LOW-REDUNDANCY MODELING FOR LATENT-FREE HUMAN MOTION GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Many recent approaches to human motion generation operate in latent spaces, which require two-stage training and inherit the reconstruction errors of their autoencoders. We revisit generation in raw motion space and identify two key bottlenecks: feature redundancy, which arises from over-parameterized representations, and state redundancy, which arises from invalid off-manifold states and unnatural motions. Building on this observation, we instantiate DirectMotion, a minimalist baseline that combines three simple designs: a compact representation consisting of root features and joint rotations, probability paths constrained to the state space of human motion, and clean-sequence prediction. With only a lightweight DiT network, DirectMotion performs on par with larger latent-space models on the HumanML3D and SnapMoGen datasets. Ablation studies confirm the contribution of removing each form of redundancy. We hope these results encourage the community to reconsider latent-free generation as an inherently simpler paradigm.

1 INTRODUCTION

Inspired by latent generative models in the image domain Rombach et al. (2022); Peebles & Xie (2023), latent-space human motion generation Zhang et al. (2023a); Chen et al. (2023); Guo et al. (2024) has become the prevailing paradigm. However, image and motion data differ substantially. Pixel space is widely acknowledged as highly redundant and semantically sparse He et al. (2022); An et al. (2026); Chen et al. (2026a), where individual pixel values typically lack clear semantic meaning, making it well-suited for latent space compression Chen et al. (2026b); Shao & Li (2025). In contrast, motion data possesses a compact structure in which every numerical value carries a clear physical interpretation. Although latent-space approaches have received considerable research attention Xiao et al. (2025); Jeong et al. (2025), methods that operate directly in motion space Tevet et al. (2022); Zhang et al. (2024a) remain a promising direction for further exploration.

The dimensionality of the modeling space is a critical determinant of generative model performance. We identify that, although motion is more compact than pixels, redundancy still exists within it and is worth removing. The first is *feature redundancy*: existing methods typically encode motion through joint coordinates, velocities, rotations, and foot contact indicators, yielding an over-parameterized representation whose redundancy has been repeatedly noted Meng et al. (2025b). Such features are the raw materials from which autoencoders extract expressive latents Guo et al. (2022), but they may be a liability once a model must generate them rather than merely consume them.

However, discarding redundant features alone is not sufficient. The second, subtler form of redundancy is *state redundancy*. Geometrically, the minimal yet sufficient state space of human motion is the product of a Euclidean space (root features) and several rotation manifolds $\mathrm{SO}(3)$ (the joint rotations). This state space is itself a curved manifold, which we refer to as the *state manifold*: its intrinsic dimension is low, yet it is embedded in a high-dimensional Euclidean ambient space. It is precisely this excess dimensionality that fills the ambient space with redundant, invalid states lying off the state manifold. Building further on the manifold hypothesis Carlsson (2009); Li & He (2025), natural motion data occupies a lower-dimensional submanifold of the state manifold, which we refer to as the *data manifold*. Invalid states in the Euclidean ambient space, together with unnatural states on the state manifold, jointly give rise to state redundancy. This may explain why latent compression has been so effective: autoencoders compress the Euclidean ambient space into a latent space where valid states are densely represented, thereby attenuating state redundancy.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

The central idea of this work is that both forms of redundancy discussed above can be alleviated within the framework of motion-space diffusion. The remedy is straightforward: discard the redundant features and prevent the model from processing invalid states wherever possible. These solutions may warrant further exploration in motion generation, alongside approaches based on latent compression Meng et al. (2025b); Zhang et al. (2025). In the pursuit of a simple but effective paradigm, this paper introduces no complex network architectures or training mechanisms. We ask: *after removing the redundancies, how far can a deliberately simple model go?*

To answer this, we instantiate **DirectMotion**, a denoising model for motion generation. We deliberately position DirectMotion as a minimalist baseline rather than a sophisticated method: *no carefully designed tokenizer Zhang et al. (2023a); Guo et al. (2024), no auxiliary losses Tevet et al. (2022); Zhang et al. (2024a), and no elaborate training procedure Li et al. (2026); Jeong et al. (2025)*. Through DirectMotion, we study three design choices, mapping onto the two forms of redundancy we have identified. (1) We encode a motion only by the root features and joint rotation; this removes feature redundancy at its source. (2) We constrain the probability path to remain on the state manifold, which ensures that the model learns only within the low-dimensional manifold and never visits invalid off-manifold states. (3) We make the model predict clean motion sequences rather than velocity fields containing noise, thereby shifting the supervision target onto the lower-dimensional natural data manifold. While (2) excludes the invalid half of state redundancy, (3) mitigates the unnatural half. Through ablation of the designs, we can not only validate our hypothesis but also reveal the contribution of the minimalist modeling philosophy of low redundancy to motion generation.

The experimental results are perhaps the most informative part of this work. This deliberately minimalist, latent-free baseline outperforms far more elaborate latent-space methods on HumanML3D Guo et al. (2022) and remains competitive on the larger SnapMoGen Guo et al. (2025) dataset, with a universal lightweight DiT network. Ablation experiments on the three design choices isolate the contribution of each form of redundancy we remove. Furthermore, we conceptually investigate whether learning in the raw motion space allows DirectMotion to form a motion prior. Such a prior facilitates downstream applications including path control and motion editing, for which latent models typically require additional modules or extra training since latent representations lack explicit physical semantics Dai et al. (2024); Pinyoanuntapong et al. (2025). We hope these findings encourage the community to revisit generation in raw motion space.

Our contributions are summarized as follows. (1) We revisit human motion generation in raw motion space and identify feature redundancy and state redundancy as bottlenecks for current models, motivating a low-redundancy formulation. (2) We instantiate this formulation in DirectMotion, a latent-free minimalist baseline defined by three choices: a minimalist state manifold, probability paths on the state manifold, and clean-sequence prediction, which respectively remove feature redundancy and mitigate the state redundancy. (3) We show that this model outperforms latent-space methods on HumanML3D and remains competitive on SnapMoGen with a lightweight DiT backbone, and that ablations recover the contribution of each form of redundancy we remove.

2 Related Work

2.1 Human Motion Generation

Modern human motion generation models can be broadly categorized into three classes: diffusion models Tevet et al. (2022); Zhang et al. (2024a); Chen et al. (2023); Mohamed et al. (2024); Zhang et al. (2023b), GPT-like models Zhang et al. (2023a; 2024b); Wu et al. (2025); Lu et al. (2025), and BERT-like models Pinyoanuntapong et al. (2024b); Guo et al. (2024); Pinyoanuntapong et al. (2024a); Jeong et al. (2025); Pinyoanuntapong et al. (2025). Early diffusion-based approaches Tevet et al. (2022); Zhang et al. (2024a) performed diffusion directly in the raw motion space. Later, inspired by latent diffusion in image generation Xi et al. (2026), latent diffusion models Chen et al. (2023); Dai et al. (2024); Li et al. (2026); Meng et al. (2025b) were introduced to motion synthesis, improving fidelity and efficiency. Orthogonally, GPT-like and BERT-like models generate motion over a discrete latent space with compelling results. Although latent-space methods require an auxiliary VAE Kingma & Welling (2014) or VQ-VAE van den Oord et al. (2017), their favorable trade-off between performance and efficiency has made them a widely adopted approach. Nevertheless, guided by the observation that progress often advances by revisiting earlier ideas rather

than proceeding in a straight line, this work returns to motion generation in the raw motion space, demonstrating that this paradigm still holds substantial untapped potential waiting to be explored.

2.2 Motion-Space Generation

Early diffusion methods Tevet et al. (2022); Zhang et al. (2024a; 2023b); Huang et al. (2024) have explored generation in the raw motion space. These methods not only apply diffusion models to motion generation but also introduce task-specific improvements, such as incorporating kinematic auxiliary losses Tevet et al. (2022); Huang et al. (2024). Effective as they are, these methods remain rather conservative in both the space they model and the way they model it: they still inherit an over-parameterized motion representation and construct the diffusion process in a high-dimensional Euclidean space. Another observation of interest is that subsequent works Shafir et al. (2024); Yuan et al. (2023); Hunyuan (2025); Rempe et al. (2026) in the motion space place greater emphasis on kinematically controllable generation and on physical plausibility. A reason that cannot be ignored is that the learned latent representation dissolves the explicit physical meaning inherent in the raw motion data, and the decoding stage can complicate direct physical-space control and simulation-based optimization Yuan et al. (2023); Hunyuan (2025). Despite these advantages, the question of whether motion-space diffusion methods can match the effectiveness of latent-space methods remains unresolved. This work revisits this question through the lens of redundancy reduction.

3 Three Spaces in Motion-Space Diffusion Models

The motion generation involves three nested spaces: (1) the ambient space, the Euclidean space $\mathbb{R}^n$ where the n-dimensional motion representation resides; (2) the state space of all valid motion representations; and (3) the data manifold formed by natural motion data. Figure 1(a) illustrates their relationships: the state space embeds within the ambient space, and the data manifold is a subset of the state space. We observe that feature redundancy and state redundancy may be factors limiting motion-space diffusion models. Here we connect these two types of redundancy to the three spaces and resolve them into three testable hypotheses.

