_0\|_1 + \|\mathrm{sg}[\mathbf{z}_0]-\mathbf{e}_{\mathrm{q}}\|_1 + \lambda_{\mathrm{commit}}\|\mathbf{z}_0-\mathrm{sg}[\mathbf{e}_{\mathrm{q}}]\|_1, \tag{15}$$

where $\mathbf{z}_0$ and $\mathbf{e}_{\mathrm{q}}$ are the concatenated body features (*i.e.*, $\mathbf{z}^{\mathrm{up}}$ and $\mathbf{z}^{\mathrm{low}}$) and quantized features (*i.e.*, $\mathbf{e}_{\mathrm{q}}^{\mathrm{up}}$ and $\mathbf{e}_{\mathrm{q}}^{\mathrm{low}}$). $\lambda_{\mathrm{commit}}$ is the weight of commitment loss and $\mathrm{sg}[\cdot]$ denotes the stopgrad operation. The first term focuses on motion reconstruction, the second term optimizes the codebook for effective quantization, and the third term ensures consistency between the body features and the quantized features. Leveraging the part-based VQ-VAE, we effectively decouple the upper and lower body parts, enabling more expressive latent representations for the next stage of diffusion training.

VQ-based latent diffusion for dance generation. As shown in Figure 7b, our VQ-based motion latent diffusion model (VQ-MLD) is designed to denoise the dual-part body features conditioned on music input. We first apply a forward diffusion operation with t steps to the concatenated encoded features $\mathbf{z}_0$ to obtain the noisy $\mathbf{z}_t$. Then, we use a transformer-based [90] denoiser to perform denoising, conditioned on the time step t and music features $\mathbf{c}$. Following [32], we optimize the denoiser θ using the following diffusion loss $\mathcal{L}_{\mathrm{diff}}$,

$$\mathcal{L}_{\mathrm{diff}} = \mathbb{E}\left[\|\mathbf{z}_0-\hat{\mathbf{z}}_0^{\theta}(\mathbf{z}_t,t,\mathbf{c})\|_2^2\right]. \tag{16}$$

As shown in Figure 7c, we first sample a latent noise $\mathbf{z}_T$ from the standard Gaussian distribution $\mathcal{N}(0,I)$, and then perform multiple iterations of denoising to obtain the clean latent representation $\hat{\mathbf{z}}_0^{\theta}$. This latent representation is then split into upper and lower body features, quantized, and finally

TABLE 1: Comparison of VAE and MLD-M performance for different latent sizes (*i.e.*, compression rates). We conduct three groups of experiments with compression rates of 512, 256, and 128. Within each group, we perform four control experiments to explore the impact of varying the number of latent tokens on reconstruction and generation performance. The results indicate that increasing the number of latent tokens generally enhances the motion reconstruction precision of VAE and additionally improves the motion generation quality of MLD-M. All models are tested using the last checkpoint. MLD-M uses CFG 7.5 and DDIM [36] 50 steps for inference to ensure a fair comparison. Top k is for R-Precision Top k.

Latents	VAE Performance			MLD-M Performance							
	FID ↓	MPJPE ↓	Fea. err. ↓	AITS ↓	Top 1 ↑	Top 2 ↑	Top 3 ↑	FID ↓	MM Dist ↓	Diversity →	MModality ↑
2×256	0.044	22.0	0.272	0.270	$0.523^{\pm .002}$	$0.718^{\pm .003}$	$0.813^{\pm .002}$	$0.255^{\pm .007}$	$2.913^{\pm .007}$	$9.444^{\pm .075}$	$1.816^{\pm .052}$
4×128	0.013	14.6	0.194	0.287	$0.557^{\pm .003}$	$0.749^{\pm .002}$	$0.838^{\pm .002}$	$0.142^{\pm .004}$	$2.763^{\pm .006}$	$9.785^{\pm .087}$	$1.458^{\pm .043}$
8×64	0.008	13.6	0.184	0.291	$0.543^{\pm .003}$	$0.735^{\pm .002}$	$0.828^{\pm .002}$	$0.119^{\pm .005}$	$2.829^{\pm .009}$	$9.720^{\pm .075}$	$1.727^{\pm .048}$
16×32	0.008	14.5	0.199	0.295	$0.546^{\pm .003}$	$0.738^{\pm .003}$	$0.829^{\pm .002}$	$0.076^{\pm .004}$	$2.806^{\pm .007}$	$9.627^{\pm .066}$	$1.668^{\pm .055}$
1×256	0.118	41.2	0.443	0.225	$0.526^{\pm .003}$	$0.718^{\pm .003}$	$0.812^{\pm .002}$	$0.325^{\pm .009}$	$2.920^{\pm .009}$	$9.696^{\pm .081}$	$1.675^{\pm .048}$
2×128	0.049	25.9	0.308	0.275	$0.542^{\pm .002}$	$0.733^{\pm .002}$	$0.823^{\pm .002}$	$0.153^{\pm .005}$	$2.826^{\pm .006}$	$9.707^{\pm .063}$	$1.138^{\pm .036}$
4×64	0.023	21.8	0.269	0.285	$0.547^{\pm .003}$	$0.738^{\pm .003}$	$0.828^{\pm .002}$	$0.137^{\pm .004}$	$2.801^{\pm .008}$	$9.763^{\pm .093}$	$1.335^{\pm .044}$
8×32	0.022	21.4	0.273	0.290	$0.545^{\pm .003}$	$0.737^{\pm .003}$	$0.827^{\pm .002}$	$0.102^{\pm .004}$	$2.820^{\pm .008}$	$9.744^{\pm .079}$	$1.494^{\pm .057}$
1×128	0.121	45.0	0.477	0.225	$0.506^{\pm .003}$	$0.695^{\pm .003}$	$0.792^{\pm .002}$	$0.378^{\pm .011}$	$3.036^{\pm .009}$	$9.408^{\pm .108}$	$1.013^{\pm .028}$
2×64	0.066	36.4	0.393	0.269	$0.539^{\pm .003}$	$0.731^{\pm .003}$	$0.822^{\pm .002}$	$0.167^{\pm .006}$	$2.864^{\pm .008}$	$9.726^{\pm .091}$	$1.104^{\pm .037}$
4×32	0.048	32.9	0.377	0.284	$0.539^{\pm .003}$	$0.727^{\pm .003}$	$0.819^{\pm .002}$	$0.110^{\pm .004}$	$2.858^{\pm .009}$	$9.629^{\pm .068}$	$1.281^{\pm .037}$
8×16	0.045	36.9	0.405	0.289	$0.524^{\pm .002}$	$0.714^{\pm .002}$	$0.809^{\pm .002}$	$0.091^{\pm .003}$	$2.929^{\pm .008}$	$9.581^{\pm .057}$	$1.395^{\pm .048}$

decoded by their respective VQ-VAE decoders. The decoded outputs are concatenated to generate the final natural dance motion sequence. Leveraging the pre-trained VQ-MLD, we successfully perform latent consistency distillation for the music-to-dance task. Our MotionLCM achieves real-time music-to-dance generation, producing high-quality dance sequences with exceptional motion expressiveness.

4 Experiments

4.1 Controllable Text-to-Motion Generation

4.1.1 Experimental setup

Datasets. We experiment on the popular HumanML3D [1] dataset, featuring 14,616 unique human motion sequences with 44,970 textual descriptions. For a fair comparison with previous methods [1], [2], [4]–[6], we take the redundant motion representation, including root velocity, root height, local joint positions, velocities, rotations in root space, and the foot contact binary labels.

Evaluation metrics. We extend the evaluation metrics of previous works [1], [6], [38]. **(1) Time cost:** We follow [6] to report the Average Inference Time per Sentence (AITS) to evaluate the inference efficiency of models. **(2) Motion quality:** Frechet Inception Distance (FID) is adopted as a principal metric to evaluate the feature distributions between the generated and real motions. The feature extractor employed is from [1]. **(3) Motion diversity:** MultiModality (MModality) measures the generation diversity conditioned on the same text and Diversity calculates variance through features [1]. **(4) Condition matching:** Following [1], we calculate the motion-retrieval precision (R-Precision) to report the text-motion Top-1/2/3 matching accuracy and Multimodal Distance (MM Dist) calculates the mean distance between motions and texts. **(5) Control error:** Trajectory error (Traj. err.) quantifies the ratio of unsuccessful trajectories, characterized by any control joint location error surpassing a predetermined threshold. Location error (Loc. err.) represents the unsuccessful joints. Average error (Avg. err.) denotes the mean location error of the control joints. **(6) Reconstruction error:** MPJPE measures the average Euclidean distance between reconstructed and ground truth joint positions after aligning the root (pelvis). Feature error (Fea. err.) refers to the L2 norm between the reconstructed motion features and the real motion features.

Implementation details. Our baseline motion diffusion model is based on MLD [6]. We reproduce MLD with higher performance. Unless otherwise specified, all our experiments are conducted on this model. For MotionLCM, we employ the AdamW [91] optimizer for 96K iterations using a cosine decay learning rate scheduler and 1K iterations of linear warm-up. A batch size of 256 and a learning rate of 2e-4 are used. We set the training guidance scale range as $[w_{\text{min}}, w_{\text{max}}] = [5, 15]$, with the testing guidance scale set to 7.5, and adopt the EMA rate $\mu = 0.95$ by default. We use the DDIM [36] solver with skipping interval $k = 20$ and choose the Huber [89] loss as the distance measuring function d. For multi-latent-token training, the VAE and MLD-M are trained using the AdamW [91] optimizer for 6K and 3K epochs, respectively, with a cosine decay learning rate scheduler and 1K iterations of linear warm-up. The learning rates are set to 2e-4 and 1e-4, with batch sizes of 128 and 64, respectively. For MotionLCM-M, we adopt a batch size of 128 and 192K training iterations. We increase the skipping interval k to 100 and dynamically select the testing guidance scale based on the number of inference steps. The remaining settings are consistent with those of its predecessor. For motion ControlNet, we use the AdamW [91] optimizer for 192K iterations with 1K iterations of linear warm-up. The batch size and learning rate are set to 128 and 1e-4. The learning rate scheduler is the same as the first stage. For the training objective, we employ d as the L2 loss and set the control loss weight λ to 1.0 by default. We set the control ratio τ as 0.25 and the number of control joints as $K = 6$ (*i.e.*, *Pelvis*, *Left foot*, *Right foot*, *Head*, *Left wrist*, and *Right wrist*) in both training and testing. We adopt the experimental settings from OmniControl [38] to perform the joint-based control experiments, where each of the K control joints mentioned above is tested individually. For consistency latent tuning,

TABLE 2: Comparison of text-conditional motion synthesis on HumanML3D [1] dataset. We compute suggested metrics following [1]. We repeat the evaluation 20 times for each metric and report the average with a 95% confidence interval. “→” indicates that the closer to the real data, the better. **Bold** and $\underline{\text{underline}}$ indicate the best and the second best result. “*” denotes the reproduced version of MLD [6]. The MotionLCM in **one-step inference (30ms)** surpasses most state-of-the-art models [2], [4]–[6]. The generation performance of **MLD-M** and **MotionLCM-M** are significantly improved compared to their predecessors, achieving the state-of-the-art on this challenging benchmark.

