RIEMANNIAN MOTION GENERATION

A UNIFIED FRAMEWORK FOR HUMAN MOTION REPRESENTATION AND GENERATION VIA RIEMANNIAN FLOW MATCHING

Anonymous authors
Paper under double-blind review



Figure 1: Text-to-motion samples under our **Riemannian Motion Generation** framework.

ABSTRACT

Human motion generation is often learned in Euclidean spaces, although valid motions follow structured non-Euclidean geometry. We present **Riemannian Motion Generation (RMG)**, a unified framework that represents motion on a product manifold and learns dynamics via Riemannian flow matching. RMG factorizes motion into several manifold factors, yielding a scale-free representation with intrinsic normalization, and uses geodesic interpolation, tangent-space supervision, and manifold-preserving ODE integration for training and sampling. On HumanML3D, RMG achieves state-of-the-art FID in the HumanML3D format (0.043) and ranks first on all reported metrics under the MotionStreamer format. On MotionMillion, it also surpasses strong baselines (FID 5.6, R@1 0.86). Ablations show that the compact $\mathcal{T} + \mathcal{R}$ (translation + rotations) representation is the most stable and effective, highlighting geometry-aware modeling as a practical and scalable route to high-fidelity motion generation.

1 INTRODUCTION

Conditional human motion generation has emerged as a key challenge in generative modeling, incorporating conditioning signals that range from text descriptions and action labels to audio, music, and scene context (Zhu et al., 2024). Synthesizing high-fidelity motion sequences is essential for advancements in human-computer interaction, embodied AI, and augmented-reality content creation.

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.

Table 1: Comparison of motion representations. $\mathcal{T}$: global translation; $\mathcal{R}_{\text{Orientation}}$: global orientation; $\mathcal{R}_{\text{Joint}}$: per-joint rotation; $\mathcal{P}$: joint position; $\mathrm{d}\cdot$: temporal difference of the variable. Compared to existing formats, our method adopts the most concise format, with each factor represented on its natural manifold.

Format	$\mathcal{T}$	$\mathcal{R}_{\text{Orientation}}$	$\mathcal{R}_{\text{Joint}}$	$\mathcal{P}$	$\mathrm{d}\mathcal{T}$	$\mathrm{d}\mathcal{R}_{\text{Orientation}}$	$\mathrm{d}\mathcal{R}_{\text{Joint}}$	$\mathrm{d}\mathcal{P}$	Foot Contact	Ambient Dimension
HumanML3D	✗	✗	✓	✓	✓	✓	✗	✓	✓	$12J-1$
MotionStreamer	✗	✗	✓	✓	✓	✓	✗	✓	✗	$12J+8$
DART	✓	✓	✓	✓	✓	✓	✗	✓	✗	$12J+12$
HY-Motion	✓	✓	✓	✓	✗	✗	✗	✗	✗	$9J+3$
RMG (Ours)	✓	✓	✓	✗	✗	✗	✗	✗	✗	$4J+3$

Recent progress in human motion generation has primarily focused on model architecture. While earlier methods relied on VAEs (e.g., TEMOS (Petrovich et al., 2022) and T2M (Guo et al., 2022)), state-of-the-art systems increasingly adopt diffusion or autoregressive frameworks (Zhang et al., 2024a; Tevet et al., 2022; Xiao et al., 2025; Guo et al., 2025; Kim et al., 2023; Guo et al., 2024; Zhang et al., 2023; Jiang et al., 2023). By contrast, the geometry of motion representation has received less systematic attention, despite its direct impact on optimization difficulty, sampling stability, and physical plausibility.

Most existing pipelines still encode motion using redundant Euclidean coordinates, enforcing validity only implicitly through constraints or post-processing. For a skeleton with J joints, an articulated pose has intrinsic degrees of freedom on the order of $3J$; however, common encodings concatenate multiple correlated views of the same state, occupying a much higher-dimensional ambient space. Table 1 summarizes these encoding strategies across previous methods. Consequently, models are trained in $\mathbb{R}^D$, whereas physically valid motions reside on, or near, a lower-dimensional manifold with intrinsic dimension d.

This observation suggests a simple but important motivation: if data already resides on a lower-dimensional manifold and follows a strong geometric structure, the generative models should benefit from operating in that space and respect that structure rather than in an unconstrained ambient vector space. Our key insight is that much of the apparent complexity of human motion does not come from arbitrary high-dimensional variation, but from composing several low-dimensional factors, each with its own natural geometry. Once this factorization is made explicit, geometric consistency no longer needs to be imposed only after generation. Instead, it can be built directly into both the representation and the generative dynamics. This viewpoint naturally leads to a geometry-aware formulation in which representation and model design are developed together, rather than treated as separate concerns.

Guided by this perspective, we first provide a unified geometric view that decomposes existing representations into common factors and their natural manifolds. Building on this view, we introduce **Riemannian Motion Generation (RMG)**, a representation-and-generation framework that models motion on a product manifold and learns dynamics via Riemannian flow matching. RMG yields a compact manifold-aware parameterization and a geometry-consistent training/inference pipeline. On HumanML3D text-to-motion benchmarks, it matches or exceeds strong baselines across quality, alignment, and diversity metrics. Moreover, RMG surpasses all baselines on the large-scale MotionMillion dataset, demonstrating superior scalability and generalization.

We summarize our core contributions as follows:

- We propose **Riemannian Motion Generation (RMG)**, a geometric paradigm for human motion generation that models motion on product manifolds and learns dynamics via Riemannian flow matching.
- We design a compact Riemannian representation that is both effective and efficient, and we provide a systematic evaluation of representation geometry in the context of human motion generation.

- To the best of our knowledge, we present the first demonstration that Riemannian flow matching scales effectively to large datasets and modern high-capacity generative architectures.
- We demonstrate strong empirical results on text-to-motion benchmarks, showing consistent gains across quality, alignment, and diversity metrics.

2 Related Works

2.1 Human Motion Generation

Human motion generation has witnessed rapid progress, propelled by improvements in deep learning and motion capture technologies. Early approaches primarily adopted regression models to map predefined action labels to motion sequences, exemplified by works such as Action2Motion(Guo et al., 2020) and SA-GAN (Yu et al., 2020). The introduction of advanced generative models—including Variational Autoencoders (Petrovich et al., 2021; Kingma et al., 2013), Generative Adversarial Networks (Degardin et al., 2022; Goodfellow et al., 2014), normalizing flows (Rezende & Mohamed, 2015; Valle-Pérez et al., 2021), and more recently, diffusion models (Tevet et al., 2022; Petrovich et al., 2021; Chen et al., 2023; Tseng et al., 2023; Kim et al., 2023; Dabral et al., 2023) have enabled the modeling of complex, multi-modal motion distributions and the synthesis of more realistic and diverse human motions.

A significant trend in recent years is the use of conditional generative models, where motion is synthesized based on various forms of contextual signals (Petrovich et al., 2022; Kim et al., 2022; Tevet et al., 2022; Guo et al., 2020; Petrovich et al., 2022; Zhang et al., 2023; Li et al., 2021; Ao et al., 2022). These methods have evolved from simple mappings to architectures that exploit joint embeddings, transformers(Petrovich et al., 2022; Guo et al., 2020), and diffusion processes (Tevet et al., 2022; Tseng et al., 2023; Dabral et al., 2023; Chen et al., 2023), resulting in higher fidelity and controllable motion sequences.