3.1 Simplifying State Space

A minimal but sufficient motion representation consists of a vector encoding the root features together with a set of $\mathrm{SO}(3)$ elements encoding joint rotations. Prior work generally adopts redundant representations that augment this with features such as joint coordinates, velocities, and foot contact indicators Guo et al. (2022; 2025), which introduces feature redundancy. Intuitively, such redundancy substantially increases data dimensionality, which consumes model capacity and may hinder generative modeling in the raw motion space.

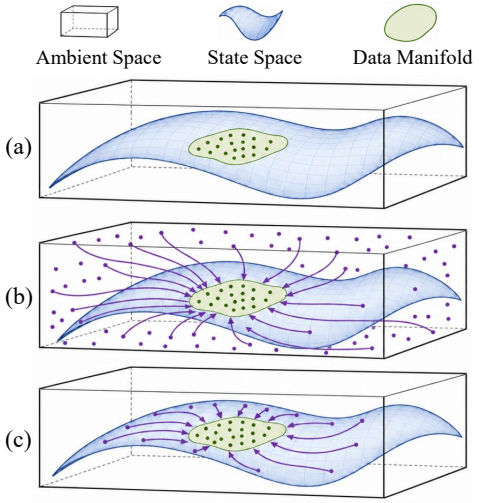


Figure 1: **The three spaces in motion-space diffusion models**. (a) The three spaces are nested. (b) Denoising is performed across the ambient space. (c) Denoising is performed only in the state space.

We are not the first to question the use of redundant representations. Recent work Meng et al. (2025b;a) investigates this issue and proposes a simplified representation that encodes motion using only coordinates, achieving improved performance. However, coordinates alone do not provide a sufficient pose representation, as they cannot characterize rotation about a bone axis and therefore omit essential rotational information. Whether generation using a minimal yet sufficient representation is beneficial therefore remains an open question. Accordingly, we propose the hypothesis **H1: A minimal sufficient representation may be easier to generate than a redundant one.**

3.2 Modeling on State Space

Representations of $\mathrm{SO}(3)$ elements are subject to intrinsic constraints. For example, a rotation matrix is orthogonal with determinant one. These constraints confine valid $\mathrm{SO}(3)$ elements to a low-dimensional manifold embedded in the Euclidean ambient space, such as the continuous 6D representation Zhou et al. (2019) lying on the Stiefel manifold $V_2(\mathbb{R}^3)$ embedded in $\mathbb{R}^6$. Consequently, the state space of motion is a low-dimensional manifold within the ambient space.

The dimension gap between the ambient space and the state manifold implies the existence of invalid states, which is one source of state redundancy. This redundancy becomes particularly pronounced when the state manifold is of measure zero, in which case almost every point of the ambient space is an invalid state. However, existing motion diffusion models operate in the ambient space, as shown in Fig. 1(b). The denoising path from noise to natural data is filled with invalid states. This forces the model to gradually transport probability mass from the ambient space onto the state space, which implies that a substantial portion of the model capacity may be spent on learning how to move away from invalid regions rather than on characterizing the distribution of the data within the state space.

To eliminate redundant invalid states, we propose modeling only in a low-dimensional state space. As shown in Fig. 1(c), constraining the denoising path to lie on the state manifold prevents the model from accessing the invalid states. We thus propose hypothesis **H2: generation in the low-dimensional state space may be more effective than generation in the entire ambient space.**

3.3 Supervising on Data Manifold

The motion state space captures only algebraic constraints. Natural motion data constitute only a subset of the state space. The manifold hypothesis Carlsson (2009); Chapelle et al. (2009), which is widely accepted in machine learning, further postulates that this subset forms a low-dimensional manifold embedded within the state space. Points on the state manifold but not on the data manifold, referred to as unnatural states, constitute another source of state redundancy.

Recent studies Lu et al. (2026); Li & He (2025) connect the data manifold to the prediction targets of diffusion models. These targets include noise, the velocity field, and the clean sample. Although the three prediction targets are only different parameterizations of the same probabilistic model and differ merely by a weighting coefficient at the level of the loss, these works show that they are not equivalent in substance and cannot be converted into one another through a loss weight. A key insight is that predicting noise or velocity field supervises the model to produce unnatural states distributed throughout the entire state space, whereas predicting clean samples only requires the model to produce natural states that lie on the low-dimensional data manifold. This distinction suggests that noise or velocity prediction places substantial demands on model capacity.

To prevent redundant unnatural states from consuming model capacity, we let the model predict clean motion sequences. Although this is not equivalent to producing natural states exclusively, the ultimate goal shared by all generative models, we conjecture that it induces a useful inductive bias, since the model is never supervised to output unnatural states. We thus propose hypothesis **H3: natural motion sequence on a low-dimensional data manifold may be easier to predict**.

4 Minimalist Baseline

Removing feature redundancy (**H1**), eliminating redundant invalid states (**H2**), and mitigating the redundancy introduced by unnatural states (**H3**) converge naturally into a unified design. This solution yields a denoising model that operates on a minimal motion representation, confines the denoising path to the state space, and directly predicts clean samples. Guided by these three decisions, we instantiate a minimal baseline, which we refer to as DirectMotion.

Motion representation. Our design principle requires the model to operate on a state space that is as simple as possible. To maintain compatibility with public evaluators Meng et al. (2025b); Guo et al. (2025), we adopt the motion representation $\boldsymbol{m}$ as:

$$\boldsymbol{m} = (\omega, v_x, v_z, y, \boldsymbol{r}^1, \ldots, \boldsymbol{r}^J) = (\boldsymbol{x}, \boldsymbol{r}^1, \ldots, \boldsymbol{r}^J). \tag{1}$$

Here, v_x and v_z denote the linear velocities of the root joint along the x-axis and z-axis; y represents the height of the root joint; ω is the angular velocity of the root joint around the y-axis. The terms

$\boldsymbol{r}^1, \ldots, \boldsymbol{r}^J \in \mathrm{SO}(3)$ represent the rotations of all joints. We collectively refer to v_x, v_z, y, and ω as Euclidean features, denoted by $\boldsymbol{x} = (\omega, v_x, v_z, y)$. The state space induced by this representation is therefore given by $\mathbb{R}^4 \times \mathrm{SO}(3)^J$, combining four Euclidean variables with J rotational components.

Probability paths. The probability path defines the trajectory along which noise evolves into a clean sample. Our design principle requires this path to stay confined to the state manifold, so that only valid states are visited along the way. In contrast to the stochastic paths adopted by diffusion models, we use the simpler deterministic probability path of flow matching Lipman et al. (2023).

For the Euclidean features $\boldsymbol{x}$, we construct the path $\boldsymbol{x}_t : [0,1] \to \mathbb{R}^4$ with the simplest choice, namely linear interpolation (LERP). Extending this to $\mathrm{SO}(3)$, we adopt spherical linear interpolation (SLERP) to build the path $\boldsymbol{r}_t^i : [0,1] \to \mathrm{SO}(3)$ for the rotational features $\boldsymbol{r}^1, \ldots, \boldsymbol{r}^J$:

$$\boldsymbol{x}_t = t\boldsymbol{x} + (1-t)\boldsymbol{\epsilon}, \tag{2a}$$

$$\boldsymbol{r}_t^i = \boldsymbol{\delta^i} \exp\left(t \log\left({\boldsymbol{\delta^i}}^\top \boldsymbol{r}^i\right)\right), \tag{2b}$$

where $\boldsymbol{\epsilon}$ and $\boldsymbol{\delta}^i$ are noises in $\mathbb{R}^4$ and $\mathrm{SO}(3)$, respectively; $\log$ and $\exp$ denote the logarithmic map and the exponential map on $\mathrm{SO}(3)$, respectively.

For the Euclidean noise $\boldsymbol{\epsilon}$, we adopt the standard zero-mean Gaussian distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{I}_4)$. For the rotational noise $\boldsymbol{\delta^i}$, our probability path requires the noise to lie exactly on $\mathrm{SO}(3)$, so we instead use a Gaussian defined on the Lie algebra. Concretely, we first draw a three-dimensional noise vector from a zero-mean Gaussian $\mathcal{N}(\mathbf{0}, \boldsymbol{I}_3)$ on $\mathfrak{so}(3) \cong \mathbb{R}^3$, and then map it to a noise rotation matrix $\boldsymbol{\delta}^i$ through the exponential map.

