MotionLCM: Real-time Controllable Motion Generation via Latent Consistency Model

Wenxun Dai[1,2], Ling-Hao Chen[2⋆], Jingbo Wang[3✉], Jinpeng Liu[1,2]
Bo Dai[3✉], Yansong Tang[1,2]

[1]Shenzhen Key Laboratory of Ubiquitous Data Enabling, Tsinghua Shenzhen International Graduate School [2]Tsinghua University [3]Shanghai AI Laboratory
{wxdai2001, thu.lhchen, wangjingbo1219, liu.jinpeng.55}@gmail.com
{doubledaibo, tangyansong15}@gmail.com
Project page: https://dai-wenxun.github.io/MotionLCM-page

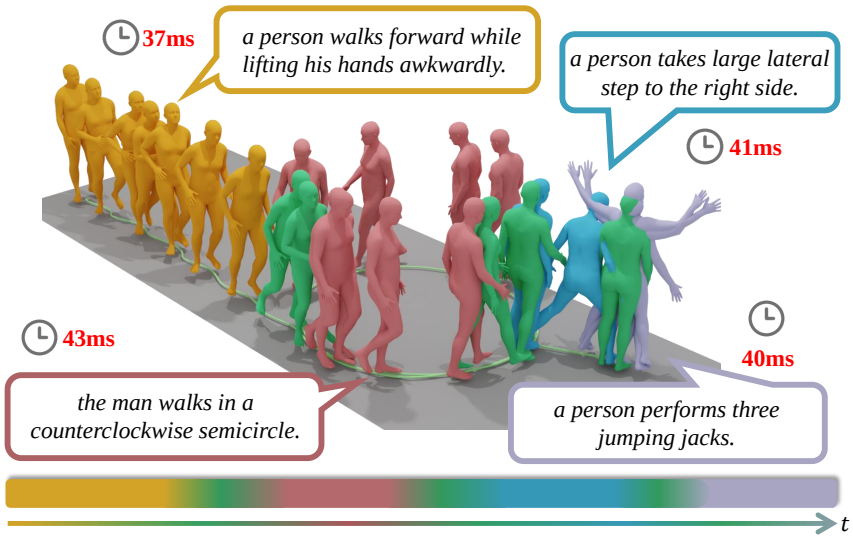


Fig. 1: We propose MotionLCM, a real-time controllable motion latent consistency model. Our model uses the last few frames of the previous motion as temporal control signals to autoregressively generate the next motion in real-time under different text prompts. Green blocks denote the junctions. The numbers in red are the inference time.

Abstract. This work introduces MotionLCM, extending controllable motion generation to a real-time level. 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,

⋆ Project lead. ✉ Corresponding author.

building upon the latent diffusion model [9]. 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 (*e.g.*, initial poses) in the vanilla motion space to control the generation process directly, similar to controlling other latent-free diffusion models [29, 73] for motion generation. By employing these techniques, our approach can generate human motions with text and control signals in real-time. Experimental results demonstrate the remarkable generation and controlling capabilities of MotionLCM while maintaining real-time runtime efficiency.

Keywords: Text-to-Motion · Real-time Control · Consistency Model

1 Introduction

Text-to-motion generation (T2M) has attracted increasing attention [1, 15, 43, 49, 65] due to its important roles in many applications [70, 72]. Previous attempts mainly focus on GANs [1, 38], VAEs [3, 19, 48, 49] and diffusion models [9, 11, 39, 56, 65, 78, 83] via pairwise text-motion data [17, 26, 45, 52, 53, 57, 63, 74] and achieve impressive generation results. Existing approaches [9, 65, 83] mainly take diffusion models [20, 46, 55, 60] 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 [61]. Specifically, MDM [65] and MLD [9] require ∼12s 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.

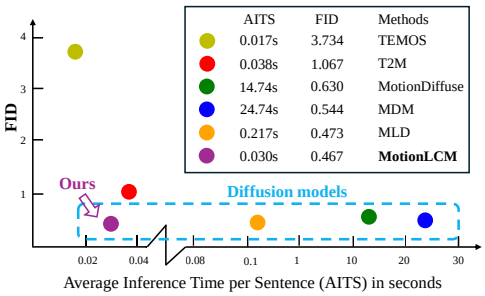


Fig. 2: Comparison of the inference time costs on HumanML3D [17]. We compare the AITS and FID metrics with five SOTA methods. The closer the model is to the origin the better. Diffusion-based models are indicated by the blue dashed box. Our MotionLCM achieves real-time inference speed while ensuring high-quality motion generation.

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 [29, 56, 73]. Although these attempts enjoy impressive controlling ability in the T2M task, there still exists a significant gap towards real-time applications. For example, OmniControl [73] 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.

Recently, the concept of consistency models [44, 62] 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 [9], 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 via latent consistency distillation [44].

Here, in MotionLCM, we are facing another challenge on how to control motions with spatial-temporal signals (*e.g.*, initial poses) in the latent space. Previous methods [56, 73] 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 [82] in controllable image generation [55], 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 [30] 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 [9].

With our key designs, our proposed MotionLCM successfully enjoys the balance between the generation quality and efficiency in controllable motion generation. Before delivering into detail, we sum up our core contributions as follows.

- We propose the Motion Latent Consistency Model (MotionLCM) via consistency distillation on the motion latent diffusion model extending controllable motion generation to a real-time level.
- Building upon our achievement of real-time motion generation, we introduce a motion ControlNet, enabling high-quality controllable motion generation.
- Extensive experimental results show that MotionLCM enjoys a good balance of generation quality, controlling capability, and real-time efficiency.