Despite these advances, human motion generation remains challenging due to the highly articulated and nonlinear nature of human movement, as well as the need for semantic alignment with conditioning signals. Evaluation protocols are still evolving, with a combination of objective metrics and user studies commonly employed to assess the naturalness, diversity, and consistency of generated motions (Guo et al., 2022; Tevet et al., 2022; Petrovich et al., 2021; Chen et al., 2021; Huang et al., 2020; Liu et al., 2022a).

2.2 Riemannian Manifold

A smooth manifold $\mathcal{M}$ (Lee, 2012) is a topological space that locally resembles Euclidean space $\mathbb{R}^n$, which allows for the application of calculus, and it becomes a ***Riemannian manifold*** $(\mathcal{M}, g)$ when it is endowed with a Riemannian metric g (Lee, 2018). The metric is a smoothly varying inner product on each tangent space, defining the inner product between any two tangent vectors $u, v \in T_p\mathcal{M}$ as $g_p(u, v)$. This structure allows for the measurement of geometric properties, such as the length of a vector, and the angle between vectors. The metric also induces a distance function on the manifold, typically the geodesic distance, which measures the length of the shortest path between two points (Lee, 2018).

Furthermore, the Riemannian metric defines a canonical volume form $\mathrm{d}V$, which is essential for integration and defining probability distributions on the manifold (Lee, 2018). A probability density function $f : \mathcal{M} \to \mathbb{R}^+$ must satisfy the normalization condition $\int_{\mathcal{M}} f(p)\mathrm{d}V(p) = 1$. This has enabled the development of statistical models on non-Euclidean domains, such as the Riemannian uniform/normal distribution, which are crucial for modern data analysis as well as the generative modeling on the Riemannian manifold (Pennec, 2006; Said et al., 2017).

2.3 General Flow Matching

Flow matching (Lipman et al., 2022; Albergo et al., 2023; Liu et al., 2022b), combining aspects from Continuous Normalizing Flows and Diffusion Models, learns a time-dependent velocity field that transports a source distribution p_0 to a target distribution p_1. A standard training objective is the

conditional flow-matching loss:

$$\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}_{t,\boldsymbol{x}_1,\boldsymbol{x}_t} \left\| v_t(\boldsymbol{x}_t|\boldsymbol{x}_1) - v_\theta(\boldsymbol{x}_t, t) \right\|_2^2 \tag{1}$$

where $v_t(\boldsymbol{x}_t \mid \boldsymbol{x}_1)$ denotes the target conditional velocity. In Euclidean space, this target is typically induced by the linear interpolation between $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$, which yields a simple closed-form velocity field. Recent work extends the same principle to Riemannian manifolds (Chen & Lipman, 2023; Lipman et al., 2024), replacing Euclidean linear paths with geodesics and defining target velocities in the corresponding tangent spaces. Consistently, the geodesic degenerates to the linear interpolation when the manifold is the Euclidean space, which means that the Riemannian flow matching is a strict generalization of the Euclidean case. These formulations show that flow matching is not restricted to flat Euclidean domains and provide the foundation for geometry-aware generative modeling.

3 METHODOLOGY

In this section, we present our proposed method, RMG.

Notation. Unless specified otherwise, $\mathcal{M}$ denotes a Riemannian manifold. For $\boldsymbol{x} \in \mathcal{M}$, $T_{\boldsymbol{x}}\mathcal{M}$ denotes the tangent space at $\boldsymbol{x}$, and $T\mathcal{M}$ denotes the tangent bundle. We write $\mathrm{Exp}_{\boldsymbol{x}} : T_{\boldsymbol{x}}\mathcal{M} \to \mathcal{M}$ and $\mathrm{Log}_{\boldsymbol{x}} : \mathcal{M} \to T_{\boldsymbol{x}}\mathcal{M}$ for the exponential and logarithm maps, respectively. For embedded manifolds, $\Pi_{T_{\boldsymbol{x}}\mathcal{M}}(\cdot)$ denotes the projection operator onto $T_{\boldsymbol{x}}\mathcal{M}$.

3.1 MOTION REPRESENTATION AND MANIFOLD

Prior work typically factorizes a single motion frame into several parts as illustrated in Table 1. We adopt this decomposition but cast each factor on its natural Riemannian manifold, yielding a scale-free representation with intrinsic normalization. Unlike previous works that still apply dataset-level mean/standard-deviation normalization after forming the representation, the manifold structure (unit quaternions, pre-shapes, and a canonical length for translation) makes normalization unnecessary and enables geometry-aware modeling.

Global Translation ($\mathcal{T}$). Following common practice, we choose a specific joint of human body (usually chosen as the root joint, pelvis) to represent the global translation, which is a simple Euclidean space of $\mathbb{R}^3$. This factor captures the global trajectory of the motion and is essential for modeling locomotion and spatial movement.

Global Orientation and Per-Joint Rotations ($\mathcal{R}$). The articulated rotations can be represented by unit quaternions. For a skeleton with J joints, we write $q = \{q_j\}_{j=1}^J$ with $q_j \in \mathbb{H}$ and $\|q_j\|_2 = 1$. Inspired by the SMPL parameterization (Loper et al., 2015), we define a canonical reference pose (typically the T-pose) and express all rotations relative to it: q_1 encodes the global orientation ($\mathcal{R}_{\mathrm{Orientation}}$), while $\{q_j\}_{j=2}^J$ capture the local joint rotations in their respective local coordinate systems ($\mathcal{R}_{\mathrm{Joint}}$). Since each q_j is a unit quaternion, it lies on the hypersphere $\mathbb{S}^3$ (embedded in $\mathbb{R}^4$), and the rotation component lies on the product manifold $(\mathbb{S}^3)^J$.

Unlike continuous 6D rotations (Zhou et al., 2019), unit quaternions provide a compact representation of $\mathrm{SO}(3)$ through the antipodal identification $q \sim -q$ and induce smooth geodesics on $\mathbb{S}^3$ after a consistent sign choice. This improves interpolation and sampling stability, avoids re-orthogonalization, and reduces dimensionality from 6 to 4 with a consistent distance metric.

Local Pose ($\mathcal{P}$). We represent the within-frame skeletal configuration as a point in the Kendall pre-shape space(Kendall, 1984; 1989; Dryden & Mardia, 2016). Treating the J joints as landmarks in $\mathbb{R}^3$, the pre-shape is invariant to global translation and scale and thus captures only the relative joint configuration within each frame. Concretely, $\mathcal{S}_3^J$ denotes the set of centered configurations with unit Frobenius norm.

Given joint coordinates $P \in \mathbb{R}^{J\times 3}$, we remove global translation by centering and remove scale by Frobenius normalization:

$$p = \frac{P - \bar{P}}{\|P - \bar{P}\|_F} \in \mathcal{S}_3^J, \quad \bar{P} = \frac{1}{J}\sum_{j=1}^{J} P_j. \tag{2}$$