Denoising process with sample prediction. Our denoising procedure fits within the framework of Riemannian flow matching Chen & Lipman (2024), where denoising reduces to numerically integrating a velocity field defined on the tangent space. The velocity fields $\boldsymbol{v}$ and $\boldsymbol{u}^i$, corresponding to the Euclidean features and the rotational features, respectively, satisfy:

$$\boldsymbol{v} = \boldsymbol{x} - \boldsymbol{\epsilon}, \tag{3a}$$

$$\boldsymbol{u}^i = \log\left({\boldsymbol{\delta}^i}^\top \boldsymbol{r}^i\right). \tag{3b}$$

Our design principle requires the model to predict the clean states $\boldsymbol{x}$ and $\boldsymbol{r}^i$ from the noisy state $\boldsymbol{x}_t$ and $\boldsymbol{r}_t^i$. Letting $\boldsymbol{x}_{\boldsymbol{\theta}}$ and $\boldsymbol{r}_{\boldsymbol{\theta}}^i$ denote the clean sample predicted by a $\boldsymbol{\theta}$-parameterized network, the velocity field is then estimated as:

$$\boldsymbol{v}_{\boldsymbol{\theta}} = \left(\boldsymbol{x}_{\boldsymbol{\theta}} - \boldsymbol{x}_t\right) / \left(1-t\right), \tag{4a}$$

$$\boldsymbol{u}_{\boldsymbol{\theta}}^i = \log({\boldsymbol{r}^i}_t^\top \boldsymbol{r}_{\boldsymbol{\theta}}^i) / \left(1-t\right). \tag{4b}$$

We perform the numerical integration with a simple Euler method to obtain the generated results $\boldsymbol{x}_1$ and $\boldsymbol{r}_1$ from pure noise $\boldsymbol{x}_0 = \boldsymbol{\epsilon}$ and $\boldsymbol{r}_0^i = \boldsymbol{\delta}^i$:

$$\boldsymbol{x}_{t+\Delta t} = \boldsymbol{x}_t + \Delta t\, \boldsymbol{v}_{\boldsymbol{\theta}}, \tag{5a}$$

$$\boldsymbol{r}_{t+\Delta t}^i = \boldsymbol{r}_t^i \exp(\Delta t\, \boldsymbol{u}_{\boldsymbol{\theta}}^i), \tag{5b}$$

where Δt is the step size of the time discretization, which is typically an inference hyperparameter.

Loss function. We employ a neural network $\boldsymbol{m}_{\boldsymbol{\theta}} = (\boldsymbol{x}_{\boldsymbol{\theta}}, \boldsymbol{r}_{\boldsymbol{\theta}}^1, \ldots, \boldsymbol{r}_{\boldsymbol{\theta}}^J)$ to predict the Euclidean feature together with all rotational features. Following a minimalist principle, our loss function includes only the prediction loss without adding any auxiliary losses:

$$\mathcal{L}_{\boldsymbol{\theta}} = \mathbb{E}_{t,\boldsymbol{m},\boldsymbol{\epsilon},\boldsymbol{\delta}} \left(\lambda \|\boldsymbol{x}_{\boldsymbol{\theta}} - \boldsymbol{x}\|_2 + \frac{1}{J} \sum_{i=1}^{J} \|\boldsymbol{r}_{\boldsymbol{\theta}}^i - \boldsymbol{r}^i\|_2 \right), \tag{6}$$

where λ is a hyperparameter that balances the Euclidean loss against the rotational loss. It is worth noting that we apply L2 loss directly to the rotation matrix rather than the SO(3) geodesic loss, which provides better numerical stability. Discussion on geodesic loss can be found in the Appendix A.2.

Parameterization of rotations. Beyond rotation matrices, 3D rotations admit several alternative representations, such as quaternions and Euler angles, and we include them in our design space. The guiding principle is that, whichever parameterization is used, interpolation and velocity field estimation are performed in quaternion space, while the loss is computed in the space of rotation matrices. The choice of parameterization thus affects only the input and the output of the model. We examine the influence of different rotation parameterizations in the ablation study.

Network. To ensure that our experimental conclusions stem from the method itself rather than from any specialized architectural design, we deliberately adopt a minimalist DiT Peebles & Xie (2023) architecture, as shown in Fig. 2. The network takes a noisy motion sequence $(\boldsymbol{m}_t^1, \ldots, \boldsymbol{m}_t^L)$ as input and predicts a clean motion sequence $(\boldsymbol{m}_{\boldsymbol{\theta}}^1, \ldots, \boldsymbol{m}_{\boldsymbol{\theta}}^L)$. We introduce a differentiable $\mathrm{SO}(3)$ projection layer in the output head of the network, which guarantees that the rotation features produced by the network lie on the $\mathrm{SO}(3)$ manifold. Furthermore, to adapt our framework to the text-to-motion generation setting, we inject the conditioning signal through the cross-attention layers within the DiT. The detailed architecture is provided in the Appendix A.1.

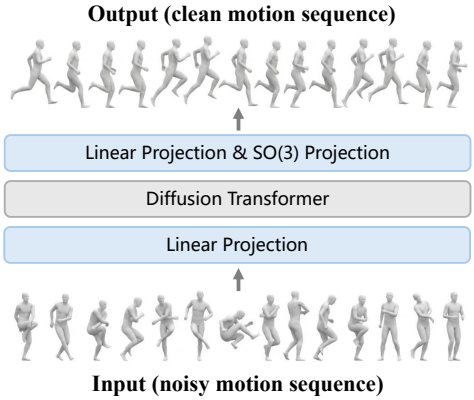


Figure 2: **The architecture of DirectMotion.** We use a minimalist DiT architecture to directly predict the clean motion sequence.

Training. The training procedure of DirectMotion follows the general flow matching paradigm Lipman et al. (2023). In addition, the generation condition is randomly dropped during training to enable classifier-free guidance (CFG) Ho & Salimans (2022). The detailed algorithm is provided in the Appendix A.1.

Inference. Our inference procedure also follows the flow matching paradigm Lipman et al. (2023). We first estimate the velocity field based on the predicted clean samples according to Eq. 4. Then, we apply the update rule shown in Eq. 5, performing numerical integration via the Euler method to obtain the generation results. We apply two universal inference techniques to enhance the model performance: CFG Ho & Salimans (2022) and timestep shifting Esser et al. (2024). A detailed discussion of the inference techniques is provided in the Appendix A.3.

5 Experiments

We train our model on public text-to-motion datasets. We first compare DirectMotion against state-of-the-art methods, then design ablation studies to investigate the three hypotheses proposed in the motivation section and assess the contributions of our design choices.

Datasets. We conduct experiments on the HumanML3D Guo et al. (2022) and SnapMoGen Guo et al. (2025) datasets. HumanML3D is a widely adopted text-to-motion benchmark comprising 28.6 hours of motion data and 44,970 text descriptions. Furthermore, SnapMoGen provides a larger-scale collection containing 43.7 hours of motion data alongside 122,565 textual annotations.

Evaluators & Metrics. We employ public evaluators to extract evaluation features. For the HumanML3D dataset, we employ a robust evaluator Meng et al. (2025b) that operates on a coordinate-based motion representation. We apply forward kinematics to convert our representation into the target representation. For the SnapMoGen dataset, we adopt the evaluator introduced in the original work Guo et al. (2025). This evaluator processes the same motion representation as our method.

Following previous works Guo et al. (2022; 2025), we employ established evaluation metrics including Fréchet Inception Distance (FID), R-precision (R@1, R@2, and R@3), and Multi-Modality (MM). Additionally, we report the Multimodal Distance (MM-Dist) and CLIP score on the HumanML3D and SnapMoGen datasets, respectively.

Implementation details. We train our models on HumanML3D and SnapMoGen for 200K and 300K steps, respectively, with a batch size of 512. The DiT backbone consists of 6 stacked DiT layers with an embedding dimension of 256, which results in a total of 9.2M parameters. In all experiments, the number of denoising steps is set to 8. Further implementation details are provided in the Appendix A.1. All rotational features are parameterized using the continuous 6D representation Zhou et al. (2019). For the HumanML3D dataset, we adopt exactly the same preprocessing protocol as the original work Guo et al. (2022). For the SnapMoGen dataset, we use the preprocessed copy released by the official implementation Guo et al. (2025).

Table 1: Quantitative evaluation results on the HumanML3D dataset. We repeat the evaluation 20 times and report the mean values with 95% confidence intervals.

Method	Params	FID ↓	R@1 ↑	R@2 ↑	R@3 ↑	MM Dist ↓	MM ↑
Discrete Latent-space Methods							
T2M-GPT Zhang et al. (2023a)	228+19.4M	$0.335^{\pm.003}$	$0.470^{\pm.003}$	$0.659^{\pm.002}$	$0.758^{\pm.002}$	$3.505^{\pm.017}$	$2.018^{\pm.053}$
MMM Pinyoanuntapong et al. (2024b)	236+18.0M	$0.132^{\pm.004}$	$0.487^{\pm.003}$	$0.683^{\pm.002}$	$0.782^{\pm.001}$	$3.359^{\pm.009}$	$1.241^{\pm.073}$
MoMask Guo et al. (2024)	25.4+19.4M	$0.116^{\pm.006}$	$0.490^{\pm.004}$	$0.687^{\pm.003}$	$0.786^{\pm.003}$	$3.353^{\pm.010}$	$1.263^{\pm.079}$
Continuous Latent-space Methods							
MLD Chen et al. (2023)	8.4+18.0M	$0.431^{\pm.014}$	$0.461^{\pm.004}$	$0.651^{\pm.004}$	$0.750^{\pm.003}$	$3.445^{\pm.019}$	$3.506^{\pm.031}$
MARDM Meng et al. (2025b)	291+18.8M	$0.114^{\pm.007}$	$0.500^{\pm.004}$	$0.695^{\pm.003}$	$0.795^{\pm.003}$	$3.270^{\pm.009}$	$2.231^{\pm.071}$
ACMDM Meng et al. (2025a)	596+17.1M	$0.058^{\pm.004}$	$0.522^{\pm.002}$	$0.713^{\pm.002}$	$0.807^{\pm.002}$	$3.205^{\pm.008}$	$2.077^{\pm.083}$
Raw Motion-space Methods							
MDM Tevet et al. (2022)	17.9M	$0.518^{\pm.032}$	$0.440^{\pm.007}$	$0.636^{\pm.006}$	$0.742^{\pm.004}$	$3.640^{\pm.028}$	$\mathbf{3.604}^{\pm.031}$
MotionDiffuse Zhang et al. (2024a)	87.2M	$0.778^{\pm.005}$	$0.450^{\pm.006}$	$0.641^{\pm.005}$	$0.753^{\pm.005}$	$3.490^{\pm.023}$	$3.179^{\pm.046}$
ReMoDiffuse Zhang et al. (2023b)	47.0M	$0.883^{\pm.003}$	$0.468^{\pm.003}$	$0.653^{\pm.002}$	$0.754^{\pm.002}$	$3.414^{\pm.020}$	$2.703^{\pm.154}$
DirectMotion	9.2M	$\mathbf{0.046}^{\pm.003}$	$\mathbf{0.548}^{\pm.003}$	$\mathbf{0.737}^{\pm.003}$	$\mathbf{0.830}^{\pm.002}$	$\mathbf{2.963}^{\pm.007}$	$1.254^{\pm.057}$