Methods	AITS ↓	R-Precision ↑			FID ↓	MM Dist ↓	Diversity →	MModality ↑
		Top 1	Top 2	Top 3				
Real	-	$0.511^{\pm.003}$	$0.703^{\pm.003}$	$0.797^{\pm.002}$	$0.002^{\pm.000}$	$2.974^{\pm.008}$	$9.503^{\pm.065}$	-
Seq2Seq [52]	-	$0.180^{\pm.002}$	$0.300^{\pm.002}$	$0.396^{\pm.002}$	$11.75^{\pm.035}$	$5.529^{\pm.007}$	$6.223^{\pm.061}$	-
JL2P [53]	-	$0.246^{\pm.002}$	$0.387^{\pm.002}$	$0.486^{\pm.002}$	$11.02^{\pm.046}$	$5.296^{\pm.008}$	$7.676^{\pm.058}$	-
T2G [54]	-	$0.165^{\pm.001}$	$0.267^{\pm.002}$	$0.345^{\pm.002}$	$7.664^{\pm.030}$	$6.030^{\pm.008}$	$6.409^{\pm.071}$	-
Hier [55]	-	$0.301^{\pm.002}$	$0.425^{\pm.002}$	$0.552^{\pm.004}$	$6.532^{\pm.024}$	$5.012^{\pm.018}$	$8.332^{\pm.042}$	-
TEMOS [2]	0.017	$0.424^{\pm.002}$	$0.612^{\pm.002}$	$0.722^{\pm.002}$	$3.734^{\pm.028}$	$3.703^{\pm.008}$	$8.973^{\pm.071}$	$0.368^{\pm.018}$
T2M [1]	0.038	$0.457^{\pm.002}$	$0.639^{\pm.003}$	$0.740^{\pm.003}$	$1.067^{\pm.002}$	$3.340^{\pm.008}$	$9.188^{\pm.002}$	$2.090^{\pm.083}$
TM2T [3]	0.760	$0.424^{\pm.003}$	$0.618^{\pm.003}$	$0.729^{\pm.002}$	$1.501^{\pm.017}$	$3.467^{\pm.011}$	$8.589^{\pm.076}$	$\underline{2.424}^{\pm.093}$
MotionDiffuse [4]	14.74	$0.491^{\pm.001}$	$0.681^{\pm.001}$	$0.782^{\pm.001}$	$0.630^{\pm.001}$	$3.113^{\pm.001}$	$9.410^{\pm.049}$	$1.553^{\pm.042}$
MDM [5]	24.74	$0.320^{\pm.005}$	$0.498^{\pm.004}$	$0.611^{\pm.007}$	$0.544^{\pm.044}$	$5.566^{\pm.027}$	$\underline{9.559}^{\pm.086}$	$\mathbf{2.799}^{\pm.072}$
MLD [6]	0.217	$0.481^{\pm.003}$	$0.673^{\pm.003}$	$0.772^{\pm.002}$	$0.473^{\pm.013}$	$3.196^{\pm.010}$	$9.724^{\pm.082}$	$2.413^{\pm.079}$
T2M-GPT [7]	0.380	$0.492^{\pm.003}$	$0.679^{\pm.002}$	$0.775^{\pm.002}$	$0.141^{\pm.005}$	$3.121^{\pm.009}$	$9.722^{\pm.082}$	$1.831^{\pm.048}$
ReMoDiffuse [8]	0.624	$0.510^{\pm.005}$	$0.698^{\pm.006}$	$0.795^{\pm.004}$	$0.103^{\pm.004}$	$2.974^{\pm.016}$	$9.018^{\pm.075}$	$1.795^{\pm.043}$
MoMask [9]	0.120	$0.521^{\pm.002}$	$0.713^{\pm.002}$	$0.807^{\pm.002}$	$\mathbf{0.045}^{\pm.002}$	$2.958^{\pm.008}$	$9.675^{\pm.068}$	$1.241^{\pm.040}$
StableMoFusion [10]	0.499	$\mathbf{0.553}^{\pm.003}$	$\mathbf{0.748}^{\pm.002}$	$\mathbf{0.841}^{\pm.002}$	$0.098^{\pm.003}$	$2.770^{\pm.006}$	$9.748^{\pm.092}$	$1.774^{\pm.051}$
MotionCLR [11]	0.343	$0.542^{\pm.001}$	$0.733^{\pm.002}$	$0.827^{\pm.003}$	$0.099^{\pm.003}$	$2.981^{\pm.011}$	$9.846^{\pm.080}$	$2.145^{\pm.043}$
MLD* [6]	0.225	$0.504^{\pm.002}$	$0.698^{\pm.003}$	$0.796^{\pm.002}$	$0.450^{\pm.011}$	$3.052^{\pm.009}$	$9.634^{\pm.064}$	$2.267^{\pm.082}$
MotionLCM (1-step)	**0.030**	$0.502^{\pm.003}$	$0.701^{\pm.002}$	$0.803^{\pm.002}$	$0.467^{\pm.012}$	$3.022^{\pm.009}$	$9.631^{\pm.066}$	$2.172^{\pm.082}$
MotionLCM (2-step)	0.035	$0.505^{\pm.003}$	$0.705^{\pm.002}$	$0.805^{\pm.002}$	$0.368^{\pm.011}$	$2.986^{\pm.008}$	$9.640^{\pm.052}$	$2.187^{\pm.094}$
MotionLCM (4-step)	0.043	$0.502^{\pm.003}$	$0.698^{\pm.002}$	$0.798^{\pm.002}$	$0.304^{\pm.012}$	$3.012^{\pm.007}$	$9.607^{\pm.066}$	$2.259^{\pm.092}$
B2A-HDM [94]	-	$0.511^{\pm.002}$	$0.699^{\pm.002}$	$0.791^{\pm.002}$	$0.084^{\pm.004}$	$3.020^{\pm.010}$	$\mathbf{9.526}^{\pm.080}$	$1.914^{\pm.078}$
MLD-M (CFG=7.5)	0.295	$0.548^{\pm.003}$	$0.738^{\pm.003}$	$0.829^{\pm.002}$	$0.073^{\pm.003}$	$2.810^{\pm.008}$	$9.658^{\pm.089}$	$1.675^{\pm.055}$
MLD-M (CFG=12.5)	0.295	$0.544^{\pm.003}$	$0.736^{\pm.002}$	$0.827^{\pm.002}$	$\underline{0.049}^{\pm.002}$	$2.828^{\pm.007}$	$9.531^{\pm.087}$	$1.672^{\pm.051}$
MotionLCM-M (1-step)	$\underline{0.031}$	$0.546^{\pm.003}$	$0.743^{\pm.002}$	$\underline{0.837}^{\pm.002}$	$0.072^{\pm.003}$	$\underline{2.767}^{\pm.007}$	$9.577^{\pm.070}$	$1.858^{\pm.056}$
MotionLCM-M (2-step)	0.038	$\underline{0.551}^{\pm.003}$	$0.745^{\pm.002}$	$0.836^{\pm.002}$	$\underline{0.049}^{\pm.003}$	$\mathbf{2.765}^{\pm.008}$	$9.584^{\pm.066}$	$1.833^{\pm.052}$
MotionLCM-M (4-step)	0.050	$\mathbf{0.553}^{\pm.003}$	$\underline{0.746}^{\pm.002}$	$\underline{0.837}^{\pm.002}$	$0.056^{\pm.003}$	$2.773^{\pm.009}$	$9.598^{\pm.067}$	$1.758^{\pm.056}$

we use Adam [92] optimizer for 400 iterations with 50 iterations of linear warm-up. The learning rate is set to 0.1 and gradient clipping is adopted. The latent decorrelation loss weight λ_{decorr} is set to 100 by default. We implement our model using PyTorch [93] with training on an NVIDIA RTX 4090 GPU and testing on a Tesla V100 GPU.

4.1.2 Explorations of Multi-Latent-Token Diffusion

The best practice for training motion latent diffusion model is first to obtain a semantically rich and compact latent space. Accordingly, the VAE must balance the compression rate (*i.e.*, latent size) while enhancing its motion reconstruction capability. To validate the effectiveness of our proposed latent adapter for high-performance multi-latent-token diffusion (MLD-M), we conduct extensive exploratory experiments in Table 1. Specifically, we divide the experiments into three groups, each representing the same compression rate (*i.e.*, 128, 256, and 512). We observe that under the same compression rate, increasing the number of latent tokens continuously improves the motion reconstruction precision of the VAE, which enhances the motion generation quality of MLD-M (*i.e.*, FID score). Moreover, increasing the number of latent tokens slightly increases the inference time AITS but remains within an acceptable range (0.2s∼0.3s). Additionally, the motion-text matching performance fluctuates accordingly. Therefore, considering both the text alignment capability and motion generation quality of MLD-M, we select the model with the latent size of 16×32 as our reported MLD-M. The reported MotionLCM-M is distilled from the best FID checkpoint of MLD-M (16x32).

4.1.3 Comparisons on Text-to-motion

In the following part, we first evaluate our MotionLCM on the text-to-motion (T2M) task. We compare our method with some T2M baselines on HumanML3D [1] with suggested metrics [1] under the 95% confidence interval from 20 times running. As MotionLCM is based on MLD, we mainly focus on the performance compared with MLD. For evaluating time efficiency, we compare the Average Inference Time per Sentence (AITS) with previous methods [1]–[11]. The results are borrowed from previous works [6], [9]–[12]. The deterministic methods [52]–[55], are unable to produce diverse results from a single input text and thus we leave their MModality metrics empty. For the quantitative results,

TABLE 3: Comparison of initial-motion-based motion control on HumanML3D [1] dataset. **Bold** and underline indicate the best and the second best result. Our MotionLCM outperforms OmniControl [38] and MLD [6] regarding generation quality, control performance, and inference speed. MotionLCM-M further extends the leading performance. "LC" and "MC" refer to the control supervision introduced in the latent space and motion space.