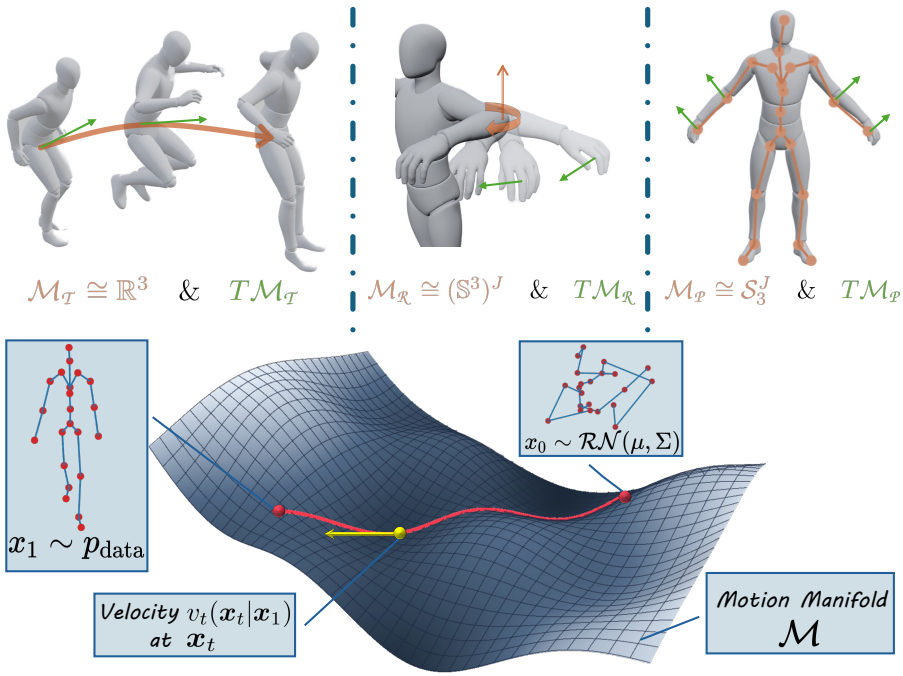


Figure 2: (**Top**) Illustration of the unified Riemannian representation for articulated motion. Each motion frame can be factorized into **global translation** ($\mathcal{M}_{\mathcal{T}}$), **global orientation and per-joint rotations** ($\mathcal{M}_{\mathcal{R}}$), and **local pose** ($\mathcal{M}_{\mathcal{P}}$) along with the **temporal differences** ($T\mathcal{M}_{\mathcal{F}}$ for $\mathcal{F} \in \{\mathcal{T}, \mathcal{R}, \mathcal{P}\}$). Each factor is represented on its natural Riemannian manifold, yielding a scale-free representation with intrinsic normalization. (**Bottom**) Illustration of the Riemannian flow matching process in the RMG manifold. $\mathcal{M}$ is defined by our proposed manifold Equation (3). The red line is the geodesic between $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$ while the yellow line with arrow is the velocity at $\boldsymbol{x}_t$.

Unlike variants that only subtract the root (or XZ-plane) translation, this pre-shape is fully translation-invariant and scale-free, making it well-suited for modeling the local pose factor.

Temporal Differentiation ($\mathrm{d}\cdot$)**.** For each factor, we can also include its temporal difference (e.g., velocity) as an additional component. Specifically, we can compute the temporal difference by $\mathrm{d}\boldsymbol{x}_t = \mathrm{Log}_{\boldsymbol{x}_t}(\boldsymbol{x}_{t+1}) \in T_{\boldsymbol{x}_t}\mathcal{M}_{\mathcal{F}}$ for $\mathcal{F} \in \{\mathcal{T}, \mathcal{R}, \mathcal{P}\}$. This captures the dynamic aspect of motion.

Unified View. Figure 2 illustrates our formulation, which provides us a unified view of motion representation as a product manifold of the composite factors and covers all the representation in the previous works. Take the HumanML3D (Guo et al., 2022) format without the foot contact indicator as an example, it corresponds to a product manifold of the form:

$$\mathcal{M}_{\text{H3D}} = T\mathcal{M}_{\mathcal{T}} \times T\mathcal{M}_{\mathcal{R}_{\text{Orientation}}} \times \mathcal{M}_{\mathcal{R}_{\text{Joint}}} \times \mathcal{M}_{\mathcal{P}} \times T\mathcal{M}_{\mathcal{P}}$$

However, using so many factors is redundant and not necessary. In this work, we argue global translation ($\mathcal{T}$), global orientation, and joint rotations ($\mathcal{R}$) are sufficient to capture articulated motion, which are verified through empirical studies (Section 4) and theoretical analysis (Section E). We therefore adopt a more compact representation that omits the pre-shape and other temporal differences, yielding

$$\mathcal{M}_{\text{RMG}} = \mathcal{M}_{\mathcal{T}} \times \mathcal{M}_{\mathcal{R}} = \mathbb{R}^3 \times (\mathbb{S}^3)^J, \tag{3}$$

We will elaborate and justify this choice in the ablation study (Section 4.3).

3.2 Prior Distribution

Riemannian Gaussian Distribution. We first introduce a mean-centered wrapped Gaussian distribution on a general Riemannian manifold $\mathcal{M}$. Given a reference (mean) point $\mu \in \mathcal{M}$, we draw

Gaussian noise in its embedded Euclidean space $\mathbb{R}^n$, map it to the tangent space $T_\mu\mathcal{M}$ (via a projection operator), and then "wrap" it onto the manifold using the exponential map:

$$\xi \sim \mathcal{N}(0, \Sigma) \in \mathbb{R}^n, \qquad v = \Pi_{T_\mu\mathcal{M}}(\xi) \in T_\mu\mathcal{M}, \qquad z = \mathrm{Exp}_\mu\big(v\big) \in \mathcal{M},$$

where Σ is typically chosen block-diagonal. Such distribution can be denoted as $\mathcal{RN}(\mu, \Sigma)$.

Choice of Reference Point. The reference point μ can be chosen arbitrarily, but a good choice improves sampling quality. For motion generation, we set μ to the *rest pose* with zero translation and identity rotations, which is a natural center of the motion manifold. This choice ensures that samples from the prior correspond to plausible static poses, providing a meaningful starting point for motion synthesis.

Specifically, for the translation factor, we set $\mu_\mathcal{T} = \mathbf{0} \in \mathbb{R}^3$; for the rotation factor, we set $\mu_\mathcal{R} = [1, 0, 0, 0] \in \mathbb{S}^3$; for the pre-shape factor (if used), we set $\mu_\mathcal{P}$ to the canonical T-pose after removing global translation and normalizing using Equation (2). In our chosen manifold $\mathcal{M}_{\mathrm{RMG}}$ (Equation (3)), the reference point is thus

$$\mu = (\mu_\mathcal{T}, \underbrace{\mu_\mathcal{R}, \mu_\mathcal{R}, \ldots, \mu_\mathcal{R}}_{J \text{ times}}) \in \mathcal{M}_{\mathrm{RMG}}.$$

3.3 Training and Inference

We train a time-dependent vector field on the motion manifold $\mathcal{M}$ with Riemannian flow matching. Let $\boldsymbol{x}_1 \sim p_{\mathrm{data}}$ denote a real motion sample and $\boldsymbol{x}_0 \sim p_0$ a prior sample (Section 3.2). For $t \sim \mathcal{U}[0, 1]$, we construct the interpolation state on the geodesic from $\boldsymbol{x}_0$ to $\boldsymbol{x}_1$:

$$\boldsymbol{x}_t \;=\; \mathrm{Exp}_{\boldsymbol{x}_0}\big(t\,\mathrm{Log}_{\boldsymbol{x}_0}(\boldsymbol{x}_1)\big),$$

On product manifolds, Exp and Log are applied factor-wise. The translation factor is Euclidean, while rotation and (optional) pre-shape factor use the corresponding manifold maps. The supervision signal is the geodesic tangent at $\boldsymbol{x}_t$, written as

$$v_t(\boldsymbol{x}_t \,|\, \boldsymbol{x}_1) \;=\; \frac{1}{1-t}\,\mathrm{Log}_{\boldsymbol{x}_t}(\boldsymbol{x}_1) \;\in\; T_{\boldsymbol{x}_t}\mathcal{M},$$

for $t < 1$, away from the cut locus. At $t = 1$, the velocity is defined by continuous extension along the geodesic. This expression reduces to the standard Euclidean flow-matching target when $\mathcal{M} = \mathbb{R}^n$.