5.1 Comparison

Tables 1 and 2 report the comparison between our method and mainstream approaches. All baselines follow the same evaluation protocol and use the official settings Meng et al. (2025b); Guo et al. (2025). For latent-space methods, the parameters include both the generator and the autoencoder components, denoted as "generator+autoencoder". The parameter count excludes the text encoder. On HumanML3D, DirectMotion outperforms baselines, including advanced latent-based models that contain more parameters. On SnapMoGen, DirectMotion achieves performance comparable to that of state-of-the-art latent-based models and substantially outperforms all methods that operate directly in the raw motion space. These results indicate that latent-free architectures may not be inherently inferior to latent-based paradigms for motion generation tasks.

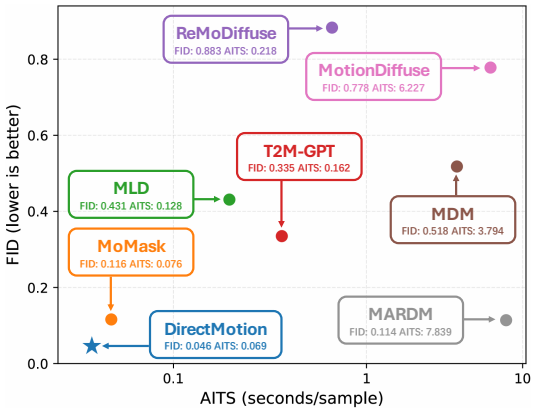


Figure 3: Comparison of motion quality and inference efficiency across different methods.

We present the motion quality and inference efficiency of various methods in Fig. 3. The average inference time per sample (AITS) is evaluated on the HumanML3D validation set using a single NVIDIA H100 GPU. The results demonstrate that DirectMotion not only achieves high-quality generation but also delivers very fast inference speed, positioning it favorably among competing approaches. This advantage stems from the minimalist design philosophy and the lightweight network architecture, which jointly reduce overhead while preserving expressive capacity.

Beyond comparison at the perceptual level, we further assess the physical properties of motion, including foot floating, foot sliding, mesh interpenetration, and jitter. The evaluation protocol follows Lin et al. (2025). These results are provided in the Appendix A.4.

5.2 Ablation Studies

We conduct comprehensive ablation studies on the SnapMoGen dataset. All experimental settings remain consistent with the comparison experiments, except for the ablated variables.

H1: toward a minimal representation. We incrementally introduce redundant features (joint position, joint velocity, and foot contact) into our motion representation and train the corresponding models. Since these features are Euclidean and **H1** does not involve manifold geometry, we construct the probability paths using only LERP and no longer use SLERP. Table 3 summarizes the results. We observe that redundant features tend to degrade model performance, and this degradation appears more pronounced as the redundant dimensions grows. This observation is consistent with **H1**.

Table 2: Quantitative evaluation results on the SnapMoGen dataset. We repeat the evaluation 20 times and report the mean values with 95% confidence intervals.

Method	Params	FID ↓	R@1 ↑	R@2 ↑	R@3 ↑	CLIP Score ↑	MM ↑
Discrete Latent-space Methods							
T2M-GPT Zhang et al. (2023a)	228+19.4M	$32.629^{\pm.087}$	$0.618^{\pm.002}$	$0.773^{\pm.002}$	$0.812^{\pm.002}$	$0.573^{\pm.001}$	$9.172^{\pm.181}$
MoMask Guo et al. (2024)	25.4+19.4M	$17.404^{\pm.051}$	$0.777^{\pm.002}$	$0.888^{\pm.002}$	$0.927^{\pm.002}$	$0.664^{\pm.001}$	$8.183^{\pm.184}$
MoMask++ Guo et al. (2025)	16.9+23.6M	$\mathbf{15.060}^{\pm.065}$	$\underline{0.802}^{\pm.001}$	$\underline{0.905}^{\pm.002}$	$\underline{0.938}^{\pm.001}$	$\underline{0.685}^{\pm.001}$	$7.259^{\pm.180}$
Continuous Latent-space Methods							
MARDM Meng et al. (2025b)	291+18.8M	$26.878^{\pm.131}$	$0.659^{\pm.002}$	$0.812^{\pm.002}$	$0.860^{\pm.002}$	$0.602^{\pm.001}$	$\underline{9.812}^{\pm.287}$
Raw Motion-space Methods							
MDM Tevet et al. (2022)	17.9M	$57.783^{\pm.092}$	$0.503^{\pm.002}$	$0.653^{\pm.002}$	$0.727^{\pm.002}$	$0.481^{\pm.001}$	$\mathbf{13.412}^{\pm.231}$
StableMoFusion Huang et al. (2024)	299M	$27.801^{\pm.063}$	$0.679^{\pm.002}$	$0.823^{\pm.002}$	$0.888^{\pm.002}$	$0.605^{\pm.001}$	$9.064^{\pm.138}$
DirectMotion	9.2M	$\underline{17.479}^{\pm.060}$	$\mathbf{0.819}^{\pm.003}$	$\mathbf{0.915}^{\pm.002}$	$\mathbf{0.947}^{\pm.002}$	$\mathbf{0.697}^{\pm.001}$	$6.625^{\pm.202}$

Table 3: Results of models trained with different amounts of feature redundancy.

Representation	Dim.	FID ↓	CLIP Score ↑
Root + Rotation	148	$\mathbf{19.458}^{\pm.072}$	$\mathbf{0.686}^{\pm.001}$
+ Joint Position	220	$\underline{21.415}^{\pm.074}$	$\underline{0.678}^{\pm.001}$
+ Joint Velocity	292	$23.748^{\pm.083}$	$0.670^{\pm.001}$
+ Foot Contact	296	$23.929^{\pm.115}$	$0.670^{\pm.001}$

Table 4: Results of models trained under different probability path and prediction targets.

Probability Path	Prediction	FID ↓	CLIP Score ↑
Linear	Velocity	$23.427^{\pm.096}$	$0.671^{\pm.001}$
Linear	Sample	$\underline{19.458}^{\pm.072}$	$\underline{0.686}^{\pm.001}$
$\mathrm{SO}(3)$	Velocity	$19.969^{\pm.080}$	$0.681^{\pm.001}$
$\mathrm{SO}(3)$	Sample	$\mathbf{17.479}^{\pm.060}$	$\mathbf{0.697}^{\pm.001}$

H2 & H3: modeling in the state space and predicting on the data manifold. We examine **H2** and **H3** jointly. Building on the minimal representation, we compare four configurations along two axes: the probability path of rotational features ($\mathrm{SO}(3)$ vs. linear) and the prediction target (velocity vs. sample). The prediction target affects only the supervision signal and does not change the form of the loss function. The results are reported in Tab. 4. By defining the probability path directly on the state space, we move the modeling domain from the redundant ambient space to the low-dimensional state space, which improves performance. This is in line with **H2**. Likewise, switching the prediction target from velocity to sample yields a further improvement, consistent with **H3**. We note that combining the two design choices yields the best results among the configurations we consider. These results converge on a consistent conclusion: respecting the low-redundancy structure inherent in motion data is the key to effective motion-space generation.

Rotation representation. The parameterization of $\mathrm{SO}(3)$ group elements is a critical design choice. Table 5 summarizes the evaluation results of models trained with different rotation formats. Consistent with prior findings Zhou et al. (2019); Brégier (2021), we observe that continuous rotation representations (rotation matrices and continuous 6D representations) achieve lower FID. Despite their generally higher dimensionality, continuous representations yield superior performance. This further demonstrates that DirectMotion effectively learns data that are intrinsically low-dimensional while residing in a high-dimensional ambient space.

Table 5: Evaluation results of models trained under different rotation representations.

Representation	Dim.	FID ↓	CLIP Score ↑
Matrix	220	$\underline{17.763}^{\pm.065}$	$\underline{0.690}^{\pm.001}$
Quaternion	100	$18.536^{\pm.054}$	$0.687^{\pm.001}$
Euler	76	$19.217^{\pm.077}$	$0.685^{\pm.001}$
Axis-angle	76	$18.321^{\pm.064}$	$0.693^{\pm.001}$
Continuous 6D	148	$\mathbf{17.479}^{\pm.060}$	$\mathbf{0.697}^{\pm.001}$

5.3 Motion prior in raw motion space

Previous work has shown that motion-space diffusion models can learn kinematic priors about human motion Shafir (2024). Such a prior enables a range of zero-shot applications, including long motion synthesis, trajectory control, and motion editing. Since DirectMotion also learns in the raw motion space, it may acquire the same prior. We examine this possibility qualitatively below.