Methods	AITS ↓	FID ↓	R-Precision ↑	Diversity →	Traj. err. ↓	Loc. err. ↓	Avg. err. ↓
			Top 3		(50cm)	(50cm)	
Real	-	0.002	0.797	9.503	0.0000	0.0000	0.0000
OmniControl [38]	81.00	2.328	0.557	8.867	0.3362	0.0322	0.0977
MLD [6] (LC)	0.552	0.469	0.723	**9.476**	0.4230	0.0653	0.1690
MotionLCM (1-step, LC)	**0.042**	0.319	0.752	9.424	0.2986	0.0344	0.1410
MotionLCM (2-step, LC)	0.047	0.315	0.770	9.427	0.2840	0.0328	0.1365
MotionLCM (4-step, LC)	0.063	0.328	0.745	9.441	0.2973	0.0339	0.1398
MotionLCM-M (1-step, LC)	0.044	0.265	0.791	9.689	0.0522	0.0080	0.0659
MotionLCM-M (2-step, LC)	0.050	0.209	0.800	9.779	0.0525	0.0082	0.0664
MotionLCM-M (4-step, LC)	0.066	**0.195**	**0.803**	9.839	0.0537	0.0084	0.0666
MLD [6] (LC&MC)	0.552	0.555	0.754	9.373	0.2722	0.0215	0.1265
MotionLCM (1-step, LC&MC)	**0.042**	0.419	0.756	9.390	0.1988	0.0147	0.1127
MotionLCM (2-step, LC&MC)	0.047	0.397	0.759	9.469	0.1960	0.0143	0.1092
MotionLCM (4-step, LC&MC)	0.063	0.444	0.753	9.355	0.2089	0.0172	0.1140
MotionLCM-M (1-step, LC&MC)	0.044	0.300	0.789	9.774	**0.0274**	**0.0023**	0.0505
MotionLCM-M (2-step, LC&MC)	0.050	0.248	0.799	9.828	**0.0274**	0.0024	**0.0503**
MotionLCM-M (4-step, LC&MC)	0.066	0.236	0.802	9.882	0.0277	0.0024	0.0504

as shown in Table 2, our MotionLCM boasts an impressive real-time runtime efficiency, averaging around **30ms per motion sequence** during inference. This performance exceeds that of previous diffusion-based methods [4]–[6] and even surpasses MLD [6] by an order of magnitude. Furthermore, despite employing only one-step inference, our MotionLCM can approximate or even surpass the performance of MLD [6] (DDIM [36] 50 steps). With two-step inference, we achieve the best R-Precision and MM Dist metrics, while increasing the sampling steps to four yields the best FID. The above results demonstrate the effectiveness of latent consistency distillation. In addition, we propose MLD-M, which achieves a significant quantitative improvement in motion generation performance compared to its predecessor, MLD [6]. Moreover, unlike B2A-HDM [94], which relies on an overcomplicated multi-denoiser framework to solve the single-latent-token limitation, MLD-M surpasses B2A-HDM by a large margin while using only one single denoiser. Thanks to the powerful MLD-M, the distillation performance of MotionLCM-M has also been significantly improved, further advancing the state of text-to-motion generation by excelling in inference speed, motion generation quality, and text alignment capability. For the qualitative results, as shown in Figure 8, MotionLCM not only accelerates motion generation to real-time speed but also delivers high-quality outputs, closely aligning with the textual descriptions. MLD-M and MotionLCM-M demonstrate substantial improvement in motion-text alignment capability while showcasing more abundant motion details.

4.1.4 Comparisons on Controllable Motion Generation

For initial-motion-based control, as shown in Table 3, we compare our MotionLCM with OmniControl [38] and MLD [6]. We observe that OmniControl struggles with multi-joint control and falls short in both generation quality and control performance compared to MotionLCM. To verify the effectiveness of the latent generated by our MotionLCM for training motion ControlNet, we conducted the following two sets of experiments: "LC" and "MC", which indicate introducing control supervision in the latent space and motion space. It can be observed that under the same experimental settings, MotionLCM maintains higher fidelity and significantly outperforms MLD [6] in motion control performance. This demonstrates that the latent generated by MotionLCM is more effective for training motion ControlNet compared to MLD [6]. In terms of inference speed, MotionLCM (1-step) is **1929**× faster compared to OmniControl [38] and **13**× faster than MLD [6]. Moreover, our MotionLCM-M enhances control performance by an order of magnitude compared to its predecessor, while delivering superior motion quality (FID) and improved motion-text alignment (R-Precision). It also maintains inference speed comparable to its predecessor, ranging from 40ms to 70ms per motion sequence. For joint-based control, as shown in Table 4, it can be observed that relying solely on motion ControlNet makes it difficult for our MotionLCM-M to achieve precise joint control. However, our proposed consistency latent tuning method outperforms OmniControl in terms of motion generation quality, control precision, and inference speed, fully validating its effectiveness. For qualitative results, as shown in Figure 9, OmniControl fails to control the initial poses in the first example and does not generate motion that aligns with the text in the second case. However, our MotionLCM not only adheres to the control of the initial poses but also generates motions that match the textual descriptions. Additionally, our MotionLCM-M achieves superior control accuracy compared to its predecessor, particularly in the first and fourth control examples.

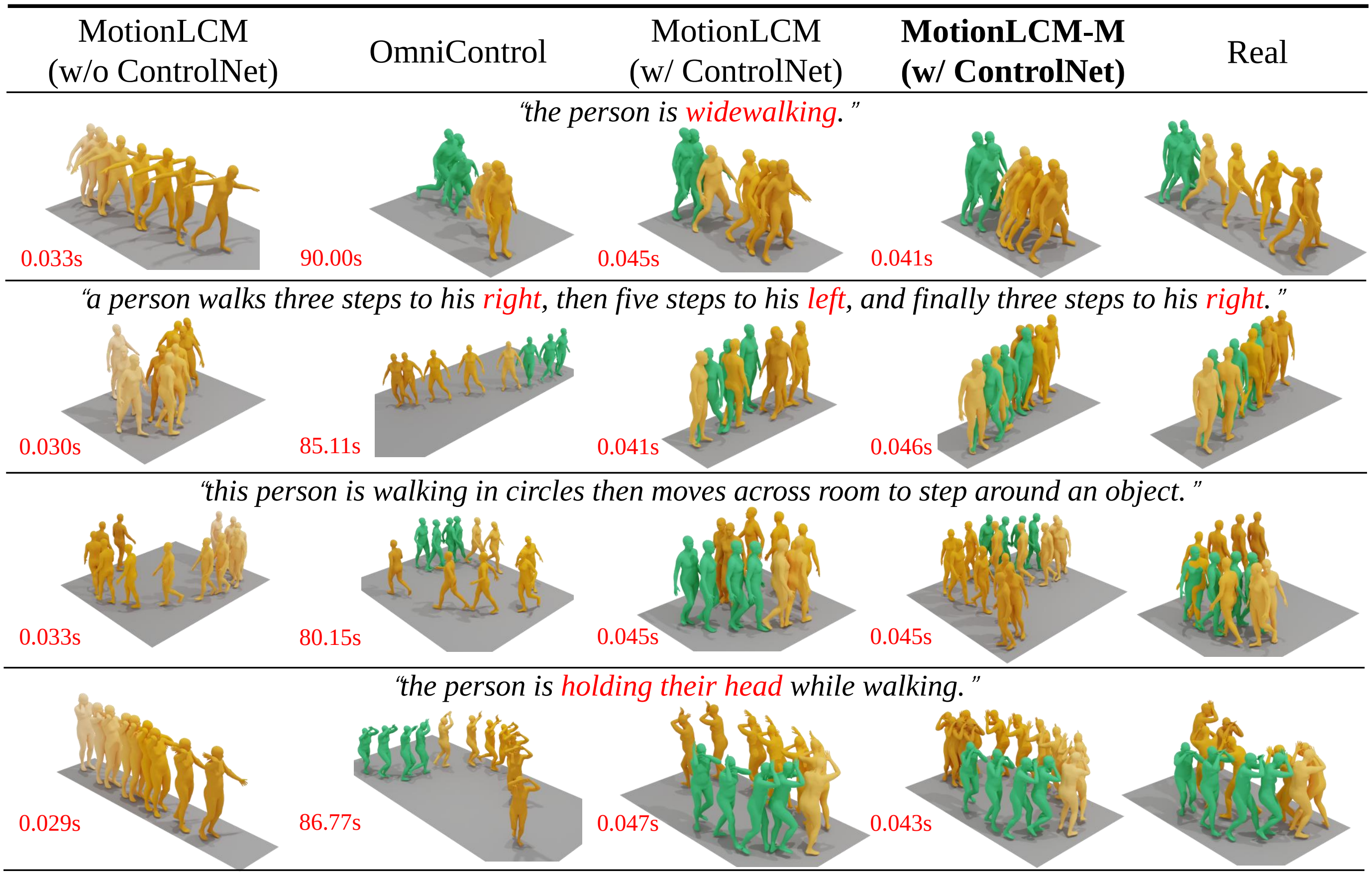

MotionLCM (w/o ControlNet)
OmniControl
MotionLCM (w/ ControlNet)
MotionLCM-M (w/ ControlNet)
Real
"the person is widewalking."
0.033s
90.00s
0.045s
0.041s
"a person walks three steps to his right, then five steps to his left, and finally three steps to his right."
0.030s
85.11s
0.041s
0.046s
"this person is walking in circles then moves across room to step around an object."
0.033s
80.15s
0.045s
0.045s
"the person is holding their head while walking."
0.029s
86.77s
0.047s
0.043s

Fig. 9: Qualitative comparison of the state-of-the-art methods in the initial-motion-based motion control task. We provide the visualized motion results and real references from four prompts. Compared to OmniControl [38], MotionLCM with ControlNet not only generates the initial poses that accurately follow the given multi-joint trajectories (*i.e.*, the green poses in real references) but also produces motions that closely align with the texts. Additionally, our MotionLCM-M offers better control accuracy than its predecessor, especially in the first and fourth control examples.

4.2 Music-to-Dance

4.2.1 Experimental setup

Datasets. We conduct experiments on the AIST++ [73] dataset, which contains 992 dance motions, each paired with its corresponding music. Following the previous work [75], 952 motion sequences are used for training, while the remaining 40 are reserved for evaluation.

Evaluation metrics. We evaluate the dance generation models from four aspects. First, similar to the T2M task, we use AITS to measure the inference speed. Second, Beat Alignment Score (BAS [73]) is utilized to quantitatively assess the motion-music correlation, based on the similarity between the kinematic beats and music beats. Lastly, for motion quality and diversity, we employ the extracted kinematic features [95] and geometric features [96] to calculate the FID (*i.e.*, FID$_k$ and FID$_g$) and Diversity (*i.e.*, Div$_k$ and Div$_g$).

Implementation details. For our part-based VQ-VAE, we follow the previous work [7] to adopt a standard CNN-based architecture with 1D convolutions and residual blocks to construct the encoders and decoders. The temporal downsampling scale is 16 and the codebook size C is set to 512. The dimension size of the codebook entries is 64. We employ the AdamW [91] optimizer to train the part-based VQ-VAE for 300 epochs. The batch size and learning rate are set to 128 and 3e-5. The commitment loss weight λ_{commit} is set to 0.02 by default. For VQ-MLD and MotionLCM, we use the AdamW [91] optimizer for 4000 and 700 epochs of training. Both models employ a cosine decay learning rate scheduler with 1K iterations of linear warm-up. The learning rate is set to 2e-4 with batch sizes of 128 and 256. In the training of MotionLCM, we set the training guidance scale range as $[w_{\text{min}}, w_{\text{max}}] = [6, 12]$ with the testing guidance scale set to 15 and the EMA rate $\mu = 0.95$ by default. We adopt the DDIM [36] solver with skipping interval $k = 20$ and choose the Huber [89] loss as the distance measuring function d. We implement our model using PyTorch [93] with training and testing on an NVIDIA RTX 4090 GPU.