2 Related Work

2.1 Human Motion Generation

Generating human motions can be divided into three main fashions according to inputs: motion synthesis (1) without any condition [54, 65, 77, 84], (2) with some

given multi-modal conditions, such as action labels [6, 12, 19, 31, 48, 75], textual description [1–5, 7, 7, 9–11, 13–15, 17, 18, 24, 27, 29, 37–39, 43, 49–51, 56, 64, 65, 68–70, 72, 78, 81, 83, 85, 86], audio or music [32–35, 59, 66], (3) with user-defined trajectories [22, 23, 29, 36, 56, 58, 68, 71, 73]. To generate diverse, natural, and high-quality human motions, many generative models have been explored by [2, 38, 49, 79, 80]. Recently, diffusion-based models significantly improved the motion generation performance and diversity [8, 9, 11, 16, 65, 76, 83] with stable training. Specifically, MotionDiffuse [83] represents the first text-based motion diffusion model that provides fine-grained instructions on body parts and achieves arbitrary-length motion synthesis with time-varied text prompts. MDM [65] introduces a motion diffusion model that operates on raw motion data, enabling both high-quality generation and generic conditioning that together comprise a good baseline for new motion generation tasks. Based on MDM [65], OmniControl [73] integrates flexible spatial-temporal control signals across different joints by combining analytic spatial guidance and realism guidance into the diffusion model, ensuring that the generated motion closely conforms to the input control signals. The work most relevant to ours is MLD [9], which introduces a motion latent-based diffusion model to enhance generation quality and reduce computational resource requirements. The key idea is training a VAE [30] for motion embedding, followed by implementing latent diffusion [55] within the learned latent space. However, these diffusion fashions inevitably require considerable sampling steps for motion synthesis during inference, even with some sampling acceleration methods [61]. Thus, we propose MotionLCM, which not only guarantees high-quality controllable motion generation but also achieves real-time runtime efficiency.

3 Method

In this section, we first briefly introduce preliminaries about latent consistency models in Sec. 3.1. Then, we describe how to conduct latent consistency distillation for motion generation in Sec. 3.2, followed by our implementation of motion control in latent space in Sec. 3.3. The overall pipeline is illustrated in Fig. 4.

3.1 Preliminaries

The Consistency Model (CM) [62] 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 [62], the consistency function should satisfy the *self-consistency property*:

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

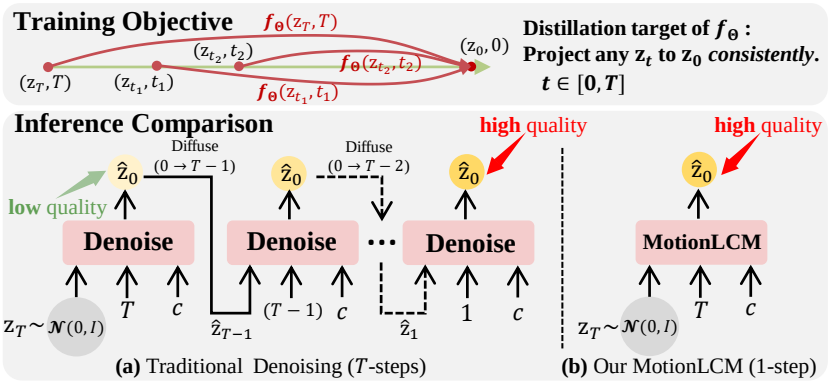


Fig. 3: The training objective of consistency distillation is to learn a consistency function $\boldsymbol{f}_{\boldsymbol{\Theta}}$, initialized with the parameters of a pre-trained diffusion model (*e.g.*, MLD [9]). This function $\boldsymbol{f}_{\boldsymbol{\Theta}}$ should projects any points (*i.e.*, $\mathbf{z}_t$) on the ODE trajectory to its solution (*i.e.*, $\mathbf{z}_0$). Once the pre-trained model [9] is distilled, unlike the traditional denoising model [65, 83] 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.

As shown in Eq. (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 Eq. (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,

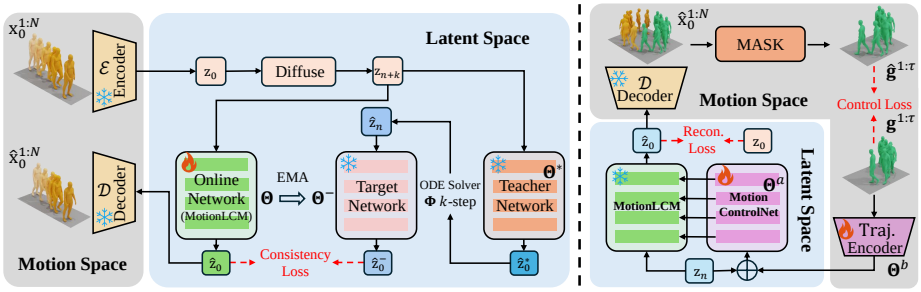
$$\mathcal{L}(\boldsymbol{\Theta}, \boldsymbol{\Theta}^-; \boldsymbol{\Phi}) = \mathbb{E}\left[d\left(\boldsymbol{f}_{\boldsymbol{\Theta}}(\mathbf{x}_{t_{n+1}}, t_{n+1}), \boldsymbol{f}_{\boldsymbol{\Theta}^-}(\hat{\mathbf{x}}^{\boldsymbol{\Phi}}_{t_n}, t_n)\right)\right], \tag{3}$$

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 [62]. Besides, $\boldsymbol{\Theta}^-$ is updated with the exponential moving average (EMA) of the parameters of $\boldsymbol{\Theta}$ [1]. In Eq. (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: $\boldsymbol{\Theta}^- \leftarrow \texttt{sg}(\mu\boldsymbol{\Theta}^- + (1-\mu)\boldsymbol{\Theta})$, where $\texttt{sg}(\cdot)$ denotes the stopgrad operation and μ satisfies $0 \leq \mu < 1$.