We first examine whether such a prior exists through inpainting-based path control. As shown in the upper part of Fig. 4, we anchor the root-joint features that encode the motion trajectory and rely on the inpainting capability intrinsic to the flow model to generate the remaining body pose, then assess whether the resulting pose is consistent with the trajectory. The results show that the completed

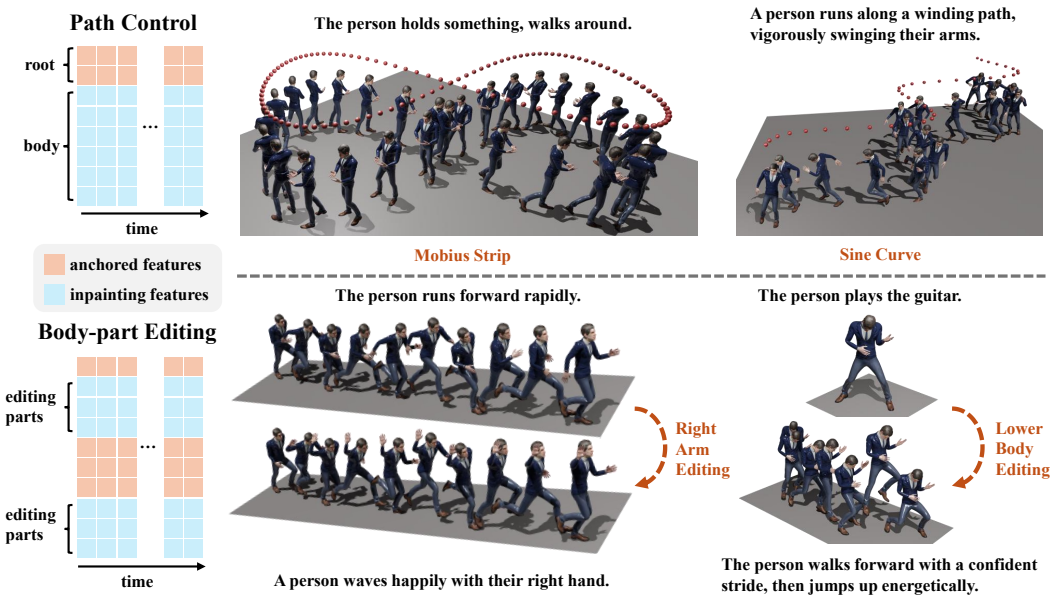


Figure 4: **Probing the motion prior via inpainting.** We test whether motion space training yields a motion prior by checking whether the model can regenerate the remaining features consistently with the anchored ones. **Top**: root features are anchored to a prescribed trajectory, where the generated pose follows the path. **Bottom:** re-noising arbitrary joints and regenerating them under text guidance gives joint-level edits consistent with the rest of the body, suggesting a fine-grained prior.

full-body poses match the path and contain rich details, such as the body leaning when turning. This suggests that DirectMotion may form a motion prior through learning in the raw motion space.

We further extend this inpainting approach to arbitrary joints, as shown in the lower half of Fig. 4. Given a motion, we re-noise some joints and regenerate them under text guidance, which yields a form of text-driven joint-level editing. The results show that the edited joints remain natural and stay consistent with the motion of the remaining joints, indicating that the motion prior captured by DirectMotion may be fine-grained enough to predict motion at the level of individual joints.

5.4 More helpful techniques & Visualization

DirectMotion is designed as a minimalist baseline. Following the design principle of simplicity and effectiveness, we keep the model complexity low and deliberately avoid additional training tricks. In engineering practice, however, such techniques are commonly introduced to improve performance. We therefore explore several techniques that prove helpful and report them in the Appendix A.5 for interested readers. In addition, we provide videos that visualize the generated motions in the media material, with further details given in the Appendix A.6.

6 Conclusion

We revisit text-driven human motion generation in raw motion space and identify feature redundancy and state redundancy as two key bottlenecks. Through DirectMotion, a minimalist baseline combining a compact representation, manifold-constrained probability paths, and clean-sequence prediction, we demonstrate that latent-free generation can match or exceed latent-space methods with a lightweight 9.2M-parameter DiT. Qualitative inpainting and editing results further suggest that DirectMotion learns a useful motion prior in raw motion space. We hope these findings encourage the community to reconsider latent-free paradigms as a simpler yet effective alternative for motion generation.

AI USE STATEMENT

In this work, we used generative AI tools to assist with translation. We also used generative AI tools to improve the readability of the manuscript and assist in creating conceptual diagrams (Fig. 1). The rest of the required disclosure tasks are not applicable to this work. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Hongjun An, Wenhan Hu, Sida Huang, Siqi Huang, Ruanjun Li, Yuanzhi Liang, Jiawei Shao, Yiliang Song, Zihan Wang, Cheng Yuan, et al. Ai flow: Perspectives, scenarios, and approaches. *Vicinagearth*, 3(1):1, 2026.

Yuxuan Bian, Ailing Zeng, Xuan Ju, Xian Liu, Zhaoyang Zhang, Wei Liu, and Qiang Xu. Motioncraft: Crafting whole-body motion with plug-and-play multimodal controls. In Toby Walsh, Julie Shah, and Zico Kolter (eds.), *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, pp. 1880–1888. AAAI Press, 2025. doi: 10.1609/AAAI.V39I2.32183. URL https://doi.org/10.1609/aaai.v39i2.32183.

Romain Brégier. Deep regression on manifolds: A 3d rotation case study. In *International Conference on 3D Vision, 3DV 2021, London, United Kingdom, December 1-3, 2021*, pp. 166–174. IEEE, 2021. doi: 10.1109/3DV53792.2021.00027. URL https://doi.org/10.1109/3DV53792.2021.00027.

Gunnar Carlsson. Topology and data. *Bulletin of the American mathematical society*, 46(2):255–308, 2009.

Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

Ricky T. Q. Chen and Yaron Lipman. Flow matching on general geometries. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=g7ohDlTITL.

Xiangyu Chen, Jixiang Luo, Yuankai Fan, Haibin Huang, Chi Zhang, and Xuelong Li. Generative transmission: Rethinking computation, bandwidth, and memory in communication, 2026a. URL https://arxiv.org/abs/2607.17482.

Xiangyu Chen, Jixiang Luo, Jingyu Xu, Fangqiu Yi, Chi Zhang, and Xuelong Li. Generative video compression: towards 0.01% compression rate for video transmission. *Vicinagearth*, 3 (1):7, 2026b.

Xin Chen, Biao Jiang, Wen Liu, Zilong Huang, Bin Fu, Tao Chen, and Gang Yu. Executing your commands via motion diffusion in latent space. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pp. 18000–18010. IEEE, 2023. doi: 10.1109/CVPR52729.2023.01726. URL https://doi.org/10.1109/CVPR52729.2023.01726.

Wenxun Dai, Ling-Hao Chen, Jingbo Wang, Jinpeng Liu, Bo Dai, and Yansong Tang. Motionlcm: Real-time controllable motion generation via latent consistency model. In Ales Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XVI*, Lecture Notes in Computer Science, pp. 390–408. Springer, 2024. doi: 10.1007/978-3-031-72640-8_22. URL https://doi.org/10.1007/978-3-031-72640-8_22.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In

Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12606–12633. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/esser24a.html.

Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 5142–5151. IEEE, 2022. doi: 10.1109/CVPR52688.2022.00509. URL https://doi.org/10.1109/CVPR52688.2022.00509.

Chuan Guo, Yuxuan Mu, Muhammad Gohar Javed, Sen Wang, and Li Cheng. Momask: Generative masked modeling of 3d human motions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 1900–1910. IEEE, 2024. doi: 10.1109/CVPR52733.2024.00186. URL https://doi.org/10.1109/CVPR52733.2024.00186.

Chuan Guo, Inwoo Hwang, Jian Wang, and Bing Zhou. Snapmogen: Human motion generation from expressive texts. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diago, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/9028b8a3ca98f58e373f0c1497a17448-Abstract-Conference.html.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 15979–15988. IEEE, 2022. doi: 10.1109/CVPR52688.2022.01553. URL https://doi.org/10.1109/CVPR52688.2022.01553.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *CoRR*, abs/2207.12598, 2022. doi: 10.48550/ARXIV.2207.12598. URL https://doi.org/10.48550/arXiv.2207.12598.

Yiheng Huang, Hui Yang, Chuanchen Luo, Yuxi Wang, Shibiao Xu, Zhaoxiang Zhang, Man Zhang, and Junran Peng. Stablemofusion: Towards robust and efficient diffusion-based motion generation framework. In Jianfei Cai, Mohan S. Kankanhalli, Balakrishnan Prabhakaran, Susanne Boll, Ramanathan Subramanian, Liang Zheng, Vivek K. Singh, Pablo César, Lexing Xie, and Dong Xu (eds.), *Proceedings of the 32nd ACM International Conference on Multimedia, MM 2024, Melbourne, VIC, Australia, 28 October 2024 - 1 November 2024*, pp. 224–232. ACM, 2024. doi: 10.1145/3664647.3681657. URL https://doi.org/10.1145/3664647.3681657.