We parameterize the vector field by a neural network $v_\theta(\boldsymbol{x}_t, t)$. Since the output of the neural network is in the ambient Euclidean space, we must project it to the tangent space to enforce valid manifold dynamics: $\Pi_{T_{\boldsymbol{x}_t}\mathcal{M}} v_\theta(\boldsymbol{x}_t, t) \in T_{\boldsymbol{x}_t}\mathcal{M}$.

Training minimizes the mean-squared error between target and predicted tangent velocities:

$$\mathcal{L}(\theta) \;=\; \mathbb{E}_{\boldsymbol{x}_1 \sim p_{\mathrm{data}},\, \boldsymbol{x}_0 \sim p_0,\, t \sim \mathcal{U}[0,1]}\Big[\big\|v_t(\boldsymbol{x}_t \,|\, \boldsymbol{x}_1) - \Pi_{T_{\boldsymbol{x}_t}\mathcal{M}} v_\theta(\boldsymbol{x}_t, t)\big\|_g^2\Big]. \tag{4}$$

For the embedded manifolds used in this work, g is the metric induced by the ambient Euclidean inner product, so this norm agrees with the Euclidean norm of tangent vectors.

At inference time, we sample $\boldsymbol{x}_0 \sim p_0$ and integrate the learned manifold ODE:

$$\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} \;=\; \Pi_{T_{\boldsymbol{x}_t}\mathcal{M}} v_\theta(\boldsymbol{x}_t, t), \qquad \boldsymbol{x}_0 \sim p_0, \tag{5}$$

from $t = 0$ to $t = 1$. With step size h, a first-order Riemannian Euler update is $\boldsymbol{x}_{t+h} = \mathrm{Exp}_{\boldsymbol{x}_t}\big(h\,\Pi_{T_{\boldsymbol{x}_t}\mathcal{M}} v_\theta(\boldsymbol{x}_t, t)\big)$, which preserves manifold constraints by construction.

We emphasize that $T\mathcal{M}_\mathcal{F}$, with $\mathcal{F} \in \{\mathcal{T}, \mathcal{R}, \mathcal{P}\}$, in Section 3.1 and $T_{\boldsymbol{x}_t}\mathcal{M}$ in Section 3.3 refer to different objects despite the similar notation. The former appears in the factorized motion representation and denotes optional temporal-difference components attached to data samples, which therefore belong to the data distribution $p_{\mathrm{data}} = p_1$. The latter is the tangent space at the interpolation state $\boldsymbol{x}_t$ and contains the time-dependent velocity field used by Riemannian flow matching, corresponding to the intermediate probability flow p_t.

4 EXPERIMENTS

4.1 EXPERIMENT SETUP

Datasets. The experiments are mainly conducted on *HumanML3D* (Guo et al., 2022). HumanML3D is a large-scale language-motion dataset comprising 14,616 motions and 44,970 text descriptions building upon the AMASS dataset (Mahmood et al., 2019). Besides HumanML3D, we also employ *MotionMillion* (Fan et al., 2025), which is a recently released large-scale motion dataset with 1 million motion clips and 4 million text descriptions, to pre-train our model and evaluate its generalization ability.

Evaluation Metrics. Following previous works (Guo et al., 2022; Chen et al., 2023; Tevet et al., 2022), we employ 4 main metrics to evaluate our framework. The Frechet Inception Distance is used to measure the motion quality as well as the feature distributions. The Diversity and MultiModality Distance are incorporated to measure the generation diversity. Lastly, the R Precision aims to evaluate the matching rate of the conditions.

Implementation Details. We use the Diffusion Transformer (Peebles & Xie, 2023) as the model backbone for the flow matching. For text encoding, we incorporate Qwen (Zhang et al., 2025; Yang et al., 2025), utilizing the encoded hidden states as the text representation. During training, we employ the AdamW (Loshchilov & Hutter, 2017) optimizer with a cosine learning-rate schedule and linear warmup: the learning rate is first warmed up linearly to 10^{-4} and then cosine-annealed to near zero. Training is performed in a classifier-free manner (Ho & Salimans, 2022) with a dropout rate of 10%. To stabilize both training and inference, we adopt an exponential moving average (EMA) strategy.

4.2 MAIN RESULTS

Table 2: Evaluation of *text-based motion generation* on HumanML3D (Guo et al., 2022) dataset. Reported metrics are FID, R@1, Diversity, and MModality. The models in bold are the optimal models, and the models in underline are the sub-optimal models. The guidance scale is set to 6.5.

Method	HumanML3D Format				MotionStreamer Format			
	FID↓	R@1↑	Div→	MModality↑	FID↓	R@1↑	Div→	MModality↑
GT	0.002	0.511	9.503	2.799	0.002	0.702	27.492	–
MLD (2023)	0.473	0.481	9.724	2.413	18.236	0.546	26.352	–
T2M-GPT(2023)	0.116	0.492	9.761	1.856	12.475	0.606	27.275	–
MotionGPT (2023)	0.232	0.492	**9.528**	2.008	14.375	0.456	27.114	–
MoMask (2024)	<u>0.045</u>	0.521	–	1.241	12.232	0.621	27.127	–
MotionLCM (2024)	0.304	0.505	9.607	2.259	–	–	–	–
MotionCLR (2024)	0.269	**0.542**	9.607	1.985	–	–	–	–
MotionLab (2025)	0.167	–	9.593	**2.912**	–	–	–	–
MARDM (2025)	0.114	0.500	–	2.231	–	–	–	–
MotionStreamer (2025)	–	–	–	–	<u>11.790</u>	<u>0.631</u>	<u>27.284</u>	–
Ours	**0.043**	<u>0.525</u>	<u>9.555</u>	<u>2.748</u>	**5.835**	**0.710**	**27.672**	**14.906**

We first evaluate our method on the HumanML3D text-to-motion generation benchmark and compare it with several strong baselines, including both diffusion-based methods (e.g., MotionDiffuse (Zhang et al., 2024a), MDM (Tevet et al., 2022), MotionLab (Guo et al., 2025)) and autoregressive methods (e.g., T2M-GPT (Zhang et al., 2023), MGPT (Jiang et al., 2023), MoMask (Guo et al., 2024)). We report results under two output formats: the standard HumanML3D format and the MotionStreamer format. *It is worth noting that we implement extra functions to convert our Riemannian representation to either the HumanML3D format or MotionStreamer format for fair comparison. Refer to supplementary materials Section G.3 for details.*

Table 2 reports quantitative results of the 4 metrics on HumanML3D under two formats, where we focus primarily on the standard HumanML3D format. In this setting, our method achieves the

best FID (0.043), slightly surpassing the previous best MoMask (0.045), indicating stronger motion realism and distribution matching. At the same time, our model maintains strong text-motion consistency with R@1 = 0.525 (second only to MotionCLR at 0.542), while preserving high generation diversity and multimodality (Div = 9.555 and MModality = 2.748). Compared with prior methods that optimize only part of this trade-off (e.g., lower FID or higher R@1 alone), our model provides a more balanced improvement across quality, alignment, and diversity, which is critical for practical text-to-motion generation. For completeness, under the MotionStreamer format our method ranks first among all reported metrics (FID = 5.835, R@1 = 0.710, Div = 27.672, MModality = 14.906), further supporting the robustness of the learned Riemannian representation across output formats. We also provide additional results in the supplementary materials.