(a) Motion Latent Consistency Distillation (b) Motion Control in Latent Space

Fig. 4: The overview of MotionLCM. (a) *Motion Latent Consistency Distillation (Sec. 3.2).* Given a raw motion sequence $\mathbf{x}_0^{1:N}$, a pre-trained VAE [30] 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 (Sec. 3.3).* 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) [44] 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 [62] using the numerical continuous PF-ODE solver [28], LCMs [44] adopt the discrete-time schedule [41, 42, 61] to adapt to Stable Diffusion [55]. Instead of ensuring consistency between adjacent time steps $t_{n+1} \to t_n$, LCMs [44] 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) [21] 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 $[w_{\min}, w_{\max}]$ 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 MotionLCM: Motion Latent Consistency Model

Motion compression into the latent space. Motivated by [44, 62], we propose MotionLCM (Motion Latent Consistency Model) to tackle the low-efficiency

problem in motion diffusion models [65, 83], unleashing the potential of LCM in the motion generation task. Similar to MLD [9], our MotionLCM adopts a consistency model in the motion latent space. We choose MLD [9] 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 a low dimensional latent vector $\mathbf{z} = \mathcal{E}(\mathbf{x})$, which is 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 Fig. 4 (a). 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 [17] to use the redundant motion representation for our experiments, which is widely used in previous work [9, 65, 83]. As shown in Fig. 4 (a), given a raw motion sequence $\mathbf{x}_0^{1:N}$, a pre-trained VAE [30] 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 Sec. 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 [61] to predict the $\hat{\mathbf{z}}_0^-$. Since the classifier-free guidance (CFG) [21] is essential for condition alignment in diffusion models [9, 55, 65], we integrate CFG into the distillation,

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

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

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

where $d(\cdot, \cdot)$ is a distance measuring function, such as L2 loss or Huber loss [25]. As discussed in Sec. 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 [9]. According to [44], the online network and target network are initialized with the parameters of the teacher network. During the inference phase, as shown in Fig. 5, 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 [9, 65].

3.3 Controllable Motion Generation in Latent Space

After addressing the low-efficiency issue in the motion latent diffusion model [9], we delve into another exploration of real-time motion control. Inspired by the

great success of ControlNet [82] in controllable image generation [55], 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, as depicted in Fig. 1, we define the motion control task as generating motions given the initial τ poses and textual description. As shown in Fig. 4 (b), 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 [67] 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. 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 [73], 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 [30] 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's 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,

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

where λ is the weight to balance the two losses. This design enables explicit control signals to directly influence the generation process, similar to controlling other latent-free diffusion models for motion generation [29,73]. Extensive experiments demonstrate the introduced supervision is very helpful in improving the motion control performance, which will be introduced in the following section.

4 Experiments

In this section, we first present the experimental setup details in Sec. 4.1. Subsequently, we provide quantitative and qualitative comparisons to evaluate the effectiveness of our proposed MotionLCM framework in Sec. 4.2 and Sec. 4.3. Finally, we conduct comprehensive ablation studies on MotionLCM in Sec. 4.4.

4.1 Experimental setup

Datasets. We experiment on the popular HumanML3D [17] dataset, featuring 14,616 unique human motion sequences with 44,970 textual descriptions. For a fair comparison with previous methods [9, 17, 49, 65, 83], 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 [9, 17, 73]. **(1) Time cost:** We follow [9] 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 [17]. **(3) Motion diversity:** MultiModality (MModality) measures the generation diversity conditioned on the same text and Diversity calculates variance through features [17]. **(4) Condition matching:** Following [17], 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.
Implementation details. Our baseline motion diffusion model is based on MLD [9]. We reproduce MLD with higher performance. Unless otherwise specified, all our experiments are conducted on this model. For MotionLCM, we employ the AdamW [40] 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 $[w_{\min}, w_{\max}] = [5, 15]$, with the testing guidance scale set to 7.5, and adopt the EMA rate $\mu = 0.95$ by default. We use the DDIM [61] solver with skipping interval $k = 20$ and choose the Huber [25] loss as the distance measuring function d. For motion ControlNet, we use the AdamW [40] 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 implement our model using PyTorch [47] with training on an NVIDIA RTX 4090 GPU and testing on a Tesla V100 GPU.

Table 1: Comparison of text-conditional motion synthesis on HumanML3D [17] dataset. We compute suggested metrics following [17]. 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 indicate the best and the second best result. "*" denotes the reproduced version of MLD [9]. The MotionLCM in **one-step inference (30ms)** surpasses all state-of-the-art models.