Tencent Hunyuan. Hy-motion 1.0: Scaling flow matching models for text-to-motion generation. *CoRR*, abs/2512.23464, 2025. doi: 10.48550/ARXIV.2512.23464. URL https://doi.org/10.48550/arXiv.2512.23464.

Minjae Jeong, Yechan Hwang, Jaejin Lee, Sungyoon Jung, and Won Hwa Kim. Hgm^3: Hierarchical generative masked motion modeling with hard token mining. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=IEul1M5pyk.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6114.

Heng Li, Xiaotong Lin, Ling-An Zeng, Yulei Kang, Shuai Li, and Jian-Fang Hu. Motionhiflow: Text-to-motion via hierarchical flow matching. *CoRR*, abs/2604.23264, 2026. doi: 10.48550/ARXIV.2604.23264. URL https://doi.org/10.48550/arXiv.2604.23264.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. *CoRR*, abs/2511.13720, 2025. doi: 10.48550/ARXIV.2511.13720. URL https://doi.org/10.48550/arXiv.2511.13720.

Jing Lin, Ruisi Wang, Junzhe Lu, Ziqi Huang, Guorui Song, Ailing Zeng, Xian Liu, Chen Wei, Wanqi Yin, Qingping Sun, Zhongang Cai, Lei Yang, and Ziwei Liu. The quest for generalizable motion generation: Data, model, and evaluation. *CoRR*, abs/2510.26794, 2025. doi: 10.48550/ARXIV.2510.26794. URL https://doi.org/10.48550/arXiv.2510.26794.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Shunlin Lu, Jingbo Wang, Zeyu Lu, Ling-Hao Chen, Wenxun Dai, Junting Dong, Zhiyang Dou, Bo Dai, and Ruimao Zhang. Scamo: Exploring the scaling law in autoregressive motion generation model. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 27872–27882. Computer Vision Foundation / IEEE, 2025. doi: 10.1109/CVPR52734.2025.02595. URL https://openaccess.thecvf.com/content/CVPR2025/html/Lu_ScaMo_Exploring_the_Scaling_Law_in_Autoregressive_Motion_Generation_Model_CVPR_2025_paper.html.

Yiyang Lu, Susie Lu, Qiao Sun, Hanhong Zhao, Zhicheng Jiang, Xianbang Wang, Tianhong Li, Zhengyang Geng, and Kaiming He. One-step latent-free image generation with pixel mean flows. *CoRR*, abs/2601.22158, 2026. doi: 10.48550/ARXIV.2601.22158. URL https://doi.org/10.48550/arXiv.2601.22158.

Zichong Meng, Zeyu Han, Xiaogang Peng, Yiming Xie, and Huaizu Jiang. Absolute coordinates make motion generation easy. *CoRR*, abs/2505.19377, 2025a. doi: 10.48550/ARXIV.2505.19377. URL https://doi.org/10.48550/arXiv.2505.19377.

Zichong Meng, Yiming Xie, Xiaogang Peng, Zeyu Han, and Huaizu Jiang. Rethinking diffusion for text-driven human motion generation: Redundant representations, evaluation, and masked autoregression. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 27859–27871. Computer Vision Foundation / IEEE, 2025b. doi: 10.1109/CVPR52734.2025.02594. URL https://openaccess.thecvf.com/content/CVPR2025/html/Meng_Rethinking_Diffusion_for_Text-Driven_Human_Motion_Generation_Redundant_Representations_Evaluation_CVPR_2025_paper.html.

Mirgahney Mohamed, Harry Jake Cunningham, Marc Peter Deisenroth, and Lourdes Agapito. Recmodiffuse: Recurrent flow diffusion for human motion generation. *CoRR*, abs/2406.07169, 2024. doi: 10.48550/ARXIV.2406.07169. URL https://doi.org/10.48550/arXiv.2406.07169.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pp. 4172–4182. IEEE, 2023. doi: 10.1109/ICCV51070.2023.00387. URL https://doi.org/10.1109/ICCV51070.2023.00387.

Ekkasit Pinyoanuntapong, Muhammad Usama Saleem, Pu Wang, Minwoo Lee, Srijan Das, and Chen Chen. BAMM: bidirectional autoregressive motion model. In Ales Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XV*, Lecture Notes in Computer Science, pp. 172–190. Springer, 2024a. doi: 10.1007/978-3-031-72633-0_10. URL https://doi.org/10.1007/978-3-031-72633-0_10.

Ekkasit Pinyoanuntapong, Pu Wang, Minwoo Lee, and Chen Chen. MMM: generative masked motion model. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 1546–1555. IEEE, 2024b. doi: 10.1109/CVPR52733.2024.00153. URL https://doi.org/10.1109/CVPR52733.2024.00153.

Ekkasit Pinyoanuntapong, Muhammad Usama Saleem, Korrawe Karunratanakul, Pu Wang, Hongfei Xue, Chen Chen, Chuan Guo, Junli Cao, Jian Ren, and Sergey Tulyakov. Maskcontrol: Spatio-temporal control for masked motion synthesis. In *IEEE/CVF International Conference on Computer Vision, ICCV 2025, Honolulu, HI, USA, October 19-25, 2025*, pp. 9955–9965. IEEE, 2025. doi: 10.1109/ICCV51701.2025.00928. URL https://doi.org/10.1109/ICCV51701.2025.00928.

Davis Rempe, Mathis Petrovich, Ye Yuan, Haotian Zhang, Xue Bin Peng, Yifeng Jiang, Tingwu Wang, Umar Iqbal, David Minor, Michael de Ruyter, Jiefeng Li, Chen Tessler, Edy Lim, Eugene Jeong, Sam Wu, Ehsan Hassani, Michael Huang, Jin-Bey Yu, Chaeyeon Chung, Lina Song, Olivier Dionne, Jan Kautz, Simon Yuen, and Sanja Fidler. Kimodo: Scaling controllable human motion generation. *CoRR*, abs/2603.15546, 2026. doi: 10.48550/ARXIV.2603.15546. URL https://doi.org/10.48550/arXiv.2603.15546.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 10674–10685. IEEE, 2022. doi: 10.1109/CVPR52688.2022.01042. URL https://doi.org/10.1109/CVPR52688.2022.01042.

Yoni Shafir, Guy Tevet, Roy Kapon, and Amit Haim Bermano. Human motion diffusion as a generative prior. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=dTpbEdN9kr.

Jiawei Shao and Xuelong Li. Ai flow at the network edge. *IEEE Network*, 2025.

Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H. Bermano. Human motion diffusion model. *CoRR*, abs/2209.14916, 2022. doi: 10.48550/ARXIV.2209.14916. URL https://doi.org/10.48550/arXiv.2209.14916.

Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 6306–6315, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/7a98af17e63a0ac09ce2e96d03992fbc-Abstract.html.

Qi Wu, Yubo Zhao, Yifan Wang, Xinhang Liu, Yu-Wing Tai, and Chi-Keung Tang. Motion-agent: A conversational framework for human motion generation with llms. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=AvOhBgsE5R.

Dianbing Xi, Zhaoqi Zhou, Youming Wang, Jiepeng Wang, and Chi Zhang. A survey: advances in multi-modal visual understanding and generation. *Vicinagearth*, 3(1):8, 2026.

Lixing Xiao, Shunlin Lu, Huaijin Pi, Ke Fan, Liang Pan, Yueer Zhou, Ziyong Feng, Xiaowei Zhou, Sida Peng, and Jingbo Wang. Motionstreamer: Streaming motion generation via diffusion-based autoregressive model in causal latent space. In *IEEE/CVF International Conference on Computer Vision, ICCV 2025, Honolulu, HI, USA, October 19-25, 2025*, pp. 10086–10096. IEEE, 2025. doi: 10.1109/ICCV51701.2025.00940. URL https://doi.org/10.1109/ICCV51701.2025.00940.

Ye Yuan, Jiaming Song, Umar Iqbal, Arash Vahdat, and Jan Kautz. Physdiff: Physics-guided human motion diffusion model. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pp. 15964–15975. IEEE, 2023. doi: 10.1109/ICCV51070.2023.01467. URL https://doi.org/10.1109/ICCV51070.2023.01467.

Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Shaoli Huang, Yong Zhang, Hongwei Zhao, Hongtao Lu, and Xi Shen. T2M-GPT: generating human motion from textual descriptions with discrete representations. *CoRR*, abs/2301.06052, 2023a. doi: 10.48550/ARXIV.2301.06052. URL https://doi.org/10.48550/arXiv.2301.06052.

Mingyuan Zhang, Xinying Guo, Liang Pan, Zhongang Cai, Fangzhou Hong, Huirong Li, Lei Yang, and Ziwei Liu. Remodiffuse: Retrieval-augmented motion diffusion model. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pp. 364–373. IEEE, 2023b. doi: 10.1109/ICCV51070.2023.00040. URL https://doi.org/10.1109/ICCV51070.2023.00040.

Mingyuan Zhang, Huirong Li, Zhongang Cai, Jiawei Ren, Lei Yang, and Ziwei Liu. Finemogen: Fine-grained spatio-temporal motion generation and editing. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023c. URL http://papers.nips.cc/paper_files/paper/2023/hash/2d52879ef2ba487445ca2e143b104c3b-Abstract-Conference.html.

Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. Motiondiffuse: Text-driven human motion generation with diffusion model. *IEEE Trans. Pattern Anal. Mach. Intell.*, 46(6):4115–4128, 2024a. doi: 10.1109/TPAMI.2024.3355414. URL https://doi.org/10.1109/TPAMI.2024.3355414.

Yaqi Zhang, Di Huang, Bin Liu, Shixiang Tang, Yan Lu, Lu Chen, Lei Bai, Qi Chu, Nenghai Yu, and Wanli Ouyang. Motiongpt: Finetuned llms are general-purpose motion generators. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan (eds.), *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pp. 7368–7376. AAAI Press, 2024b. doi: 10.1609/AAAI.V38I7.28567. URL https://doi.org/10.1609/aaai.v38i7.28567.

Zeyu Zhang, Yiran Wang, Danning Li, Dong Gong, Ian Reid, and Richard Hartley. Flashmo: Geometric interpolants and frequency-aware sparsity for scalable efficient motion generation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 5745–5753. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00589. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Zhou_On_the_Continuity_of_Rotation_Representations_in_Neural_Networks_CVPR_2019_paper.html.

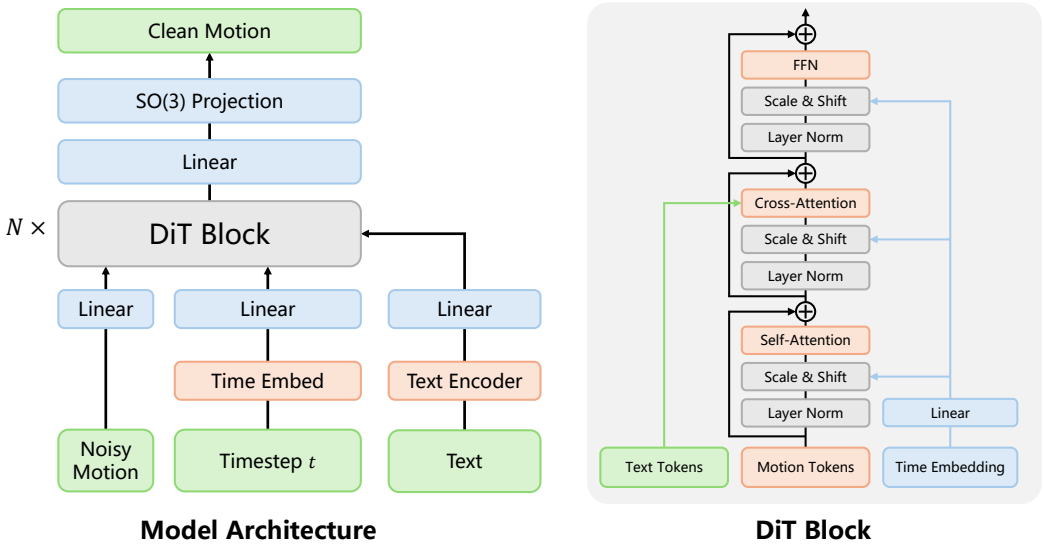


Figure 5: **Model architecture of DirectMotion.** The overall network architecture is shown on the left, and the detailed structure of each DiT block is shown on the right.

A Appendix

A.1 Implementation Details

Model Architecture. We implement DirectMotion based on the general DiT Peebles & Xie (2023) architecture. The overall network structure is shown on the left side of Figure 5. Our network injects the timestep information through adaptive layer normalization (AdaLN) and incorporates the textual condition through a cross-attention mechanism. This design constitutes a general text-conditioned diffusion framework, which ensures that our results stem from the modeling approach rather than from specialized network architectures.

$\mathrm{SO}(3)$ **projection.** Our modeling approach requires the network to output joint rotation features that lie on the SO(3) manifold. We therefore apply an SO(3) projection to the network outputs to map these features onto rotation matrices. We apply the SO(3) projection only to the joint rotation features, while the Euclidean features (root features) remain unchanged.

The operation performed by the SO(3) projection varies across different rotation representations. For the continuous 6D representation Zhou et al. (2019), we apply the Gram-Schmidt orthogonalization procedure to obtain the corresponding rotation matrix. The projection algorithms we adopt for other rotation representations follow Brégier (2021).

Training. The training procedure of DirectMotion follows the general flow matching paradigm Lipman et al. (2023). Overall, our method differs from standard flow matching only in the noise distribution, the perturbation process, and the loss function. A single training step of DirectMotion is summarized in Algorithm 1. For clarity of exposition, Algorithm 1 describes the procedure for a single motion representation $\boldsymbol{m}$. In practice, the algorithm is generalized to operate on the entire motion sequence $(\boldsymbol{m}^1, \ldots, \boldsymbol{m}^L)$.

Classifier-free guidance. We apply CFG Ho & Salimans (2022) to enhance the adherence of the model to the conditioning. However, the predicted clean samples incorporate rotational features, and direct linear extrapolation of these features would cause them to deviate from the $\mathrm{SO}(3)$ manifold. Consequently, we perform the linear extrapolation within the velocity space:

$$\boldsymbol{u} = \boldsymbol{u}_{\text{uncond}} + s\left(\boldsymbol{u}_{\text{cond}} - \boldsymbol{u}_{\text{uncond}}\right), \tag{7}$$

where u_{cond} and u_{uncond} are the conditional and unconditional rotational velocities, respectively, and s is the guidance scale. The rotational velocities lie in the Lie algebra $\mathfrak{so}(3)$, which is a linear space, and this ensures that the linear extrapolation remains valid.

Algorithm 1 A single training step of DirectMotion

Input: clean motion $\boldsymbol{m} = (\boldsymbol{x}, \boldsymbol{r}^1, ..., \boldsymbol{r}^J)$, condition $\boldsymbol{c}$, network $\text{net}_{\boldsymbol{\theta}}$

Output: updated network parameters $\hat{\boldsymbol{\theta}}$

1: Randomly drop the condition $\boldsymbol{c}$ for CFG
2: Sample a timestep $t \sim \mathcal{U}[0, 1]$
3: Sample noises $\left(\boldsymbol{\epsilon}, \boldsymbol{\delta}^1, ..., \boldsymbol{\delta}^J\right)$, where $\boldsymbol{\epsilon} \sim p_{\text{noise}}(\boldsymbol{\epsilon})$ and $\boldsymbol{\delta}^1, \ldots, \boldsymbol{\delta}^J \sim p_{\text{noise}}(\boldsymbol{\delta})$
4: Compute the noisy motion $\boldsymbol{m}_t = (\boldsymbol{x}_t, \boldsymbol{r}_t^1, ..., \boldsymbol{r}_t^J)$ following Eq. 2
5: Predict the clean motion $\boldsymbol{m}_{\boldsymbol{\theta}} = \left(\boldsymbol{x}_{\boldsymbol{\theta}}, \boldsymbol{r}_{\boldsymbol{\theta}}^1, ..., \boldsymbol{r}_{\boldsymbol{\theta}}^J\right) = \text{net}_{\boldsymbol{\theta}}(\boldsymbol{m}_t, \boldsymbol{c}, t)$
6: Compute the loss $\mathcal{L}_{\boldsymbol{\theta}}$ following Eq. 6
7: Update parameters: $\hat{\boldsymbol{\theta}} = \text{optimizer}\left(\mathcal{L}_{\boldsymbol{\theta}}\right)$
8: **return** $\hat{\boldsymbol{\theta}}$

Timestep shifting. A study on image generation Rombach et al. (2022) shows that allocating more inference steps to the high-noise stages improves model performance. Therefore, we introduce timestep shifting to adjust the timesteps during inference:

$$\hat{t} = \frac{\alpha t}{1 + (\alpha - 1)t}, \tag{8}$$

where $\hat{t}$ is the timestep fed into the model; the shifting scale $\alpha > 0$ is a hyperparameter; t is sampled at equal intervals over $[0, 1]$. When $\alpha < 1$, the inference steps concentrate on the high-noise side, whereas for $\alpha > 1$, they concentrate on the low-noise side. When $\alpha = 1$, the timestep shifting reduces to the uniform schedule.

Hyperparameters. The hyperparameters used in our experiments are listed in Table 6. Among them, "λ" is a hyperparameter that balances the Euclidean feature loss and the rotational feature loss, as introduced in Eq.6 of the main paper. In addition, the "timestep sampler" is used to randomly sample timesteps during training, and we sample directly from a uniform distribution over $[0, 1]$.

Table 6: Hyperparameters used in experiments.