We further evaluate text-based motion generation on the large-scale MotionMillion benchmark to assess whether the advantages of our representation persist in a substantially broader data regime. Table 3 shows that our model (*Ours, 0.5B+1.7B*) outperforms the MotionMillion baselines across both fidelity and text alignment, while also exhibiting a clear and favorable guidance trade-off. With guidance scale 2.0, our method attains better FID of 5.6, improving substantially over the strongest baseline MotionMillion-7B (10.3) and thus reducing the distribution gap by nearly half. Increasing the guidance scale to 3.0 further raises R@1 from 0.81 to 0.86, while still maintaining a stronger FID than previous baselines. These results are particularly notable because our approach remains superior to substantially when scaling up, indicating that the gains come not merely from model scaling, but from the effectiveness of the proposed Riemannian motion representation and flow-matching formulation.

Table 3: Evaluation of *text-based motion generation* on MotionMillion (Fan et al., 2025) dataset. Reported metrics are FID and R@1. In our method, 0.5B and 1.7B refer to the parameter sizes of our Riemannian flow matching model and Qwen3-1.7B, respectively. G.S. refers to the guidance scale.

Method	**G.S.**	**FID↓**	**R@1↑**
ScaMo (2025)	–	89.0	0.67
MotionMillion-3B	–	10.8	0.79
MotionMillion-7B	–	10.3	0.79
Ours (0.5B+1.7B)	2.0	**5.6**	<u>0.81</u>
Ours (0.5B+1.7B)	3.0	<u>7.8</u>	**0.86**

Overall, these findings highlight the scalability and generalization ability of our **RMG** framework, which can be effectively applied to both small and large datasets and models while maintaining superior performance across key metrics.

4.3 Ablation Studies

In this section, we conduct ablation studies to analyze the impact of different factors in our framework, and we answer two main research questions:

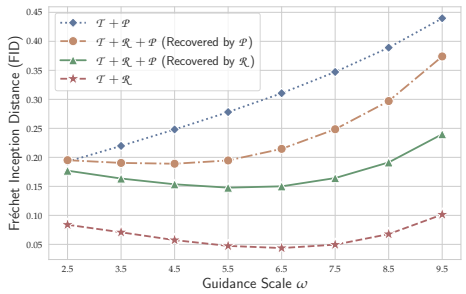
RQ1: Which factors matter for motion quality and guidance stability?

RQ2: Does temporal difference modeling improve motion quality?

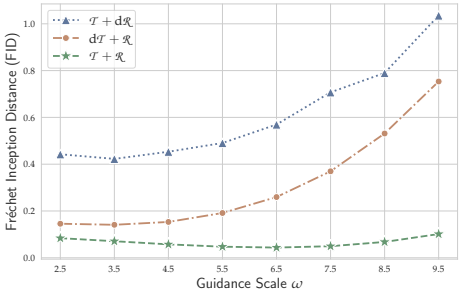
RQ1. To isolate the role of each factor, we keep the same architecture and training setup and only change the representation, then sweep the guidance scale $\omega \in [2.5, 9.5]$. Figure 3a shows that $\mathcal{T} + \mathcal{R}$ is consistently the best and most stable setting: its FID decreases from approximately 0.084 ($\omega = 2.5$) to a minimum of about 0.043 ($\omega = 6.5$), and remains low even at large guidance (0.101 at $\omega = 9.5$). In contrast, $\mathcal{T} + \mathcal{P}$ degrades monotonically as guidance increases (from 0.20 to 0.44), indicating poor robustness without rotation information.

For $\mathcal{T} + \mathcal{R} + \mathcal{P}$, the term 'Recovered by $\cdot$' means different conversion functions used to convert to the HumanML3D format for evaluation. Refer to Section G.3 for details. We find that recovering with $\mathcal{R}$ is consistently better than recovering with $\mathcal{P}$ across the entire guidance range (e.g., 0.043 vs. 0.084 at $\omega = 2.5$, and 0.101 vs. 0.44 at $\omega = 9.5$). This suggests that the rotation component is more critical for maintaining motion quality and stability under guidance, while the pose-coordinate component does not provide additional robustness in this setup.

This trend is also aligned with practical motion representations: in real animation and robotics pipelines, motion is predominantly driven by global/root translation and joint rotations. By contrast,



(a) We study the impact of the Riemannian representation including $\mathcal{T}+\mathcal{R}$, $\mathcal{T}+\mathcal{R}+\mathcal{P}$ and $\mathcal{T}+\mathcal{P}$.



(b) We study the impact of temporal difference modeling in our framework including $\mathcal{T}+\mathcal{R}$, $\mathrm{d}\mathcal{T}+\mathcal{R}$, $\mathcal{T}+\mathrm{d}\mathcal{R}$.

Figure 3: Ablation study on different factors of our framework. All the models are trained with the same setting (including the parameter size, random seed) and evaluated on the HumanML3D benchmark.

pose-coordinate representations are more affected by subject-specific factors (e.g., body scale and limb proportions), which introduces additional variability across people. Taken together, these observations suggest that $\mathcal{T}+\mathcal{R}$ is a compact and robust representation that is sufficient for motion modeling in our setting.

RQ2. To answer whether temporal difference modeling improves motion quality, we compare $\mathcal{T}+\mathcal{R}$ with $\mathrm{d}\mathcal{T}+\mathcal{R}$ and $\mathcal{T}+\mathrm{d}\mathcal{R}$ under the same guidance sweep $\omega \in [2.5, 9.5]$. Figure 3b shows that $\mathcal{T}+\mathcal{R}$ consistently attains the lowest FID across the full range of ω. In contrast, $\mathrm{d}\mathcal{T}+\mathcal{R}$ degrades steadily as guidance increases ($0.14 \rightarrow 0.75$), and $\mathcal{T}+\mathrm{d}\mathcal{R}$ is less stable still ($0.44 \rightarrow 1.03$). This behavior is consistent with the representation property of temporal differencing: it attenuates absolute/global motion state (e.g., global trajectory and orientation) while emphasizing local frame-to-frame variation, which weakens long-horizon structure modeling. Overall, temporal differencing does not improve performance in our setting, and absolute modeling of translation and rotation ($\mathcal{T}+\mathcal{R}$) remains the most robust choice for guidance-stable generation.

5 Conclusion

Limitations. Though our work has demonstrated promising results, there are certain limitations including lack of further exploration of more condition modalities and the potential for facial and hand motion modeling. We leave more discussion in the supplementary materials Section H.

In conclusion, we proposed **Riemannian Motion Generation (RMG)**, a unified geometric framework for human motion representation and generation. Instead of treating motion as an unconstrained Euclidean vector, we modeled it on a product manifold and learned generation dynamics with Riemannian flow matching. This design yields a compact representation, $\mathbb{R}^3 \times (\mathbb{S}^3)^J$, together with a geometry-consistent training and inference pipeline based on geodesic interpolation, tangent-space vector-field learning, and manifold-preserving integration.

Extensive experiments demonstrated that this geometric formulation is both effective and scalable. RMG achieves strong and balanced performance on HumanML3D across motion quality, text alignment, and diversity, and generalizes well to the larger MotionMillion benchmark. Our ablation studies further showed that translation and rotation factors are sufficient in practice, while adding extra factors does not necessarily improve generation quality. Overall, the results support representation geometry as a core design principle for motion generation.