Methods	AITS ↓	R-Precision ↑			FID ↓	MM Dist ↓	Diversity →	MModality ↑
		Top 1	Top 2	Top 3				
Real	-	$0.511^{\pm.003}$	$0.703^{\pm.003}$	$0.797^{\pm.002}$	$0.002^{\pm.000}$	$2.794^{\pm.008}$	$9.503^{\pm.065}$	-
Seq2Seq [37]	-	$0.180^{\pm.002}$	$0.300^{\pm.002}$	$0.396^{\pm.002}$	$11.75^{\pm.035}$	$5.529^{\pm.007}$	$6.223^{\pm.061}$	-
JL2P [2]	-	$0.246^{\pm.001}$	$0.387^{\pm.002}$	$0.486^{\pm.002}$	$11.02^{\pm.046}$	$5.296^{\pm.008}$	$7.676^{\pm.058}$	-
T2G [5]	-	$0.165^{\pm.001}$	$0.267^{\pm.002}$	$0.345^{\pm.002}$	$7.664^{\pm.030}$	$6.030^{\pm.008}$	$6.409^{\pm.071}$	-
Hier [14]	-	$0.301^{\pm.002}$	$0.425^{\pm.002}$	$0.552^{\pm.004}$	$6.532^{\pm.024}$	$5.012^{\pm.018}$	$8.332^{\pm.042}$	-
TEMOS [49]	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 [17]	0.038	$0.457^{\pm.002}$	$0.639^{\pm.003}$	$0.740^{\pm.003}$	$1.067^{\pm.002}$	$3.340^{\pm.008}$	$9.188^{\pm.002}$	$2.090^{\pm.083}$
MDM [65]	24.74	$0.320^{\pm.005}$	$0.498^{\pm.004}$	$0.611^{\pm.007}$	$0.544^{\pm.044}$	$5.566^{\pm.027}$	$\mathbf{9.559}^{\pm.086}$	$\mathbf{2.799}^{\pm.072}$
MotionDiffuse [83]	14.74	$0.491^{\pm.001}$	$0.681^{\pm.001}$	$0.782^{\pm.001}$	$0.630^{\pm.001}$	$3.113^{\pm.001}$	$\underline{9.410}^{\pm.049}$	$1.553^{\pm.042}$
MLD [9]	0.217	$0.481^{\pm.003}$	$0.673^{\pm.003}$	$0.772^{\pm.002}$	$0.473^{\pm.013}$	$3.196^{\pm.010}$	$9.724^{\pm.082}$	$\underline{2.413}^{\pm.079}$
MLD* [9]	0.225	$\underline{0.504}^{\pm.002}$	$0.698^{\pm.003}$	$0.796^{\pm.002}$	$0.450^{\pm.011}$	$3.052^{\pm.009}$	$9.634^{\pm.064}$	$2.267^{\pm.082}$
MotionLCM (1-step)	**0.030**	$0.502^{\pm.003}$	$\underline{0.701}^{\pm.002}$	$\underline{0.803}^{\pm.002}$	$0.467^{\pm.012}$	$3.022^{\pm.009}$	$9.631^{\pm.066}$	$2.172^{\pm.082}$
MotionLCM (2-step)	$\underline{0.035}$	$\mathbf{0.505}^{\pm.003}$	$\mathbf{0.705}^{\pm.002}$	$\mathbf{0.805}^{\pm.002}$	$\underline{0.368}^{\pm.011}$	$\mathbf{2.986}^{\pm.008}$	$9.640^{\pm.052}$	$2.187^{\pm.094}$
MotionLCM (4-step)	0.043	$0.502^{\pm.003}$	$0.698^{\pm.002}$	$0.798^{\pm.002}$	$\mathbf{0.304}^{\pm.012}$	$\underline{3.012}^{\pm.007}$	$9.607^{\pm.066}$	$2.259^{\pm.092}$

4.2 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 [17] with suggested metrics [17] 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 TEMOS [49], T2M [17], MDM [65], MotionDiffuse [83] and MLD [9]. The results are borrowed from MLD [9]. The deterministic methods [2,5,14,37], are unable to produce diverse results from a single input text and thus we leave their MModality metrics empty. For the quantitative results, as shown in Tab. 1, 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 [9,65,83] and even surpasses MLD [9] by an order of magnitude. Furthermore, despite employing only one-step inference, our MotionLCM can approximate or even surpass the performance of MLD [9] (DDIM [61] 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. For the qualitative results, as shown in Fig. 5, MotionLCM not only accelerates motion generation to real-time speed but also delivers high-quality outputs, closely aligning with the textual descriptions.

4.3 Comparisons on Controllable Motion Generation

As shown in Tab. 2, we compare our MotionLCM with OmniControl [73] and MLD [9]. We observe that OmniControl struggles with multi-joint control and

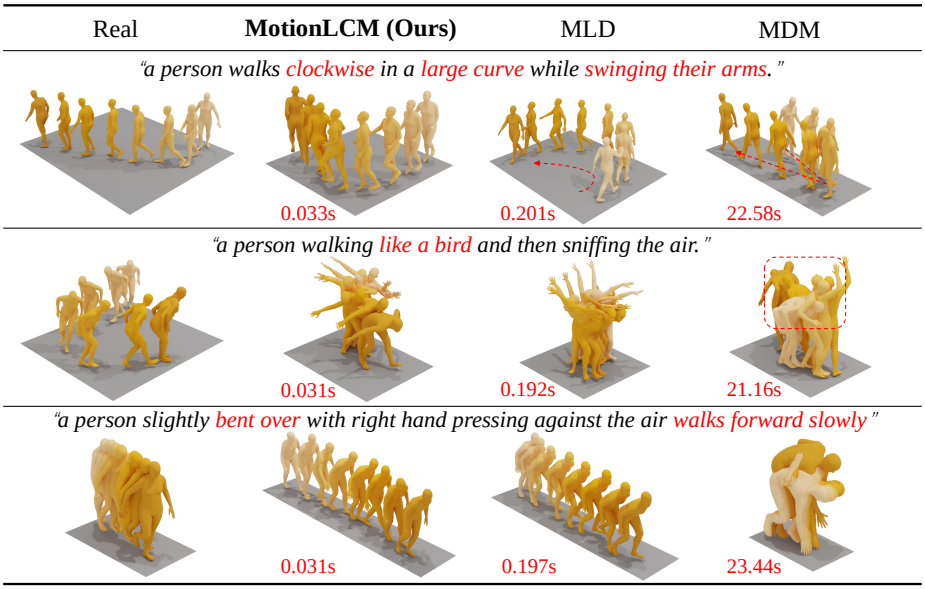


Fig. 5: Qualitative comparison of the state-of-the-art methods in the text-to-motion task. With only one-step inference, MotionLCM achieves the fastest motion generation while producing high-quality movements that closely match the textual descriptions.