Hyperparameter	HumanML3D	SnapMoGen
Model		
token dim	256	
attention heads	4	
DiT blocks	6	
activation	SiLU	
QK-norm	RMSNorm	
FFN ratio	4.0	
FFN activation	SwiGLU	
Text Encoder	T5-base	
Training		
training steps	200K	300K
batch size	512	
condition drop	0.1	
LR	2e-4	
LR schedule	constant	
optimizer	AdamW	
weight decay	0.0	
dropout	0.0	
Inference		
CFG scale	4.6	3.6
timestep shifting scale	0.5	0.3
sampling steps	8	
Other		
λ	0.1	0.05
timestep sampler	uniform on [0, 1]	

A.2 Geodesic Loss VS. L2 Loss

For the rotation features, we adopt a loss function that directly computes the L2 loss on the rotation matrices, as shown in Eq. 6 in the main paper. While geodesic distance serves as an intuitive loss function for rotation from the perspective of manifold geometry, the $\text{SO}(3)$ manifold represents a special case. Consider two rotation matrices R_1 and R_2, whose geodesic distance $d(R_1, R_2)$ is defined as:

$$d(R_1, R_2)_{\text{geo}} = \|\log(R_1^\top R_2)\|_2 = \arccos\left(\frac{\text{tr}(R_1^\top R_2) - 1}{2}\right) = \theta, \tag{9}$$

where $\theta \in [0, \pi]$ is the rotation angle of the relative rotation $R_1^\top R_2$. On the other hand, it can be proved that the L2 distance between R_1 and R_2 satisfies:

$$\|R_1 - R_2\|_2 = 2\sqrt{2}\sin(\theta/2). \tag{10}$$

Since $\theta \in [0, \pi]$, the L2 distance $\|R_1 - R_2\|_2$ is strictly monotonically increasing with respect to the geodesic distance $d_{\text{geo}}(R_1, R_2)$. The two distance measures exhibit consistent error ordering for individual rotation pairs, and both attain their minimum when $R_1 = R_2$. Notably, when the error is small, the L2 distance linearly approximates the geodesic distance. Therefore, on the $\text{SO}(3)$ manifold, the L2 distance serves as a viable surrogate for the geodesic distance.

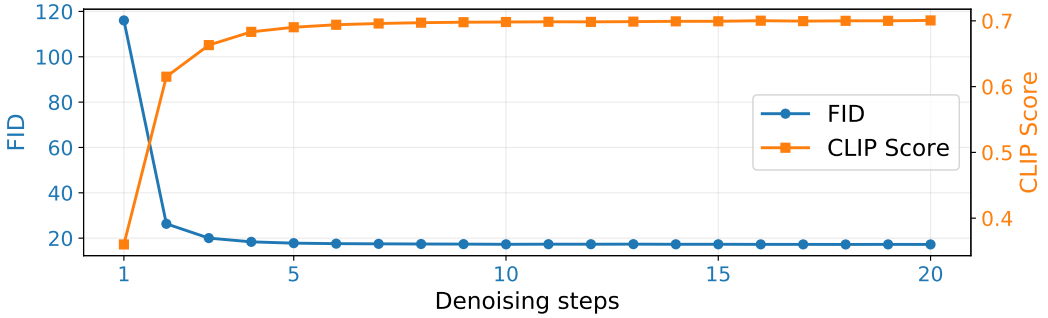


Figure 7: Impact of denoising steps on performance on the SnapMoGen dataset.

We use L2 distance rather than geodesic distance because it offers better numerical stability. As shown in Eq 9, computing the geodesic loss requires inverse trigonometric operations. Such computations are costly and prone to numerical instability. For example, inverse sine and cosine functions return NaN when their inputs fall outside $[-1, 1]$. In contrast, the L2 loss avoids these issues.

Table 7: Comparison of models trained with different loss functions.

Loss	FID↓	CLIP Score↑
Geodesic	17.354	0.696
L2	17.479	0.697

We also empirically compare models trained with the L2 loss and the geodesic loss, as shown in Tab. 7. For the geodesic loss, we use the code provided by prior work Brégier (2021), which includes appropriate safeguards against numerical instability. The results show no notable difference in performance between the two losses.

A.3 Analysis of Inference Hyperparameters

CFG & Timestep shifting. Figure 6 illustrates the effects of CFG guidance scale and timestep shifting scale on model performance. The improvement brought by CFG falls within the empirical range observed in conventional motion generation models and exhibits no distinctive characteristics. The timestep scale, however, deserves particular attention: DirectMotion achieves better performance at smaller scales (< 1). This indicates that the main difficulty of generation is concentrated in high-noise regions, whereas low-noise regions can be traversed with only a few steps. This characteristic may also account for why directmotion requires relatively few denoising steps.

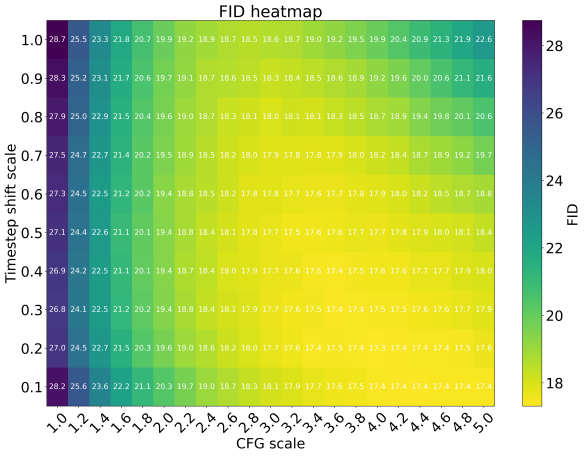


Figure 6: Effects of the classifier-free guidance scale and timestep shifting scale on performance.

Denoising Steps. Typically, denoising models operating in raw motion space require hundreds or even thousands of steps to achieve satisfactory performance Tevet et al. (2022); Zhang et al. (2024a). In contrast, DirectMotion requires significantly fewer denoising steps. Figure 7 presents the performance metrics of DirectMotion with different numbers of denoising steps. The results show that DirectMotion achieves strong performance after 5 steps of Euler iteration.

A.4 Evaluation on Physical Properties

In addition to using Fréchet Inception Distance (FID) to assess the overall quality of generated motions, we follow the evaluation protocol proposed by Lin et al. (2025) to evaluate their individual

Table 8: Evaluation results of DirectMotion on physical-plausibility metrics.

Method	Jitter Degree ↓	Foot Floating ↓	Foot Sliding ↓	Body Penetration ↓
MDM Tevet et al. (2022)	0.0136	0.156	0.0136	1.68
T2M-GPT Zhang et al. (2023a)	0.0156	0.209	0.0156	1.33
FineMoGen Zhang et al. (2023c)	0.0118	0.281	0.0091	<u>1.18</u>
MotionLCM Dai et al. (2024)	0.0218	0.193	0.0202	1.73
MoMask Guo et al. (2024)	0.0147	0.178	0.0147	1.48
MotionDiffuse Zhang et al. (2024a)	0.0111	<u>0.126</u>	**0.0063**	1.35
MotionCraft Bian et al. (2025)	0.0132	0.402	0.0090	**1.15**
ViMoGen Lin et al. (2025)	<u>0.0108</u>	0.204	<u>0.0064</u>	1.78
DirectMotion	**0.0101**	**0.116**	0.0073	1.47

Table 9: Evaluation results of DirectMotion with more complexity on SnapMoGen dataset.

Method	Size	Attention	Text Encoder	EMA	FID ↓	CLIP Score ↑
Baseline	9.2M	cross-attention	T5-base	No	17.479	0.697
More Parameters	27.21M				16.243	0.704
Full Attention		full-attention			16.647	0.702
Better Text Encoder			T5-large		17.089	0.701
EMA				Yes	17.115	0.699

physical properties. While FID measures distributional similarity, the physical property evaluation provides an assessment of physical plausibility, thereby yielding a more comprehensive evaluation of motion quality. The evaluation metrics include (1) jitter degree, which measures instability in acceleration; (2) foot floating, which detects unnatural foot elevation above the ground; (3) foot sliding, which measures foot displacement during ground contact; and (4) body penetration, which measures the frequency of penetration events.

Table 8 presents the evaluation results. DirectMotion achieves the best performance on jitter degree and foot floating, while remaining competitive on the other metrics, demonstrating consistently strong physical plausibility across multiple dimensions.

A.5 More Helpful Techniques

DirectMotion is designed as a minimalist baseline. In accordance with the design philosophy of simplicity and effectiveness, we constrain the model complexity and deliberately refrain from using additional training tricks. In engineering practice, however, it is common to introduce additional techniques to improve performance. We therefore explore several schemes that can improve model performance and briefly report them in Tab. 9. In this discussion, baseline refers to the minimalist DirectMotion introduced in the main paper.

More parameters. We scale the DiT backbone of DirectMotion from 9.2M parameters to a size comparable to that of common text-to-motion methods, using 8 layers, 6 attention heads, and an embedding dimension of 384, for a total of 27.21M parameters. The increased number of trainable parameters further improves performance.

Full attention mechanism. We use cross-attention to inject text conditions, where the text features are not allowed to attend to either the motion features or themselves. This design avoids additional computational overhead and thus accelerates both training and inference. We also explore a full-attention scheme that concatenates the text and motion features into a single long sequence and applies self-attention throughout. This design effectively improves model performance.

A better text encoder. We use T5-base, a text encoder commonly adopted in text-to-motion models. We evaluate the performance of DirectMotion with a stronger text encoder. The results indicate that a stronger text encoder leads to improved performance.

Exponential moving average. For flow models, applying an exponential moving average (EMA) to the model weights during training is a common practice. We incorporate EMA into the training procedure. The evaluation results demonstrate that EMA remains effective for our method.

A.6 VISUALIZATION

The motion visualizations in the main paper (Fig. 4) are rendered with Blender, retargeting the motions onto a character model via an automatic rigging plugin (Rokoko), without any manual modification to the generated motions.

In the media materials, we provide videos corresponding to Figure 4 in the main paper. Additionally, we include additional generated samples from our model. The prompts are randomly sampled from the SnapMoGen test set without manual selection.