4.2.2 Comparisons on Music-to-Dance

As shown in Table 5, we present a quantitative comparison of our VQ-MLD and MotionLCM with existing state-of-the-art methods [70]–[73], [75], [76], [78]. The results are borrowed from the previous works [75], [78]. Since the primary contribution of our MotionLCM is to accelerate the inference speed of diffusion models, we mainly focus on comparing diffusion-based methods (*i.e.*, Lodge [78] and EDGE [76]). We observe that our VQ-MLD outperforms both Lodge and EDGE in terms of motion generation quality and diversity, while also producing dance sequences with better beat alignment. Furthermore, VQ-MLD achieves a **10**× faster inference speed compared to Lodge and EDGE, effectively

TABLE 4: Comparison of joint-based motion control on HumanML3D [1] dataset. We observe that relying solely on motion ControlNet is less effective for sparse control signals compared to dense initial-motion-based motion control. In contrast, our consistency latent tuning (CLT) method handles such sparse control signals well and outperforms OmniControl [38] in terms of motion generation quality, control precision, and inference speed (**5**× faster). **Bold** indicate the best result.

Methods	Joint	AITS ↓	FID ↓	R-Precision ↑ Top 3	Diversity →	Traj. err. ↓ (50cm)	Loc. err. ↓ (50cm)	Avg. err. ↓
Real	-	-	0.002	0.797	9.503	0.0000	0.0000	0.0000
MDM [5]	Pelvis	16.34	0.698	0.602	9.197	0.4022	0.3076	0.5959
PriorMDM [23]		20.19	0.475	0.583	9.156	0.3457	0.2132	0.4417
GMD [37]		137.63	0.576	0.665	9.206	0.0931	0.0321	0.1439
OmniControl [38]		81.00	0.218	0.687	**9.422**	0.0387	0.0096	0.0338
MotionLCM-M (w/ ControlNet)		**0.066**	0.393	0.787	9.837	0.1080	0.0581	0.1386
MotionLCM-M (w/ CLT)		13.68	**0.202**	**0.816**	9.707	**0.0039**	**0.0003**	**0.0203**
OmniControl [38]	Pelvis	81.00	0.322	0.691	**9.545**	0.0404	0.0085	0.0367
MotionLCM-M (w/ ControlNet)		**0.066**	0.475	0.779	10.013	0.1617	0.0841	0.1838
MotionLCM-M (w/ CLT)		13.68	**0.202**	**0.816**	9.707	**0.0039**	**0.0003**	**0.0203**
OmniControl [38]	Left foot	81.00	0.280	0.696	**9.553**	0.0594	0.0094	0.0314
MotionLCM-M (w/ ControlNet)		**0.066**	0.498	0.779	9.984	0.2607	0.1229	0.2304
MotionLCM-M (w/ CLT)		13.68	**0.207**	**0.823**	9.622	**0.0113**	**0.0005**	**0.0218**
OmniControl [38]	Right foot	81.00	0.319	0.701	**9.481**	0.0666	0.0120	0.0334
MotionLCM-M (w/ ControlNet)		**0.066**	0.467	0.782	9.975	0.2459	0.1127	0.2278
MotionLCM-M (w/ CLT)		13.68	**0.218**	**0.818**	9.641	**0.0123**	**0.0006**	**0.0219**
OmniControl [38]	Head	81.00	0.335	0.696	**9.480**	0.0422	0.0079	0.0349
MotionLCM-M (w/ ControlNet)		**0.066**	0.449	0.782	9.962	0.1971	0.0977	0.2136
MotionLCM-M (w/ CLT)		13.68	**0.179**	**0.817**	9.694	**0.0027**	**0.0002**	**0.0214**
OmniControl [38]	Left wrist	81.00	0.304	0.680	**9.436**	0.0801	0.0134	0.0529
MotionLCM-M (w/ ControlNet)		**0.066**	0.404	0.789	9.999	0.3965	0.1912	0.3150
MotionLCM-M (w/ CLT)		13.68	**0.194**	**0.822**	9.614	**0.0105**	**0.0005**	**0.0265**
OmniControl [38]	Right wrist	81.00	0.299	0.692	**9.519**	0.0813	0.0127	0.0519
MotionLCM-M (w/ ControlNet)		**0.066**	0.418	0.786	10.012	0.3822	0.1806	0.3079
MotionLCM-M (w/ CLT)		13.68	**0.204**	**0.821**	9.656	**0.0094**	**0.0007**	**0.0265**
OmniControl [38]	Average	81.00	0.310	0.693	**9.502**	0.0617	0.0107	0.0404
MotionLCM-M (w/ ControlNet)		**0.066**	0.452	0.783	9.991	0.2740	0.1315	0.2464
MotionLCM-M (w/ CLT)		13.68	**0.201**	**0.820**	9.656	**0.0084**	**0.0005**	**0.0231**

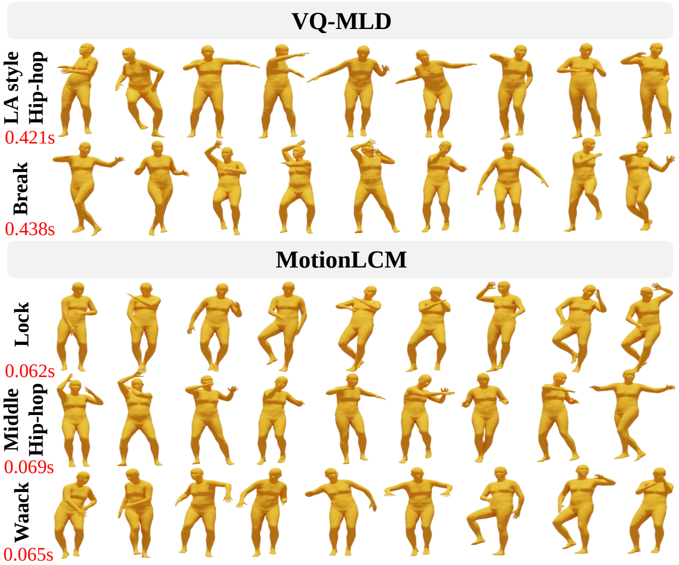


Fig. 10: Qualitative results of VQ-MLD and MotionLCM in the music-to-dance task. We provide visualized results for five different dance genres, demonstrating that our models are capable of generating impressive dance motion sequences. Moreover, the generated dance movements align well with the given music styles.

demonstrating the efficacy of our proposed dual-part VQ-based latent diffusion framework. Building on the powerful VQ-MLD, our MotionLCM achieves an inference speed an order of magnitude faster than VQ-MLD (50ms∼80ms per motion sequence), while also achieving state-of-the-art performance in motion quality and beat alignment. This demonstrates the effectiveness of MotionLCM for real-time music-to-dance generation. For the qualitative results, as shown in Figure 10, our models can generate diverse dance movements based on different types of music, with the motion rhythms aligning well with the beats of the music. Moreover, our MotionLCM significantly outperforms previous approaches in terms of inference speed, achieving real-time music-to-dance generation.

5 Conclusion

In this paper, we extend our ECCV'24 conference paper [12] with three key contributions. First, we propose MotionLCM-M, which incorporates a latent adapter to directly control the VAE compression rate, addressing the lack of expressive motion details caused by suboptimal latent space and enabling a more compact and expressive latent space for multi-latent-token consistency distillation. Second, we introduce consistency latent tuning, a mechanism that iteratively

TABLE 5: Comparison of music-to-dance on AIST++ [73] test set. We present the results of (1, 2, 4)-step inference. To ensure a fair comparison, we evaluate the average runtime of each model by generating a dance sequence consisting of 1200 frames. "*" denotes the diffusion-based method. **Bold** and underline indicate the best and the second best result.

Method	AITS ↓	Motion Quality		Motion Diversity		BAS ↑
		FID_k ↓	FID_g ↓	Div_k ↑	Div_g ↑	
Real	-	17.10	10.60	8.19	7.45	0.2374
Li *et al.* [70]	5.589	86.43	43.46	6.85	3.32	0.1607
DanceNet [71]	8.335	69.18	25.49	2.86	2.85	0.1430
DanceRevolution [72]	0.308	73.42	25.92	3.52	4.87	0.1950
FACT [73]	23.21	35.35	22.11	5.94	6.18	0.2209
Bailando [75]	5.211	28.16	**9.62**	**7.83**	6.34	0.2332
EDGE* [76]	3.995	42.16	22.12	3.96	4.61	0.2334
Lodge* [78]	4.671	37.09	18.79	5.58	4.85	0.2423
VQ-MLD*	0.446	30.38	17.82	6.91	**9.08**	0.2549
MotionLCM (1-step)	**0.058**	52.29	13.78	3.45	5.38	0.2358
MotionLCM (2-step)	0.064	42.89	13.99	4.07	6.06	0.2229
MotionLCM (4-step)	0.078	**26.03**	13.70	5.22	5.67	**0.2874**

refines the learnable latent noise using gradients of error derived from the motion space, effectively handling sparse control signals while preserving the naturalness of generated motions. Finally, we extend our method to the music-to-dance task by developing a VQ-based motion latent diffusion model (VQ-MLD) that jointly captures upper and lower body dynamics, achieving state-of-the-art performance with real-time inference speed. As the highlight of the entire paper, we break the curse of single-latent-token diffusion in an extremely minimalist way, enabling the scaling of the denoiser model. This marks an important step towards future exploration, with a focus on **big data and big model**. At the same time, we remain dedicated to open-sourcing our work to serve the community, staying true to our mission.

Acknowledgements

The research is supported by Shenzhen Science and Technology Program under grant JCYJ20240813111903006. This project is also supported by Shanghai Artificial Intelligence Laboratory. The author team would like to acknowledge Siyu Zhou, Yiming Xie, Zhiyang Dou, and Shunlin Lu for their helpful technical discussions and suggestions.

References

[1] 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 *CVPR*, 2022, pp. 5152–5161.

[2] M. Petrovich, M. J. Black, and G. Varol, "Temos: Generating diverse human motions from textual descriptions," in *ECCV*, 2022, pp. 480–497.

[3] C. Guo, X. Zuo, S. Wang, and L. Cheng, "Tm2t: Stochastic and tokenized modeling for the reciprocal generation of 3d human motions and texts," in *ECCV*, 2022, pp. 580–597.

[4] 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.

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

[6] 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 *CVPR*, 2023, pp. 18 000–18 010.

[7] J. Zhang, Y. Zhang, X. Cun, Y. Zhang, H. Zhao, H. Lu, X. Shen, and Y. Shan, "Generating human motion from textual descriptions with discrete representations," in *CVPR*, 2023, pp. 14 730–14 740.

[8] M. Zhang, X. Guo, L. Pan, Z. Cai, F. Hong, H. Li, L. Yang, and Z. Liu, "Remodiffuse: Retrieval-augmented motion diffusion model," in *ICCV*, 2023, pp. 364–373.

[9] C. Guo, Y. Mu, M. G. Javed, S. Wang, and L. Cheng, "Momask: Generative masked modeling of 3d human motions," in *CVPR*, 2024, pp. 1900–1910.

[10] Y. Huang, H. Yang, C. Luo, Y. Wang, S. Xu, Z. Zhang, M. Zhang, and J. Peng, "Stablemofusion: Towards robust and efficient diffusion-based motion generation framework," *arXiv preprint arXiv:2405.05691*, 2024.

[11] L.-H. Chen, W. Dai, X. Ju, S. Lu, and L. Zhang, "Motionclr: Motion generation and training-free editing via understanding attention mechanisms," *arxiv:2410.18977*, 2024.

[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," in *ECCV*, 2025, pp. 390–408.

[13] H. Ahn, T. Ha, Y. Choi, H. Yoo, and S. Oh, "Text2action: Generative adversarial synthesis from language to action," in *ICRA*, 2018, pp. 5915–5920.