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 [9] in motion control performance. This demonstrates that the latent generated by MotionLCM is more effective for training motion ControlNet compared to MLD [9]. In terms of inference speed, MotionLCM (1-step) is **1929**× faster compared to OmniControl [73] and **13**× faster than MLD [9]. For qualitative results, as shown in Fig. 6, OmniControl fails to control the initial poses in the second example and does not generate motion that aligns with the text in the third case. However, our MotionLCM not only adheres to the control of the initial poses but also generates motions that match the textual descriptions.

4.4 Ablation Studies

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

Table 2: Comparison of motion control on HumanML3D [17] dataset. **Bold** indicates the best result. Our MotionLCM outperforms OmniControl [73] and MLD [9] regarding generation quality, control performance, and inference speed. "LC" and "MC" refer to the control supervision introduced in the latent space and motion space.

Methods	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
OmniControl [73]	81.00	2.328	0.557	8.867	0.3362	0.0322	0.0977
MLD [9] (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
MLD [9] (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

Table 3: 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 Tab. 1 and adopt a one-step inference setting with the testing CFG scale of 7.5 for fair comparison.

Methods	R-Precision ↑ Top 1	FID ↓	MM Dist ↓	Diversity →	MModality ↑
Real	$0.511^{\pm.003}$	$0.002^{\pm.000}$	$2.794^{\pm.008}$	$9.503^{\pm.065}$	-
MotionLCM ($w \in [5, 15]$)	$\mathbf{0.502}^{\pm.003}$	$0.467^{\pm.012}$	$3.022^{\pm.009}$	$9.631^{\pm.066}$	$2.172^{\pm.082}$
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 ($\mu = 0.95$)	$\mathbf{0.502}^{\pm.003}$	$0.467^{\pm.012}$	$3.022^{\pm.009}$	$9.631^{\pm.066}$	$2.172^{\pm.082}$
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 ($k = 50$)	$0.488^{\pm.003}$	$0.547^{\pm.011}$	$3.096^{\pm.010}$	$\mathbf{9.511}^{\pm.074}$	$\mathbf{2.324}^{\pm.091}$
MotionLCM ($k = 20$)	$\mathbf{0.502}^{\pm.003}$	$0.467^{\pm.012}$	$3.022^{\pm.009}$	$9.631^{\pm.066}$	$2.172^{\pm.082}$
MotionLCM ($k = 10$)	$0.497^{\pm.003}$	$0.449^{\pm.009}$	$\mathbf{3.017}^{\pm.010}$	$9.693^{\pm.075}$	$2.133^{\pm.086}$
MotionLCM ($k = 5$)	$0.488^{\pm.003}$	$\mathbf{0.438}^{\pm.009}$	$3.044^{\pm.009}$	$9.647^{\pm.074}$	$2.147^{\pm.083}$
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)	$\mathbf{0.502}^{\pm.003}$	$0.467^{\pm.012}$	$3.022^{\pm.009}$	$9.631^{\pm.066}$	$2.172^{\pm.082}$
MotionLCM (w/ L2)	$0.486^{\pm.002}$	$0.622^{\pm.010}$	$3.114^{\pm.009}$	$9.573^{\pm.069}$	$2.218^{\pm.086}$

inference in Tab. 3. 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 $\boldsymbol{\Theta}^-$ 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 [25] significantly outperforms the L2 loss, demonstrating its superior robustness.

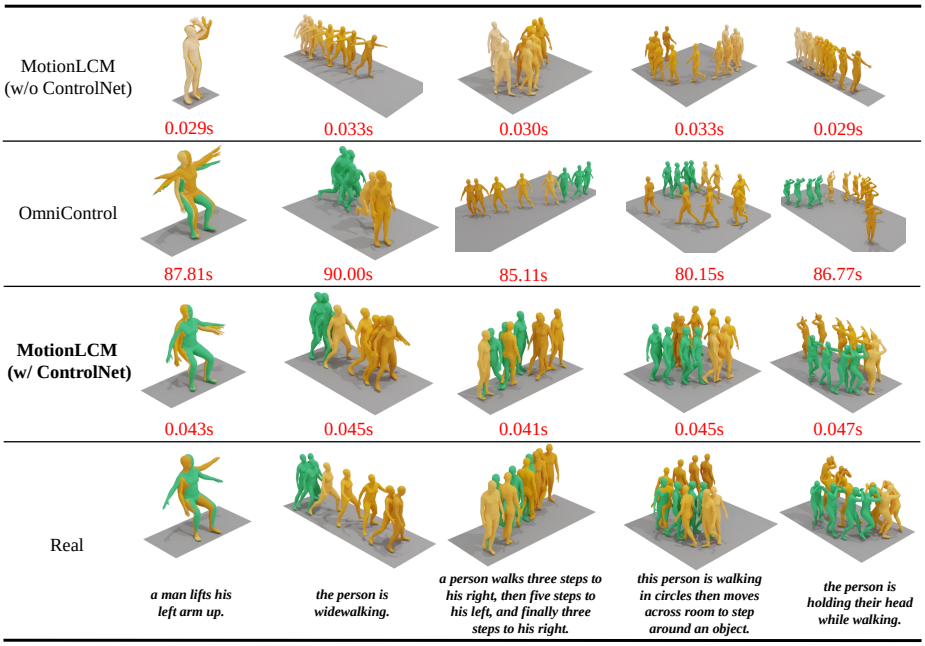


Fig. 6: Qualitative comparison of the state-of-the-art methods in the motion control task. We provide the visualized motion results and real references from five prompts. Compared to OmniControl [73], MotionLCM with ControlNet not only generates the initial poses that accurately follow the given multi-joint trajectories (*i.e.*, the green poses in real references) but also produces motions that closely align with the texts.

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 Tab. 4. 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 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.

Impact of different control ratios τ and number of control joints K. In Tab. 5, 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 $K = 22$.