AI Usage Statement

In this work, we used generative AI tools for generating parts of the evaluation code to match standard evaluation protocols. We have not used generative AI tools for other tasks requiring disclosure, and the remaining required disclosure tasks are not applicable to this work. Additionally, we used

generative AI tools for text editing and language polishing. We reviewed all AI-assisted text and verified and tested the AI-assisted evaluation code as part of our experimental workflow. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

Ethics Statement

This work advances large-scale human motion generation and may support applications in gaming, virtual reality, human–computer interaction, and embodied agents. It may also inform research on geometric representations and low-dimensional manifold modeling in other domains.

More capable human motion generation systems can be misused to synthesize deceptive or misleading behaviors, and biases or artifacts in the training data may be inherited or amplified in generated motions. Deploying such models in embodied or interactive systems without sufficient safeguards can also create safety, fairness, or reliability concerns. We therefore encourage careful dataset curation, clear disclosure of synthetic content, and application-level safety mechanisms in downstream use.

Reproducibility Statement

We support transparent assessment and facilitate reproduction by specifying the RMG representation, Riemannian flow-matching objective, and inference procedure in Section 3. The supplementary PDF provides training and inference pseudocode, theoretical analysis, model architectures, training hyperparameters, conversion functions for evaluating our representation under the HumanML3D and MotionMillion protocols, full benchmark results, and uncertainty estimates from 20 independent HumanML3D runs. Section 4 describes the datasets, evaluation metrics, implementation choices, and ablation settings. The accompanying ZIP archive contains a local webpage for qualitative visualization of generated motions. Together, these materials document the method and experimental evaluation in detail and enable inspection of the reported results.

References

Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

Tenglong Ao, Qingzhe Gao, Yuke Lou, Baoquan Chen, and Libin Liu. Rhythmic gesticulator: Rhythm-aware co-speech gesture synthesis with hierarchical neural embeddings. *ACM Transactions on Graphics (TOG)*, 41(6):1–19, 2022.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Huanqia Cai, Sihan Cao, Ruoyi Du, Peng Gao, Steven Hoi, Zhaohui Hou, Shijie Huang, Dengyang Jiang, Xin Jin, Liangchen Li, et al. Z-image: An efficient image generation foundation model with single-stream diffusion transformer. *arXiv preprint arXiv:2511.22699*, 2025.

Kang Chen, Zhipeng Tan, Jin Lei, Song-Hai Zhang, Yuan-Chen Guo, Weidong Zhang, and Shi-Min Hu. Choreomaster: choreography-oriented music-driven dance synthesis. *ACM Transactions on Graphics (TOG)*, 40(4):1–13, 2021.

Ling-Hao Chen, Shunlin Lu, Wenxun Dai, Zhiyang Dou, Xuan Ju, Jingbo Wang, Taku Komura, and Lei Zhang. Pay attention and move better: Harnessing attention for interactive motion generation and training-free editing. *arXiv preprint arXiv:2410.18977*, 2024.

Ricky TQ Chen and Yaron Lipman. Flow matching on general geometries. *arXiv preprint arXiv:2302.03660*, 2023.

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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18000–18010, 2023.

Rishabh Dabral, Muhammad Hamza Mughal, Vladislav Golyanik, and Christian Theobalt. Mofusion: A framework for denoising-diffusion-based motion synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9760–9770, 2023.

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 *European Conference on Computer Vision*, pp. 390–408. Springer, 2024.

Bruno Degardin, Joao Neves, Vasco Lopes, Joao Brito, Ehsan Yaghoubi, and Hugo Proença. Generative adversarial graph convolutional networks for human action synthesis. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1150–1159, 2022.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL https://arxiv.org/abs/1810.04805.

Ian L. Dryden and Kanti V. Mardia. *Statistical Shape Analysis, with Applications in R*. Wiley, 2 edition, 2016.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Ke Fan, Shunlin Lu, Minyue Dai, Runyi Yu, Lixing Xiao, Zhiyang Dou, Junting Dong, Lizhuang Ma, and Jingbo Wang. Go to zero: Towards zero-shot motion generation with million-scale data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13336–13348, 2025.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Chuan Guo, Xinxin Zuo, Sen Wang, Shihao Zou, Qingyao Sun, Annan Deng, Minglun Gong, and Li Cheng. Action2motion: Conditioned generation of 3d human motions. In *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 2021–2029, 2020.

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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5152–5161, 2022.

Chuan Guo, Yuxuan Mu, Muhammad Gohar Javed, Sen Wang, and Li Cheng. Momask: Generative masked modeling of 3d human motions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1900–1910, 2024.

Ziyan Guo, Zeyu Hu, De Wen Soh, and Na Zhao. Motionlab: Unified human motion generation and editing via the motion-condition-motion paradigm. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13869–13879, 2025.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Ruozi Huang, Huang Hu, Wei Wu, Kei Sawada, Mi Zhang, and Daxin Jiang. Dance revolution: Long-term dance generation with music via curriculum learning. In *International conference on learning representations*, 2020.

Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. Motiongpt: Human motion as a foreign language. *Advances in Neural Information Processing Systems*, 36:20067–20079, 2023.

David G. Kendall. Shape manifolds, procrustean metrics, and complex projective spaces. *Bulletin of the London Mathematical Society*, 16(2):81–121, 1984.

David G Kendall. A survey of the statistical theory of shape. *Statistical Science*, 4(2):87–99, 1989.

Jihoon Kim, Jiseob Kim, and Sungjoon Choi. Flame: Free-form language-based motion synthesis & editing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 8255–8263, 2023.

Jinwoo Kim, Heeseok Oh, Seongjean Kim, Hoseok Tong, and Sanghoon Lee. A brand new dance partner: Music-conditioned pluralistic dancing controlled by multiple dance genres. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3490–3500, 2022.

Diederik P Kingma, Max Welling, et al. Auto-encoding variational bayes, 2013.

John M. Lee. *Introduction to Smooth Manifolds*. Springer, 2 edition, 2012.

John M. Lee. *Introduction to Riemannian Manifolds*. Springer, 2 edition, 2018.

Ruilong Li, Shan Yang, David A Ross, and Angjoo Kanazawa. Ai choreographer: Music conditioned 3d dance generation with aist++. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 13401–13412, 2021.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky TQ Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code. *arXiv preprint arXiv:2412.06264*, 2024.

Haiyang Liu, Naoya Iwamoto, Zihao Zhu, Zhengqing Li, You Zhou, Elif Bozkurt, and Bo Zheng. Disco: Disentangled implicit content and rhythm learning for diverse co-speech gestures synthesis. In *Proceedings of the 30th ACM international conference on multimedia*, pp. 3764–3773, 2022a.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022b.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, October 2015.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Shunlin Lu, Ling-Hao Chen, Ailing Zeng, Jing Lin, Ruimao Zhang, Lei Zhang, and Heung-Yeung Shum. Humantomato: Text-aligned whole-body motion generation. *arXiv preprint arXiv:2310.12978*, 2023.

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 *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 27872–27882, 2025.

Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *International Conference on Computer Vision*, pp. 5442–5451, October 2019.

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 *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 27859–27871, 2025.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pp. 8162–8171. PMLR, 2021.

William S Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

Xavier Pennec. Intrinsic statistics on riemannian manifolds: Basic tools for geometric measurements. *Journal of Mathematical Imaging and Vision*, 25(1):127–154, 2006.

Mathis Petrovich, Michael J Black, and Gül Varol. Action-conditioned 3d human motion synthesis with transformer vae. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10985–10995, 2021.

Mathis Petrovich, Michael J Black, and Gül Varol. Temos: Generating diverse human motions from textual descriptions. In *European Conference on Computer Vision*, pp. 480–497. Springer, 2022.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pp. 1530–1538. PMLR, 2015.