[14] S. Lu, L.-H. Chen, A. Zeng, J. Lin, R. Zhang, L. Zhang, and H.-Y. Shum, "Humantomato: Text-aligned whole-body motion generation," *arXiv preprint arXiv:2310.12978*, 2023.

[15] Z. Wang, Y. Chen, T. Liu, Y. Zhu, W. Liang, and S. Huang, "Humanise: Language-conditioned human motion generation in 3d scenes," *NeurIPS*, pp. 14 959–14 971, 2022.

[16] Z. Xiao, T. Wang, J. Wang, J. Cao, W. Zhang, B. Dai, D. Lin, and J. Pang, "Unified human-scene interaction via prompted chain-of-contacts," in *ICLR*, 2024.

[17] X. Lin and M. R. Amer, "Human motion modeling using dvgans," *arXiv preprint arXiv:1804.10652*, 2018.

[18] 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 *ACMMM*, 2020, pp. 2021–2029.

[19] M. Petrovich, M. J. Black, and G. Varol, "Action-conditioned 3d human motion synthesis with transformer vae," in *ICCV*, 2021, pp. 10 985–10 995.

[20] N. Athanasiou, M. Petrovich, M. J. Black, and G. Varol, "Teach: Temporal action composition for 3d humans," in *3DV*, 2022, pp. 414–423.

[21] R. Dabral, M. H. Mughal, V. Golyanik, and C. Theobalt, "Mofusion: A framework for denoising-diffusion-based motion synthesis," in *CVPR*, 2023, pp. 9760–9770.

[22] Y. Yuan, J. Song, U. Iqbal, A. Vahdat, and J. Kautz, "Physdiff: Physics-guided human motion diffusion model," in *ICCV*, 2023, pp. 16 010–16 021.

[23] Y. Shafir, G. Tevet, R. Kapon, and A. H. Bermano, "Human motion diffusion as a generative prior," in *ICLR*, 2024.

[24] J. Liu, W. Dai, C. Wang, Y. Cheng, Y. Tang, and X. Tong, "Plan, posture and go: Towards open-world text-to-motion generation," *arXiv preprint arXiv:2312.14828*, 2023.

[25] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "Ntu rgb+ d: A large

scale dataset for 3d human activity analysis," in *CVPR*, 2016, pp. 1010–1019.

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

[27] Y. Ji, F. Xu, Y. Yang, F. Shen, H. T. Shen, and W.-S. Zheng, "A large-scale rgb-d database for arbitrary-view human action recognition," in *ACMMM*, 2018, pp. 1510–1518.

[28] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black, "Amass: Archive of motion capture as surface shapes," in *ICCV*, 2019, pp. 5442–5451.

[29] A. R. Punnakkal, A. Chandrasekaran, N. Athanasiou, A. Quiros-Ramirez, and M. J. Black, "Babel: Bodies, action and behavior with english labels," in *CVPR*, 2021, pp. 722–731.

[30] Y. Tang, J. Liu, A. Liu, B. Yang, W. Dai, Y. Rao, J. Lu, J. Zhou, and X. Li, "Flag3d: A 3d fitness activity dataset with language instruction," in *CVPR*, 2023, pp. 22 106–22 117.

[31] L. Xu, X. Lv, Y. Yan, X. Jin, S. Wu, C. Xu, Y. Liu, Y. Zhou, F. Rao, X. Sheng *et al.*, "Inter-x: Towards versatile human-human interaction analysis," in *CVPR*, 2024, pp. 22 260–22 271.

[32] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *NeurIPS*, pp. 6840–6851, 2020.

[33] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *ICML*, 2015, pp. 2256–2265.

[34] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *ICML*, 2021, pp. 8162–8171.

[35] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *CVPR*, 2022, pp. 10 684–10 695.

[36] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *ICLR*, 2021.

[37] K. Karunratanakul, K. Preechakul, S. Suwajanakorn, and S. Tang, "Guided motion diffusion for controllable human motion synthesis," in *CVPR*, 2023, pp. 2151–2162.

[38] Y. Xie, V. Jampani, L. Zhong, D. Sun, and H. Jiang, "Omnicontrol: Control any joint at any time for human motion generation," in *ICLR*, 2024.

[39] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever, "Consistency models," in *ICML*, 2023.

[40] S. Luo, Y. Tan, L. Huang, J. Li, and H. Zhao, "Latent consistency models: Synthesizing high-resolution images with few-step inference," *arXiv preprint arXiv:2310.04378*, 2023.

[41] L. Zhang, A. Rao, and M. Agrawala, "Adding conditional control to text-to-image diffusion models," in *ICCV*, 2023, pp. 3836–3847.

[42] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[43] K. Karunratanakul, K. Preechakul, E. Aksan, T. Beeler, S. Suwajanakorn, and S. Tang, "Optimizing diffusion noise can serve as universal motion priors," in *CVPR*, 2024, pp. 1334–1345.

[44] O. V. Aaron Van Den Oord, "Neural discrete representation learning," *NeurIPS*, vol. 30, 2017.

[45] S. Yan, Z. Li, Y. Xiong, H. Yan, and D. Lin, "Convolutional sequence generation for skeleton-based action synthesis," in *ICCV*, 2019, pp. 4394–4402.

[46] R. Zhao, H. Su, and Q. Ji, "Bayesian adversarial human motion synthesis," in *CVPR*, 2020, pp. 6225–6234.

[47] S. Raab, I. Leibovitch, P. Li, K. Aberman, O. Sorkine-Hornung, and D. Cohen-Or, "Modi: Unconditional motion synthesis from diverse data," in *CVPR*, 2023, pp. 13 873–13 883.

[48] P. Cervantes, Y. Sekikawa, I. Sato, and K. Shinoda, "Implicit neural representations for variable length human motion generation," in *ECCV*, 2022, pp. 356–372.

[49] L. Xu, Z. Song, D. Wang, J. Su, Z. Fang, C. Ding, W. Gan, Y. Yan, X. Jin, X. Yang *et al.*, "Actformer: A gan-based transformer towards general action-conditioned 3d human motion generation," in *ICCV*, 2023, pp. 2228–2238.

[50] T. Lee, G. Moon, and K. M. Lee, "Multiact: Long-term 3d human motion generation from multiple action labels," in *AAAI*, 2023, pp. 1231–1239.

[51] Z. Dou, X. Chen, Q. Fan, T. Komura, and W. Wang, "C· ase: Learning conditional adversarial skill embeddings for physics-based characters," in *SIGGRAPH Asia*, 2023, pp. 1–11.

[52] A. S. Lin, L. Wu, R. Corona, K. Tai, Q. Huang, and R. J. Mooney, "Generating animated videos of human activities from natural language descriptions," *Learning*, vol. 1, no. 2018, p. 1, 2018.

[53] C. Ahuja and L.-P. Morency, "Language2pose: Natural language grounded pose forecasting," in *3DV*, 2019, pp. 719–728.

[54] U. Bhattacharya, N. Rewkowski, A. Banerjee, P. Guhan, A. Bera, and D. Manocha, "Text2gestures: A transformer-based network for generating emotive body gestures for virtual agents," in *VR*, 2021, pp. 1–10.

[55] A. Ghosh, N. Cheema, C. Oguz, C. Theobalt, and P. Slusallek, "Synthesis of compositional animations from textual descriptions," in *ICCV*, 2021, pp. 1396–1406.

[56] G. Tevet, B. Gordon, A. Hertz, A. H. Bermano, and D. Cohen-Or, "Motionclip: Exposing human motion generation to clip space," in *ECCV*, 2022, pp. 358–374.

[57] M. Petrovich, M. J. Black, and G. Varol, "Tmr: Text-to-motion retrieval using contrastive 3d human motion synthesis," in *ICCV*, 2023, pp. 9488–9497.

[58] B. Jiang, X. Chen, W. Liu, J. Yu, G. Yu, and T. Chen, "Motiongpt: Human motion as a foreign language," *NeurIPS*, 2024.

[59] W. Wan, Z. Dou, T. Komura, W. Wang, D. Jayaraman, and L. Liu, "Tlcontrol: Trajectory and language control for human motion synthesis," *arXiv preprint arXiv:2311.17135*, 2023.

[60] W. Zhou, Z. Dou, Z. Cao, Z. Liao, J. Wang, W. Wang, Y. Liu, T. Komura, W. Wang, and L. Liu, "Emdm: Efficient motion diffusion model for fast, high-quality motion generation," *arXiv preprint arXiv:2312.02256*, 2023.

[61] M. Petrovich, O. Litany, U. Iqbal, M. J. Black, G. Varol, X. Bin Peng, and D. Rempe, "Multi-track timeline control for text-driven 3d human motion generation," in *CVPRW*, 2024, pp. 1911–1921.

[62] G. Barquero, S. Escalera, and C. Palmero, "Seamless human motion composition with blended positional encodings," in *CVPR*, 2024, pp. 457–469.

[63] Z. Wang, Y. Chen, B. Jia, P. Li, J. Zhang, J. Zhang, T. Liu, Y. Zhu, W. Liang, and S. Huang, "Move as you say interact as you can: Language-guided human motion generation with scene affordance," in *CVPR*, 2024, pp. 433–444.

[64] K. Fan, J. Tang, W. Cao, R. Yi, M. Li, J. Gong, J. Zhang, Y. Wang, C. Wang, and L. Ma, "Freemotion: A unified framework for number-free text-to-motion synthesis," *arXiv preprint arXiv:2405.15763*, 2024.

[65] L. Zhong, Y. Xie, V. Jampani, D. Sun, and H. Jiang, "Smoodi: Stylized motion diffusion model," *arXiv preprint arXiv:2407.12783*, 2024.

[66] P. Cong, Z. W. Dou, Y. Ren, W. Yin, K. Cheng, Y. Sun, X. Long, X. Zhu, and Y. Ma, "Laserhuman: Language-guided scene-aware human motion generation in free environment," *arXiv preprint arXiv:2403.13307*, 2024.

[67] L.-H. Chen, S. Lu, A. Zeng, H. Zhang, B. Wang, R. Zhang, and L. Zhang, "Motionllm: Understanding human behaviors from human motions and videos," *arXiv preprint arXiv:2405.20340*, 2024.

[68] Z. Meng, Y. Xie, X. Peng, Z. Han, and H. Jiang, "Rethinking diffusion for text-driven human motion generation," *arXiv preprint arXiv:2411.16575*, 2024.

[69] 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," *arXiv preprint arXiv:2412.14559*, 2024.

[70] J. Li, Y. Yin, H. Chu, Y. Zhou, T. Wang, S. Fidler, and H. Li, "Learning to generate diverse dance motions with transformer," *arXiv preprint arXiv:2008.08171*, 2020.

[71] W. Zhuang, C. Wang, S. Xia, J. Chai, and Y. Wang, "Music2dance: Music-driven dance generation using wavenet," *arXiv preprint arXiv:2002.03761*, 2020.

[72] R. Huang, H. Hu, W. Wu, K. Sawada, M. Zhang, and D. Jiang, "Dance revolution: Long-term dance generation with music via curriculum learning," *arXiv preprint arXiv:2006.06119*, 2020.