Table 4: 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

Table 5: Ablation study on different control ratios τ and number of control joints K. We report the results of (1, 2, 4)-step inference."*" denotes the default training setting.

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, $\tau = 0.25$, $K = 6$)	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

5 Conclusion

This work proposes an efficient controllable motion generation framework, MotionLCM. By introducing latent consistency distillation, MotionLCM enjoys the trade-off between runtime efficiency and generation quality. Moreover, thanks to the motion ControlNet manipulation in the latent space, our method obtains excellent controlling ability with given conditions. Extensive experimental results show the effectiveness of the proposed method. As the VAE of MLD lacks explicit temporal modeling, the MotionLCM cannot achieve a good temporal explanation. Therefore, our future work will lie in developing a more explainable compression architecture for efficient motion control.

Acknowledgements

The research is supported by Shenzhen Ubiquitous Data Enabling Key Lab under grant ZDSYS20220527171406015 and CCF-Tencent Rhino-Bird Open Research Fund. This project is also supported by Shanghai Artificial Intelligence Laboratory. The author team would like to acknowledge Yiming Xie, Zhiyang Dou, and Shunlin Lu for their helpful technical discussions and suggestions.

References

1. Ahn, H., Ha, T., Choi, Y., Yoo, H., Oh, S.: Text2action: Generative adversarial synthesis from language to action. In: ICRA. pp. 5915–5920 (2018)
2. Ahuja, C., Morency, L.P.: Language2pose: Natural language grounded pose forecasting. In: 3DV. pp. 719–728 (2019)
3. Athanasiou, N., Petrovich, M., Black, M.J., Varol, G.: Teach: Temporal action composition for 3d humans. In: 3DV. pp. 414–423 (2022)
4. Barquero, G., Escalera, S., Palmero, C.: Seamless human motion composition with blended positional encodings. In: CVPR. pp. 457–469 (2024)
5. Bhattacharya, U., Rewkowski, N., Banerjee, A., Guhan, P., Bera, A., Manocha, D.: Text2gestures: A transformer-based network for generating emotive body gestures for virtual agents. In: VR. pp. 1–10 (2021)
6. Cervantes, P., Sekikawa, Y., Sato, I., Shinoda, K.: Implicit neural representations for variable length human motion generation. In: ECCV. pp. 356–372 (2022)
7. Chen, L.H., Lu, S., Zeng, A., Zhang, H., Wang, B., Zhang, R., Zhang, L.: Motionllm: Understanding human behaviors from human motions and videos. arXiv preprint arXiv:2405.20340 (2024)
8. Chen, L.H., Zhang, J., Li, Y., Pang, Y., Xia, X., Liu, T.: Humanmac: Masked motion completion for human motion prediction. In: ICCV. pp. 9544–9555 (2023)
9. Chen, X., Jiang, B., Liu, W., Huang, Z., Fu, B., Chen, T., Yu, G.: Executing your commands via motion diffusion in latent space. In: CVPR. pp. 18000–18010 (2023)
10. Cong, P., Dou, Z.W., Ren, Y., Yin, W., Cheng, K., Sun, Y., Long, X., Zhu, X., Ma, Y.: Laserhuman: Language-guided scene-aware human motion generation in free environment. arXiv preprint arXiv:2403.13307 (2024)
11. Dabral, R., Mughal, M.H., Golyanik, V., Theobalt, C.: Mofusion: A framework for denoising-diffusion-based motion synthesis. In: CVPR. pp. 9760–9770 (2023)
12. Dou, Z., Chen, X., Fan, Q., Komura, T., Wang, W.: C· ase: Learning conditional adversarial skill embeddings for physics-based characters. In: SIGGRAPH Asia. pp. 1–11 (2023)
13. Fan, K., Tang, J., Cao, W., Yi, R., Li, M., Gong, J., Zhang, J., Wang, Y., Wang, C., Ma, L.: Freemotion: A unified framework for number-free text-to-motion synthesis. arXiv preprint arXiv:2405.15763 (2024)
14. Ghosh, A., Cheema, N., Oguz, C., Theobalt, C., Slusallek, P.: Synthesis of compositional animations from textual descriptions. In: ICCV. pp. 1396–1406 (2021)
15. Guo, C., Mu, Y., Javed, M.G., Wang, S., Cheng, L.: Momask: Generative masked modeling of 3d human motions. In: CVPR. pp. 1900–1910 (2024)
16. Guo, C., Mu, Y., Zuo, X., Dai, P., Yan, Y., Lu, J., Cheng, L.: Generative human motion stylization in latent space. arXiv preprint arXiv:2401.13505 (2024)
17. Guo, C., Zou, S., Zuo, X., Wang, S., Ji, W., Li, X., Cheng, L.: Generating diverse and natural 3d human motions from text. In: CVPR. pp. 5152–5161 (2022)