Salem Said, Lionel Bombrun, Yannick Berthoumieu, and Jonathan H. Manton. Riemannian gaussian distributions on the space of symmetric positive definite matrices. *IEEE Transactions on Information Theory*, 63(4):2153–2170, 2017.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. pmlr, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H Bermano. Human motion diffusion model. *arXiv preprint arXiv:2209.14916*, 2022.

Jonathan Tseng, Rodrigo Castellon, and Karen Liu. Edge: Editable dance generation from music. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 448–458, 2023.

Guillermo Valle-Pérez, Gustav Eje Henter, Jonas Beskow, Andre Holzapfel, Pierre-Yves Oudeyer, and Simon Alexanderson. Transflower: probabilistic autoregressive dance generation with multimodal attention. *ACM Transactions on Graphics (TOG)*, 40(6):1–14, 2021.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10086–10096, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Ping Yu, Yang Zhao, Chunyuan Li, Junsong Yuan, and Changyou Chen. Structure-aware human-action generation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX 16*, pp. 18–34. Springer, 2020.

Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Yong Zhang, Hongwei Zhao, Hongtao Lu, Xi Shen, and Ying Shan. Generating human motion from textual descriptions with discrete representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14730–14740, 2023.

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 transactions on pattern analysis and machine intelligence*, 46(6):4115–4128, 2024a.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 7368–7376, 2024b.

Chongyang Zhong, Lei Hu, Zihao Zhang, and Shihong Xia. Attt2m: Text-driven human motion generation with multi-perspective attention mechanism. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 509–519, 2023.

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5745–5753, 2019.

Wentao Zhu, Xiaoxuan Ma, Dongwoo Ro, Hai Ci, Jinlu Zhang, Jiaxin Shi, Feng Gao, Qi Tian, and Yizhou Wang. Human Motion Generation: A Survey . *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 46(04):2430–2449, April 2024. ISSN 1939-3539. doi: 10.1109/TPAMI.2023.3330935. URL https://doi.ieeecomputersociety.org/10.1109/TPAMI.2023.3330935.

A SUPPLEMENTARY MOTION VIDEOS

The accompanying ZIP archive contains a webpage for browsing videos of human motions generated by our method. After extracting the archive, double-click `index.html` to open the webpage in a browser. Individual videos can also be viewed directly in the `static/videos/` directory.

B EXTENDED RELATED WORKS

B.1 DIFFUSION- AND FLOW-BASED GENERATIVE MODELS

The generative models for continuous data have been dominated by diffusion- and flow-based approaches (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2020; Lipman et al., 2022; Albergo et al., 2023; Liu et al., 2022b), especially in image and video generation.

Early diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020) are based on a forward noising process that gradually corrupts data with Gaussian noise, and a reverse denoising process that learns to recover the original data from noisy inputs.

In a unified view of diffusion- and flow-based model, the generative process is formulated as a probability flow that transports a simple reference distribution (e.g., Gaussian noise, $\boldsymbol{x}_0 \sim p_0$) to the data distribution ($\boldsymbol{x}_1 \sim p_1$) through an ordinary differential equation (ODE). Formally, let α_t a monotonically increasing function with respect to t such that $\alpha_0 = 0$ and $\alpha_1 = 1$, while σ_t is a monotonically decreasing function with respect to t such that $\sigma_0 = 1$ and $\sigma_1 = 0$, then:

$$\boldsymbol{x}_t = \alpha_t \boldsymbol{x}_1 + \sigma_t \boldsymbol{x}_0$$

The learning objective of diffusion- and flow-based models are:

$$\mathcal{L}_{\text{Diff}} = \mathbb{E}_{t,\boldsymbol{x}_1,\boldsymbol{x}_0} \left\| \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t) - \boldsymbol{x}_0 \right\|_2^2, \qquad \boldsymbol{x}_0\text{-parameterization} \tag{6}$$

$$\mathcal{L}_{\text{CFM}} = \mathbb{E}_{t,\boldsymbol{x}_1,\boldsymbol{x}_0} \left\| \boldsymbol{v}_\theta(\boldsymbol{x}_t, t) - (\dot{\alpha}_t \boldsymbol{x}_1 + \dot{\sigma}_t \boldsymbol{x}_0) \right\|_2^2, \quad \boldsymbol{v}\text{-parameterization} \tag{7}$$

The α_t and σ_t are called scheduler, and different across different literatures. For example, the scheduler of original diffusion model (Ho et al., 2020) are any monotonically increasing and decreasing functions restricted by $\alpha_t^2 + \sigma_t^2 = 1$. The cosine scheduler (Nichol & Dhariwal, 2021) are formulated as $\alpha_t = \sin(\pi t/2), \sigma_t = \cos(\pi t/2)$. Recent popular scheduler (Lipman et al., 2022; Liu et al., 2022b) are just linear scheduler with $\alpha_t = t, \sigma_t = 1 - t$.

The generation process is the integration of the learned velocity field from $t = 0$ to $t = 1$, which can be solved by any ODE solver. For t such that $\alpha_t > 0$, the velocity can be written as

$$\mathrm{d}\boldsymbol{x}_t = \underbrace{\boldsymbol{v}_\theta(\boldsymbol{x}_t, t)\mathrm{d}t}_{\boldsymbol{v}-\text{parameterization}} = \underbrace{\left[\frac{\dot{\alpha}_t}{\alpha_t} \boldsymbol{x}_t + \left(\dot{\sigma}_t - \frac{\dot{\alpha}_t}{\alpha_t} \sigma_t \right) \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t) \right] \mathrm{d}t}_{\boldsymbol{x}_0-\text{parameterization}}.$$

B.2 LANGUAGE MODEL-STYLE GENERATIVE MODELS

Motivated by the success of Transformer-based language models (Brown et al., 2020; Devlin et al., 2019), discrete-token generative models have been widely adopted for human motion generation. GPT-style approaches (Fan et al., 2025; Xiao et al., 2025; Jiang et al., 2023; Lu et al., 2023; 2025; Zhang et al., 2023; 2024b; Zhong et al., 2023) first quantize motion sequences into discrete tokens and then train autoregressive Transformers to predict tokens sequentially. Another line of work (Kim et al., 2023; Guo et al., 2024) uses BERT-style bidirectional Transformers to reconstruct masked motion segments under text guidance.

However, the motions are actually continuous in nature, and the quantization method is an important question. Current approaches typically use a Vector Quantized Variational Autoencoder (VQ-VAE) (Van Den Oord et al., 2017) to learn a discrete codebook, while Guo et al. introduces Residual VQ-VAE (RVQ-VAE) to further reduce the quantization error.

C ALGORITHMS

We present the training and inference algorithms for Riemannian Flow Matching in Algorithm 1 and Algorithm 2, respectively. The algorithms are mainly focused on the unconditional generation setting, but they can be easily extended to the conditional generation setting by incorporating conditioning signals into the velocity field $\boldsymbol{v}_\theta$ and the training loss.