[73] R. Li, S. Yang, D. A. Ross, and A. Kanazawa, "Ai choreographer: Music conditioned 3d dance generation with aist++," in *ICCV*, 2021, pp. 13 401–13 412.

[74] B. Li, Y. Zhao, S. Zhelun, and L. Sheng, "Danceformer: Music conditioned 3d dance generation with parametric motion transformer," in *AAAI*, 2022, pp. 1272–1279.

[75] 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 *CVPR*, 2022, pp. 11 050–11 059.

[76] J. Tseng, R. Castellon, and K. Liu, "Edge: Editable dance generation from music," in *CVPR*, 2023, pp. 448–458.

[77] R. Li, J. Zhao, Y. Zhang, M. Su, Z. Ren, H. Zhang, Y. Tang, and X. Li, "Finedance: A fine-grained choreography dataset for 3d full body dance generation," in *ICCV*, 2023, pp. 10 234–10 243.

[78] R. Li, Y. Zhang, Y. Zhang, H. Zhang, J. Guo, Y. Zhang, Y. Liu, and X. Li, "Lodge: A coarse to fine diffusion network for long

dance generation guided by the characteristic dance primitives," in *CVPR*, 2024, pp. 1524–1534.

[79] Z. Wang, J. Wang, D. Lin, and B. Dai, "Intercontrol: Generate human motion interactions by controlling every joint," *arXiv preprint arXiv:2311.15864*, 2023.

[80] E. Pinyoanuntapong, M. U. Saleem, K. Karunratanakul, P. Wang, H. Xue, C. Chen, C. Guo, J. Cao, J. Ren, and S. Tulyakov, "Controlmm: Controllable masked motion generation," *arXiv preprint arXiv:2410.10780*, 2024.

[81] D. Holden, J. Saito, and T. Komura, "A deep learning framework for character motion synthesis and editing," *TOG*, vol. 35, no. 4, pp. 1–11, 2016.

[82] D. Holden, T. Komura, and J. Saito, "Phase-functioned neural networks for character control," *TOG*, vol. 36, no. 4, pp. 1–13, 2017.

[83] Y. Shi, J. Wang, X. Jiang, and B. Dai, "Controllable motion diffusion model," *arXiv preprint arXiv:2306.00416*, 2023.

[84] T. Li, C. Qiao, G. Ren, K. Yin, and S. Ha, "Aamdm: Accelerated auto-regressive motion diffusion model," in *CVPR*, 2024, pp. 1813–1823.

[85] T. Karras, M. Aittala, T. Aila, and S. Laine, "Elucidating the design space of diffusion-based generative models," *NeurIPS*, pp. 26 565–26 577, 2022.

[86] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu, "Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps," *NeurIPS*, pp. 5775–5787, 2022.

[87] ——, "Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models," *arXiv preprint arXiv:2211.01095*, 2022.

[88] J. Ho and T. Salimans, "Classifier-free diffusion guidance," *arXiv preprint arXiv:2207.12598*, 2022.

[89] P. J. Huber, "Robust estimation of a location parameter," *The Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 73–101, 1964.

[90] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *NeurIPS*, 2017.

[91] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[92] D. P. Kingma, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[93] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *NeurIPS*, 2019.

[94] 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," in *AAAI*, 2024, pp. 6252–6260.

[95] K. Onuma, C. Faloutsos, and J. K. Hodgins, "Fmdistance: A fast and effective distance function for motion capture data." *Eurographics*, vol. 7, 2008.

[96] M. C. Meinard Müller, Tido Röder, "Efficient content-based retrieval of motion capture data," *TOG*, vol. 24, no. 3, pp. 677–685, 2005.

Wenxun Dai is currently a Master's student in Tsinghua Shenzhen International Graduate School, Tsinghua University. Before that, he received his B.Eng. degree from Xidian University in 2023. His current research interest is character animation. He has published papers in top-tier conferences, such as ECCV and CVPR.

Ling-Hao Chen is currently a Ph.D. candidate at Tsinghua University, supervised by Prof. Heung-Yeung Shum. His current research interest lies in character animation, digital generation, and embodied intelligence. He obtained his B.E. degree from Xidian University. He has published papers in top-tier conferences and journals, such as ICCV, ECCV, ICML, ICLR, *etc.*. He was rated as the Top Reviewer of NeurIPS 2023. He is regularly the reviewer of CVPR, ICML, NeurIPS, ICLR, AAAI, *etc.*.

Yufei Huo is currently a Master's student in Tsinghua Shenzhen International Graduate School, Tsinghua University. Before that, he received his bachelor's degree from Dalian University of Technology in 2024. His current research interest focuses on music-driven 3D motion generation.

Jingbo Wang is a research scientist at the Shanghai Artificial Intelligence Laboratory. He earned his Ph.D. from the Chinese University of Hong Kong in 2023. He has published over 30 papers in top-tier journals and conferences, focusing on human motion generation and simulation, as well as image segmentation. Currently, his research interests are centered on versatile skill learning for embodied agents.

Jinpeng Liu is currently a Master's student in Tsinghua Shenzhen International Graduate School, Tsinghua University. Before that, he received his bachelor's degree in the School of Intelligent System Engineering from Sun Yat-sen University in 2022. His current research interest is computer vision, including motion and 3D generation.

Bo Dai is an assistant professor in the Musketeers Foundation Institute of Data Science, The University of Hong Kong. He obtained his PhD degree from The Chinese University of Hong Kong. He was a research scientist with the Shanghai Artificial Intelligence Laboratory, and was a research assistant professor with S-Lab for advanced intelligence, Nanyang Technological University. He has authored or co-authored more than 70 papers in top-tier conferences and journals, with over 8900 google scholar citations. His research interests include Generative AI and its interdisciplinary applications in areas covering Embodied AI, Scientific Discovery, Metaverse and Creativity. He is an area chair of NeurIPS2024 and AAAI 2021.

Yansong Tang (Member, IEEE) received the BS and PhD degrees both from the Department of Automation, Tsinghua University, in 2015 and 2020, respectively. From 2020 to 2022, he served as a postdoctoral fellow with the Department of Engineering Science of the University of Oxford. He is currently a tenure-track Assistant Professor of Shenzhen International Graduate School, Tsinghua University. His research interests include computer vision, pattern recognition, and video processing. In recent years, he has authored more than 40 papers in top peer-reviewed journals and conferences such as the IEEE Transactions on Pattern Analysis and Machine Intelligence, the IEEE Transactions on Image Processing, and CVPR. He is a member of the IEEE and an area chair of CVPR25.

Real-time Controllable Motion Generation via Latent Consistency Model

Supplementary Material

Appendix A
Additional Experiments

A.1 Eliminating structural defects in the original denoiser architecture

In the denoising transformer, the original MLD architecture leverages stacked transformer encoder layers with a skip-connection structure to improve modeling capability. Its self-attention module incorporates three distinct token types: **(1) latent tokens derived from the VAE encoder, (2) sentence-level text feature, and (3) diffusion time step embedding**. Within the network, we identified two structural defects: (i) Unlike other tokens, the VAE latent tokens are **directly** fed into the self-attention module without passing through a learnable linear layer. The specific reason here is that the VAE latent tokens have a feature dimension of 256, which is consistent with the hidden dimension of the self-attention module. Therefore, **dimensional adjustment is omitted**. However, this bypass means that the VAE latent tokens are not modulated to better handle multimodal signals in the self-attention, potentially making their integration into the model less effective. (ii) The text feature passes through a ReLU activation function **first** before the learnable linear layer. This ReLU function suppresses negative values, leading to the loss of valuable textual information encoded in these negative components. To rectify these structural flaws, we define two types of operations.

- ***Op1***: introduces a trainable linear layer after the VAE latent tokens to enhance multimodal signal modulation.
- ***Op2***: removes the unnecessary ReLU activation function to preserve negative components in the text feature.

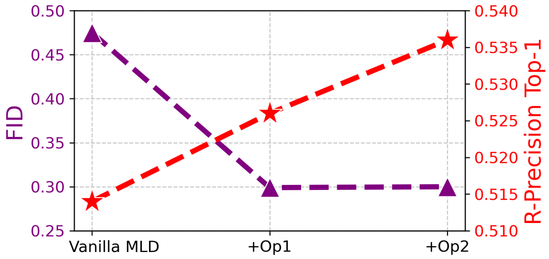


Fig. 11: Impact of *Op1* and *Op2* on MLD performance.

As shown in Figure 11, ***Op1*** significantly improves both motion generation quality (FID) and motion-text alignment capability (R-Precision Top-1), while ***Op2*** demonstrates that preserving the negative information filtered out by the ReLU activation function is essential for enhancing text alignment (**Here, we observe that using activation functions like SiLU, which preserve negative values, achieves the same effect.**). We obtain the results using the VAE checkpoint provided by the authors of MLD and train MLDs using our custom training settings. These two simple yet effective operations can have a significant impact on the generation performance of MLD and are adopted in our subsequent exploratory experiments.

A.2 Impact of the hyperparameters of training MotionLCM

We conduct a comprehensive analysis of the training hyperparameters of MotionLCM, including the training guidance scale range $[w_{\min}, w_{\max}]$, EMA rate μ, skipping interval k, and the type of loss. We summarize the evaluation results based on one-step inference in Table 6. We find out that using a dynamic training guidance scale (*e.g.*, $w \in [5, 15]$) during training leads to an improvement in model performance compared to using a static training guidance scale (*e.g.*, $w = 7.5$). Additionally, an excessively large range for the training guidance scale can also negatively impact the performance of the model (*e.g.*, $w \in [2, 18]$). Regarding the EMA rate μ, we observe that the larger the value of μ, the better the performance of the model. This indicates that maintaining a slower update rate for the target network $\mathbf{\Theta}^-$ helps improve the performance of latent consistency distillation. When the skipping interval k continues to increase, the performance of the distillation model improves progressively, but larger values of k (*e.g.*, $k = 50$) may result in inferior results. As for the type of loss, the Huber loss [89] significantly outperforms the L2 loss, demonstrating its superior robustness.

A.3 Impact of control loss weights λ

To verify the impact of different control loss weights λ on the control performance of MotionLCM, we report the experimental results in Table 7. We also include experiments of MotionLCM without ControlNet (*i.e.*, only text-to-motion) for comparison. We found a significant improvement in control-related metrics (*e.g.*, Loc. err.) after introducing motion

TABLE 6: Ablation study on different training guidance scale ranges $[w_{\min}, w_{\max}]$, EMA rates μ, skipping intervals k and types of loss. We use metrics in Table 2 and adopt a one-step inference setting with the CFG scale of 7.5 for fair comparison.