18. Guo, C., Zuo, X., Wang, S., Cheng, L.: Tm2t: Stochastic and tokenized modeling for the reciprocal generation of 3d human motions and texts. In: ECCV. pp. 580–597 (2022)
19. Guo, C., Zuo, X., Wang, S., Zou, S., Sun, Q., Deng, A., Gong, M., Cheng, L.: Action2motion: Conditioned generation of 3d human motions. In: ACMMM. pp. 2021–2029 (2020)
20. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. NeurIPS pp. 6840–6851 (2020)
21. Ho, J., Salimans, T.: Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598 (2022)
22. Holden, D., Komura, T., Saito, J.: Phase-functioned neural networks for character control. TOG **36**(4), 1–13 (2017)
23. Holden, D., Saito, J., Komura, T.: A deep learning framework for character motion synthesis and editing. TOG **35**(4), 1–11 (2016)
24. Huang, Y., Yang, H., Luo, C., Wang, Y., Xu, S., Zhang, Z., Zhang, M., Peng, J.: Stablemofusion: Towards robust and efficient diffusion-based motion generation framework. arXiv preprint arXiv:2405.05691 (2024)
25. Huber, P.J.: Robust estimation of a location parameter. The Annals of Mathematical Statistics **35**(1), 73–101 (1964)
26. Ji, Y., Xu, F., Yang, Y., Shen, F., Shen, H.T., Zheng, W.S.: A large-scale rgb-d database for arbitrary-view human action recognition. In: ACMMM. pp. 1510–1518 (2018)
27. Jiang, B., Chen, X., Liu, W., Yu, J., Yu, G., Chen, T.: Motiongpt: Human motion as a foreign language. NeurIPS (2024)
28. Karras, T., Aittala, M., Aila, T., Laine, S.: Elucidating the design space of diffusion-based generative models. NeurIPS pp. 26565–26577 (2022)
29. Karunratanakul, K., Preechakul, K., Suwajanakorn, S., Tang, S.: Guided motion diffusion for controllable human motion synthesis. In: CVPR. pp. 2151–2162 (2023)
30. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
31. Lee, T., Moon, G., Lee, K.M.: Multiact: Long-term 3d human motion generation from multiple action labels. In: AAAI. pp. 1231–1239 (2023)
32. Li, B., Zhao, Y., Zhelun, S., Sheng, L.: Danceformer: Music conditioned 3d dance generation with parametric motion transformer. In: AAAI. pp. 1272–1279 (2022)
33. Li, R., Zhang, Y., Zhang, Y., Zhang, H., Guo, J., Zhang, Y., Liu, Y., Li, X.: Lodge: A coarse to fine diffusion network for long dance generation guided by the characteristic dance primitives. In: CVPR. pp. 1524–1534 (2024)
34. Li, R., Zhao, J., Zhang, Y., Su, M., Ren, Z., Zhang, H., Tang, Y., Li, X.: Finedance: A fine-grained choreography dataset for 3d full body dance generation. In: ICCV. pp. 10234–10243 (2023)
35. Li, R., Yang, S., Ross, D.A., Kanazawa, A.: Ai choreographer: Music conditioned 3d dance generation with aist++. In: ICCV. pp. 13401–13412 (2021)
36. Li, T., Qiao, C., Ren, G., Yin, K., Ha, S.: Aamdm: Accelerated auto-regressive motion diffusion model. In: CVPR. pp. 1813–1823 (2024)
37. Lin, A.S., Wu, L., Corona, R., Tai, K., Huang, Q., Mooney, R.J.: Generating animated videos of human activities from natural language descriptions. Learning **1**(2018), 1 (2018)
38. Lin, X., Amer, M.R.: Human motion modeling using dvgans. arXiv preprint arXiv:1804.10652 (2018)

39. Liu, J., Dai, W., Wang, C., Cheng, Y., Tang, Y., Tong, X.: Plan, posture and go: Towards open-world text-to-motion generation. arXiv preprint arXiv:2312.14828 (2023)
40. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
41. Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., Zhu, J.: Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. NeurIPS pp. 5775–5787 (2022)
42. Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., Zhu, J.: Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. arXiv preprint arXiv:2211.01095 (2022)
43. Lu, S., Chen, L.H., Zeng, A., Lin, J., Zhang, R., Zhang, L., Shum, H.Y.: Humantomato: Text-aligned whole-body motion generation. arXiv preprint arXiv:2310.12978 (2023)
44. Luo, S., Tan, Y., Huang, L., Li, J., Zhao, H.: Latent consistency models: Synthesizing high-resolution images with few-step inference. arXiv preprint arXiv:2310.04378 (2023)
45. Mahmood, N., Ghorbani, N., Troje, N.F., Pons-Moll, G., Black, M.J.: Amass: Archive of motion capture as surface shapes. In: ICCV. pp. 5442–5451 (2019)
46. Nichol, A.Q., Dhariwal, P.: Improved denoising diffusion probabilistic models. In: ICML. pp. 8162–8171 (2021)
47. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. NeurIPS (2019)
48. Petrovich, M., Black, M.J., Varol, G.: Action-conditioned 3d human motion synthesis with transformer vae. In: ICCV. pp. 10985–10995 (2021)
49. Petrovich, M., Black, M.J., Varol, G.: Temos: Generating diverse human motions from textual descriptions. In: ECCV. pp. 480–497 (2022)
50. Petrovich, M., Black, M.J., Varol, G.: Tmr: Text-to-motion retrieval using contrastive 3d human motion synthesis. In: ICCV. pp. 9488–9497 (2023)
51. Petrovich, M., Litany, O., Iqbal, U., Black, M.J., Varol, G., Bin Peng, X., Rempe, D.: Multi-track timeline control for text-driven 3d human motion generation. In: CVPRW. pp. 1911–1921 (2024)
52. Plappert, M., Mandery, C., Asfour, T.: The kit motion-language dataset. Big data **4**(4), 236–252 (2016)
53. Punnakkal, A.R., Chandrasekaran, A., Athanasiou, N., Quiros-Ramirez, A., Black, M.J.: Babel: Bodies, action and behavior with english labels. In: CVPR. pp. 722–731 (2021)
54. Raab, S., Leibovitch, I., Li, P., Aberman, K., Sorkine-Hornung, O., Cohen-Or, D.: Modi: Unconditional motion synthesis from diverse data. In: CVPR. pp. 13873–13883 (2023)
55. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR. pp. 10684–10695 (2022)
56. Shafir, Y., Tevet, G., Kapon, R., Bermano, A.H.: Human motion diffusion as a generative prior. In: ICLR (2024)
57. Shahroudy, A., Liu, J., Ng, T.T., Wang, G.: Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In: CVPR. pp. 1010–1019 (2016)
58. Shi, Y., Wang, J., Jiang, X., Dai, B.: Controllable motion diffusion model. arXiv preprint arXiv:2306.00416 (2023)