Methods	R-Precision ↑ Top 1	FID ↓	MM Dist ↓	Diversity →	MModality ↑
Real	$0.511^{\pm.003}$	$0.002^{\pm.000}$	$2.974^{\pm.008}$	$9.503^{\pm.065}$	-
MotionLCM ($w \in [5, 15]$)	**0.502**$^{\pm.003}$	$0.467^{\pm.012}$	$3.022^{\pm.009}$	$9.631^{\pm.066}$	$2.172^{\pm.082}$
MotionLCM ($w \in [2, 18]$)	$0.497^{\pm.003}$	$0.481^{\pm.009}$	$3.030^{\pm.010}$	$9.644^{\pm.073}$	$2.226^{\pm.091}$
MotionLCM ($w = 7.5$)	$0.486^{\pm.002}$	$0.479^{\pm.009}$	$3.094^{\pm.009}$	$9.610^{\pm.072}$	$2.320^{\pm.097}$
MotionLCM ($\mu = 0.95$)	**0.502**$^{\pm.003}$	$0.467^{\pm.012}$	$3.022^{\pm.009}$	$9.631^{\pm.066}$	$2.172^{\pm.082}$
MotionLCM ($\mu = 0.50$)	$0.498^{\pm.003}$	$0.478^{\pm.009}$	$3.022^{\pm.010}$	$9.655^{\pm.071}$	$2.188^{\pm.087}$
MotionLCM ($\mu = 0$)	$0.499^{\pm.003}$	$0.505^{\pm.008}$	$3.018^{\pm.009}$	$9.706^{\pm.070}$	$2.123^{\pm.085}$
MotionLCM ($k = 50$)	$0.488^{\pm.003}$	$0.547^{\pm.011}$	$3.096^{\pm.010}$	**9.511**$^{\pm.074}$	**2.324**$^{\pm.091}$
MotionLCM ($k = 20$)	**0.502**$^{\pm.003}$	$0.467^{\pm.012}$	$3.022^{\pm.009}$	$9.631^{\pm.066}$	$2.172^{\pm.082}$
MotionLCM ($k = 10$)	$0.497^{\pm.003}$	$0.449^{\pm.009}$	**3.017**$^{\pm.010}$	$9.693^{\pm.075}$	$2.133^{\pm.086}$
MotionLCM ($k = 5$)	$0.488^{\pm.003}$	**0.438**$^{\pm.009}$	$3.044^{\pm.009}$	$9.647^{\pm.074}$	$2.147^{\pm.083}$
MotionLCM ($k = 1$)	$0.442^{\pm.002}$	$0.635^{\pm.011}$	$3.255^{\pm.008}$	$9.384^{\pm.080}$	$2.146^{\pm.075}$
MotionLCM (w/ Huber)	**0.502**$^{\pm.003}$	$0.467^{\pm.012}$	$3.022^{\pm.009}$	$9.631^{\pm.066}$	$2.172^{\pm.082}$
MotionLCM (w/ L2)	$0.486^{\pm.002}$	$0.622^{\pm.010}$	$3.114^{\pm.009}$	$9.573^{\pm.069}$	$2.218^{\pm.086}$

ControlNet (*i.e.*, $\lambda = 0$). Furthermore, control performance can be further improved by introducing control loss (*i.e.*, $\lambda > 0$). Increasing the weight λ enhances control performance but leads to a decline in the generation quality, which is reflected in higher FID scores. To balance these two aspects, we adopt $\lambda = 1$ as our default setting for training motion ControlNet.

TABLE 7: Ablation study on different control loss weights λ. We present the results of (1, 2, 4)-step inference. We add the MotionLCM without ControlNet for comparison.

Methods	FID ↓	R-Precision ↑ Top 3	Diversity →	Traj. err. ↓ (50cm)	Loc. err. ↓ (50cm)	Avg. err. ↓
Real	0.002	0.797	9.503	0.0000	0.0000	0.0000
MotionLCM (1-step, w/o control)	0.467	0.803	9.631	0.7605	0.2302	0.3493
MotionLCM (2-step, w/o control)	0.368	**0.805**	9.640	0.7646	0.2214	0.3386
MotionLCM (4-step, w/o control)	**0.304**	0.798	9.607	0.7739	0.2207	0.3359
MotionLCM (1-step, $\lambda = 0$)	0.319	0.752	9.424	0.2986	0.0344	0.1410
MotionLCM (2-step, $\lambda = 0$)	0.315	0.770	9.427	0.2840	0.0328	0.1365
MotionLCM (4-step, $\lambda = 0$)	0.328	0.745	9.441	0.2973	0.0339	0.1398
MotionLCM (1-step, $\lambda = 0.1$)	0.344	0.753	9.386	0.2711	0.0275	0.1310
MotionLCM (2-step, $\lambda = 0.1$)	0.324	0.759	9.428	0.2631	0.0256	0.1268
MotionLCM (4-step, $\lambda = 0.1$)	0.357	0.743	9.415	0.2713	0.0268	0.1309
MotionLCM (1-step, $\lambda = 1.0$)	0.419	0.756	9.390	0.1988	0.0147	0.1127
MotionLCM (2-step, $\lambda = 1.0$)	0.397	0.759	9.469	0.1960	0.0143	0.1092
MotionLCM (4-step, $\lambda = 1.0$)	0.444	0.753	9.355	0.2089	0.0172	0.1140
MotionLCM (1-step, $\lambda = 10.0$)	0.636	0.744	9.479	**0.1465**	**0.0097**	**0.0967**
MotionLCM (2-step, $\lambda = 10.0$)	0.551	0.757	9.569	0.1590	0.0107	0.0987
MotionLCM (4-step, $\lambda = 10.0$)	0.568	0.742	**9.486**	0.1723	0.0132	0.1045

A.4 Impact of different control ratios τ and number of control joints K

In Table 8, we present the results of all models with the testing control ratio as 0.25 and keep the number of control joints K equal in both training and testing. We found that the model with the fixed training control ratio (*i.e.*, $\tau = 0.25$) performs better compared to a dynamic ratio (*e.g.*, $\tau \in [0.1, 0.5]$), and we discover that our model maintains good performance when incorporating additional redundant control signals, such as whole-body joints with $K = 22$.

A.5 Comparison to other ODE Solvers

To validate the effectiveness of latent consistency distillation, we compare three ODE solvers (DDIM [36], DPM [86], DPM++ [87]). The quantitative results shown in Table 9 demonstrate that our MotionLCM notably outperforms baseline methods. Moreover, unlike DDIM [36], DPM [86], and DPM++ [87], requiring more peak memory per sampling step when using CFG [88], MotionLCM only requires one forward pass, saving both time and memory cost.

TABLE 8: Ablation study on different control ratios τ and number of control joints K. We report the results of (1, 2, 4)-step inference. "$*$" is the default training setting.

Methods	FID ↓	R-Precision ↑	Diversity →	Traj. err. ↓	Loc. err. ↓	Avg. err. ↓
		Top 3		(50cm)	(50cm)	
Real	0.002	0.797	9.503	0.0000	0.0000	0.0000
MotionLCM* (1-step, $\tau = 0.25$, $K = 6$)	0.419	0.756	9.390	0.1988	0.0147	0.1127
MotionLCM* (2-step, $\tau = 0.25$, $K = 6$)	**0.397**	0.759	9.469	**0.1960**	0.0143	0.1092
MotionLCM* (4-step, $\tau = 0.25$, $K = 6$)	0.444	0.753	9.355	0.2089	0.0172	0.1140
MotionLCM (1-step, $\tau \in [0.1, 0.25]$)	0.456	0.757	9.477	0.2821	0.0234	0.1214
MotionLCM (2-step, $\tau \in [0.1, 0.25]$)	0.409	**0.769**	9.592	0.2707	0.0230	0.1179
MotionLCM (4-step, $\tau \in [0.1, 0.25]$)	0.457	0.757	9.540	0.2928	0.0256	0.1228
MotionLCM (1-step, $\tau \in [0.1, 0.5]$)	0.448	0.763	9.538	0.2390	0.0182	0.1182
MotionLCM (2-step, $\tau \in [0.1, 0.5]$)	0.413	0.768	9.517	0.2349	0.0180	0.1153
MotionLCM (4-step, $\tau \in [0.1, 0.5]$)	0.446	0.753	**9.498**	0.2517	0.0199	0.1196
MotionLCM (1-step, $K = 12$)	0.412	0.753	9.412	0.2072	0.0110	0.1029
MotionLCM (2-step, $K = 12$)	0.410	0.758	9.509	0.1979	0.0108	0.1000
MotionLCM (4-step, $K = 12$)	0.442	0.755	9.380	0.2169	0.0132	0.1048
MotionLCM (1-step, $K = 22$(whole-body))	0.436	0.748	9.379	0.2143	0.0083	0.0914
MotionLCM (2-step, $K = 22$(whole-body))	0.413	0.758	9.492	0.2061	**0.0082**	**0.0881**
MotionLCM (4-step, $K = 22$(whole-body))	0.461	0.745	9.459	0.2173	0.0097	0.0918

TABLE 9: Quantitative results with the testing CFG scale $w = 7.5$. MotionLCM notably outperforms baseline methods [36], [86], [87] on HumanML3D [1] dataset, demonstrating the effectiveness of latent consistency distillation.

Methods	R-Precision (Top 3) ↑			FID ↓		
	1-Step	2-Step	4-Step	1-Step	2-Step	4-Step
DDIM [36]	$0.651^{\pm.003}$	$0.691^{\pm.002}$	$0.765^{\pm.003}$	$4.022^{\pm.043}$	$2.802^{\pm.038}$	$0.966^{\pm.018}$
DPM [86]	$0.651^{\pm.003}$	$0.691^{\pm.002}$	$0.777^{\pm.003}$	$4.022^{\pm.043}$	$2.798^{\pm.038}$	$0.727^{\pm.015}$
DPM++ [87]	$0.651^{\pm.003}$	$0.691^{\pm.002}$	$0.777^{\pm.003}$	$4.022^{\pm.043}$	$2.798^{\pm.038}$	$0.684^{\pm.015}$
MotionLCM	**0.803**$^{\pm.002}$	**0.805**$^{\pm.002}$	**0.798**$^{\pm.002}$	**0.467**$^{\pm.012}$	**0.368**$^{\pm.011}$	**0.304**$^{\pm.012}$

A.6 Impact of different testing CFGs

As shown in Figure 12, we provide an extensive ablation study on the testing CFG [88]. It can be observed that, under different testing CFGs, increasing the number of inference steps continuously improves the performance. However, further increasing the inference steps results in comparable performance but significantly increases the time cost.

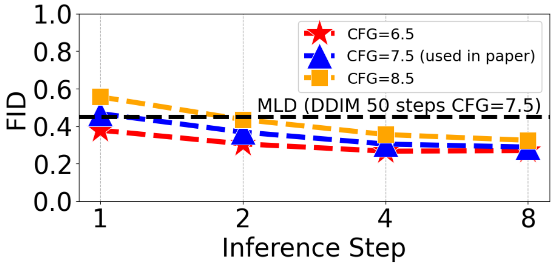


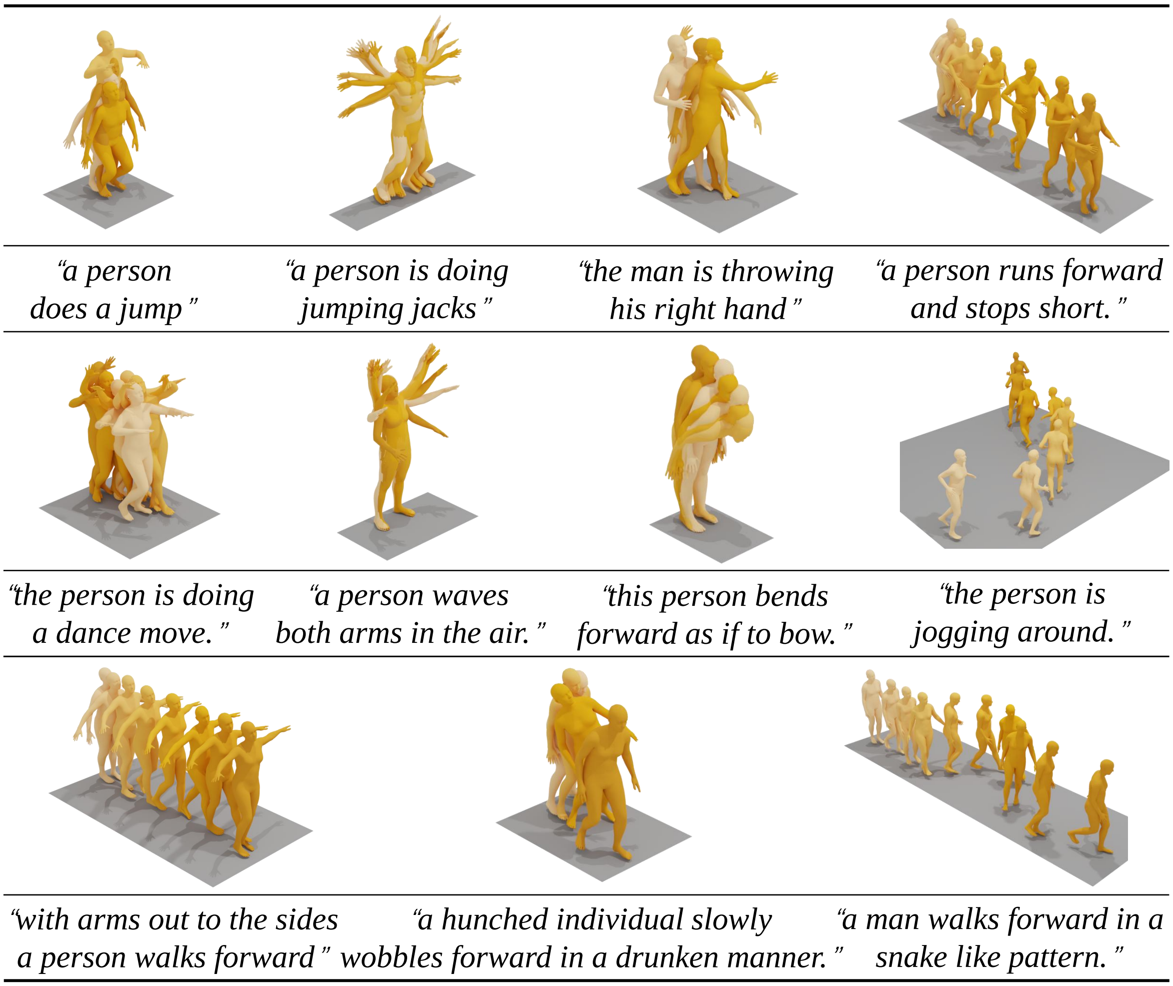
Fig. 12: Comparison of testing CFGs

Appendix B
More Qualitative Results

In this section, we provide more qualitative results of our MotionLCM. Figure 13 presents more generation results on the text-to-motion task. Figure 14 displays additional visualization results on the motion control task. All videos shown in the figures can be found on our website (https://dai-wenxun.github.io/MotionLCM-page).

Appendix C
Metric Definitions

Time cost: To assess the inference efficiency of models, we follow [6] to report the Average Inference Time per Sentence (AITS) measured in seconds. We calculate AITS on the test set of HumanML3D [1] by setting the batch size to 1 and excluding the time cost for model and dataset loading parts.



“a person does a jump”

“a person is doing jumping jacks”

“the man is throwing his right hand”

“a person runs forward and stops short.”

“the person is doing a dance move.”

“a person waves both arms in the air.”

“this person bends forward as if to bow.”

“the person is jogging around.”

“with arms out to the sides a person walks forward”

“a hunched individual slowly wobbles forward in a drunken manner.”

“a man walks forward in a snake like pattern.”

Fig. 13: More qualitative results of MotionLCM on the text-to-motion task.

Motion quality: Frechet Inception Distance (FID) measures the distributional difference between the generated and real motions, calculated using the feature extractor associated with a specific dataset, *e.g.*, HumanML3D [1].

Motion diversity: Following [3], [18], we report Diversity and MultiModality to evaluate the generated motion diversity. Diversity measures the variance of the generated motions across the whole set. Specifically, two subsets of the same size S_d are randomly sampled from all generated motions with their extracted motion feature vectors $\{\mathbf{v}_1, ..., \mathbf{v}_{S_d}\}$ and $\{\mathbf{v}_1^{'}, ..., \mathbf{v}_{S_d}^{'}\}$. Diversity is defined as follows,

$$\text{Diversity} = \frac{1}{S_d} \sum_{i=1}^{S_d} ||\mathbf{v}_i - \mathbf{v}_i^{'}||_2. \tag{17}$$

Different from Diversity, MultiModality (MModality) measures how much the generated motions diversify within each textual description. Specifically, a set of textual 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 the c-th textual description, with extracted feature vectors $\{\mathbf{v}_{c,1}, ..., \mathbf{v}_{c,I}\}$ and $\{\mathbf{v}_{c,1}^{'}, ..., \mathbf{v}_{c,I}^{'}\}$. MModality is formalized as follows,

$$\text{MModality} = \frac{1}{C \times I} \sum_{c=1}^{C} \sum_{i=1}^{I} ||\mathbf{v}_{c,i} - \mathbf{v}_{c,i}^{'}||_2. \tag{18}$$

Condition matching: [1] provides motion/text feature extractors to generate geometrically closed features for matched text-motion pairs and vice versa. Under this feature space, evaluating motion-retrieval precision (R-Precision) involves mixing the generated motion with 31 mismatched motions and then calculating the text-motion Top-1/2/3 matching accuracy. Multimodal Distance (MM Dist) calculates the mean distance between the generated motions and texts.

Control error: Following [38], we report Trajectory error, Location error, and Average error to assess the motion control performance. Trajectory error (Traj. err.) is defined as the proportion of unsuccessful trajectories, *i.e.*, if a control joint in the generated motion exceeds a certain distance threshold from the corresponding joint in the given control trajectory, it is considered a failed trajectory. Similar to the Trajectory error, Location error (Loc. err.) is defined as the ratio of unsuccessful joints. In our experiments, we adopt 50cm as the distance threshold to calculate the Trajectory error and Location error.

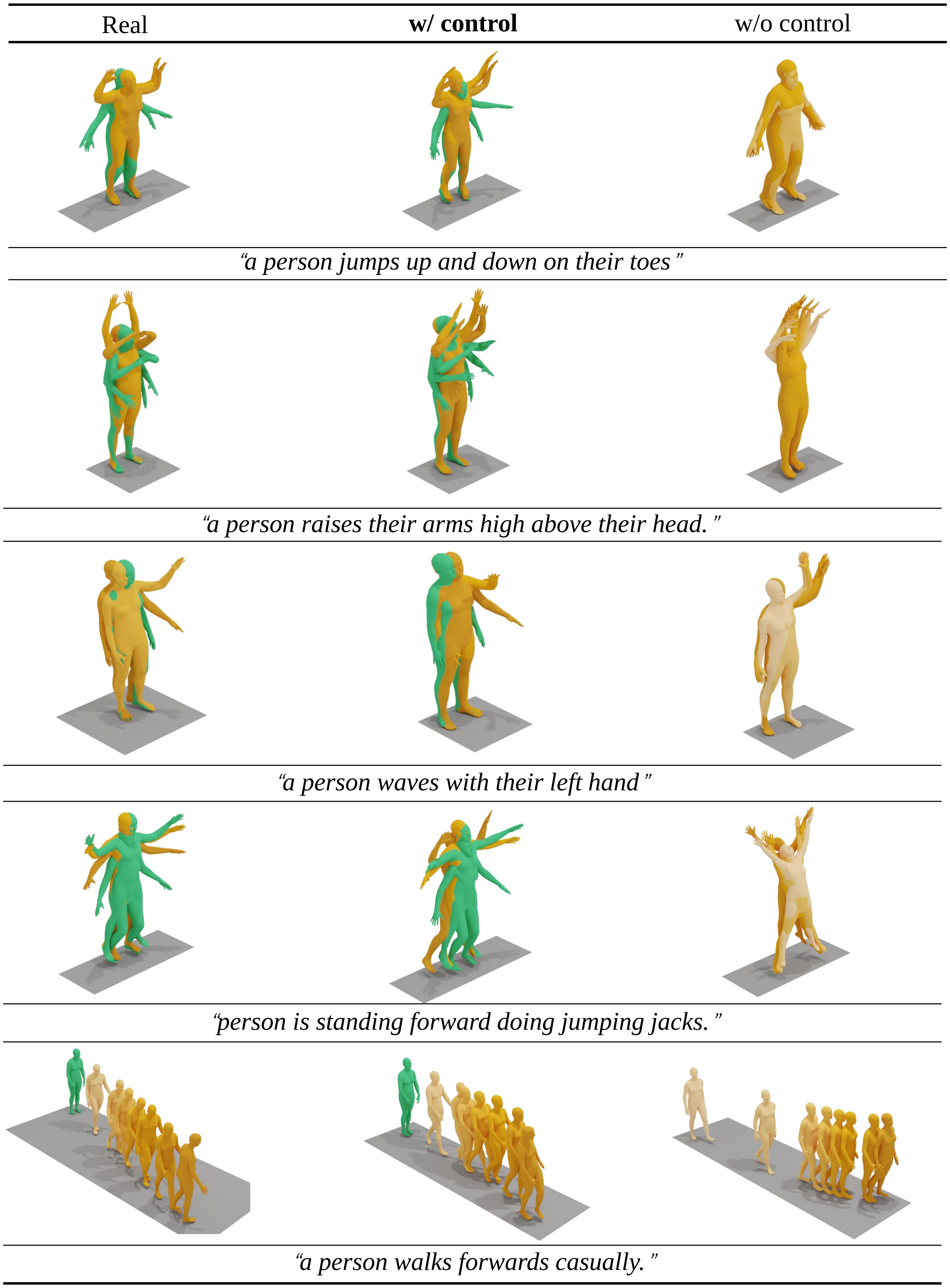

Real
w/ control
w/o control
"a person jumps up and down on their toes"
"a person raises their arms high above their head."
"a person waves with their left hand"
"person is standing forward doing jumping jacks."
"a person walks forwards casually."

Fig. 14: More qualitative results of MotionLCM on the motion control task.

Average error (Avg. err.) denotes the mean distance between the control joint positions in the generated motion and those in the given control trajectory.

Reconstruction error: Mean Per Joint Position Error (MPJPE) quantifies the average Euclidean distance between the reconstructed joint positions and the corresponding ground truth joint positions, following an alignment of the root joint (typically the pelvis) to eliminate global positional discrepancies. This metric provides an indication of the accuracy of joint localization in 3D space. Feature error (Fea. err.) measures the L2 norm between the reconstructed motion features and the real motion features, providing a feature-level assessment of the reconstruction quality.