59. Siyao, L., Yu, W., Gu, T., Lin, C., Wang, Q., Qian, C., Loy, C.C., Liu, Z.: Bailando: 3d dance generation by actor-critic gpt with choreographic memory. In: CVPR. pp. 11050–11059 (2022)
60. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: ICML. pp. 2256–2265 (2015)
61. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: ICLR (2021)
62. Song, Y., Dhariwal, P., Chen, M., Sutskever, I.: Consistency models. In: ICML (2023)
63. Tang, Y., Liu, J., Liu, A., Yang, B., Dai, W., Rao, Y., Lu, J., Zhou, J., Li, X.: Flag3d: A 3d fitness activity dataset with language instruction. In: CVPR. pp. 22106–22117 (2023)
64. Tevet, G., Gordon, B., Hertz, A., Bermano, A.H., Cohen-Or, D.: Motionclip: Exposing human motion generation to clip space. In: ECCV. pp. 358–374 (2022)
65. Tevet, G., Raab, S., Gordon, B., Shafir, Y., Cohen-Or, D., Bermano, A.H.: Human motion diffusion model. In: ICLR (2022)
66. Tseng, J., Castellon, R., Liu, K.: Edge: Editable dance generation from music. In: CVPR. pp. 448–458 (2023)
67. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. NeurIPS (2017)
68. Wan, W., Dou, Z., Komura, T., Wang, W., Jayaraman, D., Liu, L.: Tlcontrol: Trajectory and language control for human motion synthesis. arXiv preprint arXiv:2311.17135 (2023)
69. Wang, Z., Chen, Y., Jia, B., Li, P., Zhang, J., Zhang, J., Liu, T., Zhu, Y., Liang, W., Huang, S.: Move as you say interact as you can: Language-guided human motion generation with scene affordance. In: CVPR. pp. 433–444 (2024)
70. Wang, Z., Chen, Y., Liu, T., Zhu, Y., Liang, W., Huang, S.: Humanise: Language-conditioned human motion generation in 3d scenes. NeurIPS pp. 14959–14971 (2022)
71. Wang, Z., Wang, J., Lin, D., Dai, B.: Intercontrol: Generate human motion interactions by controlling every joint. arXiv preprint arXiv:2311.15864 (2023)
72. Xiao, Z., Wang, T., Wang, J., Cao, J., Zhang, W., Dai, B., Lin, D., Pang, J.: Unified human-scene interaction via prompted chain-of-contacts. In: ICLR (2024)
73. Xie, Y., Jampani, V., Zhong, L., Sun, D., Jiang, H.: Omnicontrol: Control any joint at any time for human motion generation. In: ICLR (2024)
74. Xu, L., Lv, X., Yan, Y., Jin, X., Wu, S., Xu, C., Liu, Y., Zhou, Y., Rao, F., Sheng, X., et al.: Inter-x: Towards versatile human-human interaction analysis. In: CVPR. pp. 22260–22271 (2024)
75. Xu, L., Song, Z., Wang, D., Su, J., Fang, Z., Ding, C., Gan, W., Yan, Y., Jin, X., Yang, X., et al.: Actformer: A gan-based transformer towards general action-conditioned 3d human motion generation. In: ICCV. pp. 2228–2238 (2023)
76. Xu, L., Zhou, Y., Yan, Y., Jin, X., Zhu, W., Rao, F., Yang, X., Zeng, W.: Regennet: Towards human action-reaction synthesis. In: CVPR. pp. 1759–1769 (2024)
77. Yan, S., Li, Z., Xiong, Y., Yan, H., Lin, D.: Convolutional sequence generation for skeleton-based action synthesis. In: ICCV. pp. 4394–4402 (2019)
78. Yuan, Y., Song, J., Iqbal, U., Vahdat, A., Kautz, J.: Physdiff: Physics-guided human motion diffusion model. In: ICCV. pp. 16010–16021 (2023)
79. Zhang, B., Cheng, Y., Wang, C., Zhang, T., Yang, J., Tang, Y., Zhao, F., Chen, D., Guo, B.: Rodinhd: High-fidelity 3d avatar generation with diffusion models. arXiv preprint arXiv:2407.06938 (2024)

80. Zhang, B., Cheng, Y., Yang, J., Wang, C., Zhao, F., Tang, Y., Chen, D., Guo, B.: Gaussiancube: Structuring gaussian splatting using optimal transport for 3d generative modeling. arXiv preprint arXiv:2403.19655 (2024)
81. Zhang, J., Zhang, Y., Cun, X., Zhang, Y., Zhao, H., Lu, H., Shen, X., Shan, Y.: Generating human motion from textual descriptions with discrete representations. In: CVPR. pp. 14730–14740 (2023)
82. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: ICCV. pp. 3836–3847 (2023)
83. Zhang, M., Cai, Z., Pan, L., Hong, F., Guo, X., Yang, L., Liu, Z.: Motiondiffuse: Text-driven human motion generation with diffusion model. arXiv preprint arXiv:2208.15001 (2022)
84. Zhao, R., Su, H., Ji, Q.: Bayesian adversarial human motion synthesis. In: CVPR. pp. 6225–6234 (2020)
85. Zhong, L., Xie, Y., Jampani, V., Sun, D., Jiang, H.: Smoodi: Stylized motion diffusion model. arXiv preprint arXiv:2407.12783 (2024)
86. Zhou, W., Dou, Z., Cao, Z., Liao, Z., Wang, J., Wang, W., Liu, Y., Komura, T., Wang, W., Liu, L.: Emdm: Efficient motion diffusion model for fast, high-quality motion generation. arXiv preprint arXiv:2312.02256 (2023)