Algorithm 1: Training

Input: Training data $\boldsymbol{x}_1 \sim p_{\text{data}}$, The Motion Manifold $\mathcal{M}$ (e.g. $\mathcal{M}_{\text{RMG}}$ in our case)

Output: $\boldsymbol{v}_\theta$

while *Training not end* **do**

- Get a batch of training data $\boldsymbol{x}_1 \sim p_{\text{data}}$
- Project $\boldsymbol{x}_1 = \text{Proj}_{\mathcal{M}}(\boldsymbol{x}_1)$ `// Ensure` $\boldsymbol{x}_1$ `resides on the manifold.`
- Sample $\boldsymbol{x}_0 \sim \mathcal{RN}(\mu, \mathcal{I}),\ t \sim \text{Uniform}(0, 1)$
- Compute the linear interpolation on manifold $\boldsymbol{x}_t = \text{Geodesic}(\boldsymbol{x}_0, \boldsymbol{x}_1, t)$ along with the geodesic velocity $\boldsymbol{v}_t = \dot{\boldsymbol{x}}_t$
- Compute the population loss $\mathcal{L} = \mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{x}_1, t}[\|\boldsymbol{v}_t - \Pi_{T_{\boldsymbol{x}_t}\mathcal{M}} \boldsymbol{v}_\theta(\boldsymbol{x}_t, t)\|^2]$
- Backpropagation and update θ

Algorithm 2: Inference

Input: The learned velocity field $\boldsymbol{v}_\theta$, The Motion Manifold $\mathcal{M}$ (e.g. $\mathcal{M}_{\text{RMG}}$ in our case), Timesteps $0 = t_0 < t_1 < \ldots < t_N = 1$

Output: Generated motion samples $\boldsymbol{x}_1$

Sample $\boldsymbol{x}_0 \sim \mathcal{RN}(\mu, \mathcal{I})$

Let $\boldsymbol{x}_{t_0} = \boldsymbol{x}_0$

for $i = 0$ **to** $N - 1$ **do**

- Compute the velocity $\boldsymbol{v}_i = \Pi_{T_{\boldsymbol{x}_t}\mathcal{M}} \boldsymbol{v}_\theta(\boldsymbol{x}_{t_i}, t_i)$ `// Ensure the velocity resides on the tangent space of` $\boldsymbol{x}_{t_i}$
- Update $\boldsymbol{x}_{t_{i+1}} = \text{Exp}_{\boldsymbol{x}_{t_i}}((t_{i+1} - t_i) \cdot \boldsymbol{v}_i)$ on the manifold $\mathcal{M}$

return $\boldsymbol{x}_1 = \boldsymbol{x}_{t_N}$

D MANIFOLD

In this paper, we focus on three types of Riemannian manifolds: Euclidean space $\mathbb{R}^3$, the hypersphere $\mathbb{S}^3$, and the pre-shape space $\mathcal{S}_3^J$.

Euclidean space. Euclidean space $\mathbb{R}^3$ is a flat manifold with zero curvature. It is the most familiar manifold, where the distance between two points is simply their straight-line distance. Most existing diffusion models are built in this space.

Hypersphere. The hypersphere $\mathbb{S}^3$ is a curved manifold with constant positive curvature. It is embedded in $\mathbb{R}^4$ and consists of points at a fixed distance from the origin. In this work, we use the hypersphere to model quaternions. Since q and $-q$ represent the same rotation, we select a canonical sign during dataset construction by choosing the representative in the closed upper hemisphere of $\mathbb{S}^3$. In other words, for any point $q = (w, x, y, z) \in \mathbb{S}^3$, we choose $q_0 \geq 0$, with a fixed tie-breaking rule when $q_0 = 0$.

For Riemannian flow matching on a general hyper-sphere $\mathcal{S}^d$, consider two points $\boldsymbol{x}_0, \boldsymbol{x}_1 \in \mathbb{R}^{d+1}$ on the hypersphere $\mathcal{S}^d$. The geodesic between them can be written as

$$\gamma(t) = \frac{\sin((1-t)\theta)}{\sin(\theta)}\boldsymbol{x}_0 + \frac{\sin(t\theta)}{\sin(\theta)}\boldsymbol{x}_1,$$

where $\theta = \arccos(\langle \boldsymbol{x}_0, \boldsymbol{x}_1 \rangle)$ is the angle between $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$. The geodesic velocity at time t is given by:

$$\dot{\gamma}(t) = \frac{\theta}{\sin(\theta)}\left(-\cos((1-t)\theta)\boldsymbol{x}_0 + \cos(t\theta)\boldsymbol{x}_1\right).$$

Pre-shape space. The pre-shape space models the relative configuration of points. For example, the set of all triangles in $\mathbb{R}^2$ forms a pre-shape space, denoted by $\mathcal{S}_2^3$. In this work, we use the pre-shape space $\mathcal{S}_3^J$ to model the human skeleton, which consists of J joints in $\mathbb{R}^3$. A pre-shape is represented by a $J \times 3$ matrix, where each row gives the coordinates of one joint. The pre-shape space is defined as the set of all such matrices that are centered at the centroid and normalized to unit Frobenius norm. A concrete formulation is given in Equation (2). Intrinsically, the pre-shape space can be viewed as a high-dimensional hypersphere, so geodesics can be computed in the same way as on the hypersphere.

Beyond the pre-shape space, one can also consider the shape space, which is the quotient of the pre-shape space by the rotation group: $\mathcal{S}_3^J/\mathrm{SO}(3)$. However, geodesic computation in the shape space is substantially more involved, so we only study the pre-shape space in this work.

For Riemannian flow matching on a general pre-shape space $\mathcal{S}_m^k$, for any two pre-shapes $\boldsymbol{X}_0, \boldsymbol{X}_1 \in \mathcal{S}_m^k$, define

$$\theta = \arccos \operatorname{trace}(\boldsymbol{X}_0^\top \boldsymbol{X}_1), \qquad 0 \le \theta \le \pi.$$

If $0 < \theta < \pi$, the unique minimizing geodesic connecting $\boldsymbol{X}_0$ and $\boldsymbol{X}_1$ on $\mathcal{S}_m^k$ is given by

$$\Gamma(t) = \frac{\sin((1-t)\theta)}{\sin\theta}\,\boldsymbol{X}_0 + \frac{\sin(t\theta)}{\sin\theta}\,\boldsymbol{X}_1, \qquad t \in [0,1].$$

and the geodesic velocity at time t is given by

$$\dot{\Gamma}(t) = \frac{\theta}{\sin\theta}\left(-\cos((1-t)\theta)\boldsymbol{X}_0 + \cos(t\theta)\boldsymbol{X}_1\right).$$

When $\boldsymbol{X}_0 = \boldsymbol{X}_1$, the unique minimizing geodesic is constant. When $\boldsymbol{X}_0 = -\boldsymbol{X}_1$, minimizing geodesics are not unique.

E Theoretical Support

E.1 Redundancy induces flat directions

Proposition 1 (Redundancy induces flat directions)**.** *Let $\mathcal{M}$ be a smooth m-dimensional manifold and let $\pi : U \subset \mathbb{R}^d \to \mathcal{M}$ be a smooth submersion with $d > m$. For any loss $L(z) = \ell(\pi(z))$, the gradient $\nabla L(z)$ is orthogonal to $\ker(D\pi(z))$, and the Hessian $\nabla^2 L(z)$ has at least $d - m$ zero eigenvalues at any critical point. In other words, representing an m-dimensional quantity in $\mathbb{R}^d$ introduces $d - m$ flat directions in the loss landscape.*

Proof. **Gradient claim.** By the chain rule, the differential of L at z satisfies

$$\mathrm{d}L_z = \mathrm{d}\ell_{\pi(z)} \circ D\pi(z).$$

For any $v \in \ker(D\pi(z))$, we have $D\pi(z)v = 0$, so

$$\nabla L(z)^\top v = \mathrm{d}L_z(v) = \mathrm{d}\ell_{\pi(z)}(D\pi(z)v) = \mathrm{d}\ell_{\pi(z)}(0) = 0.$$

Hence $\nabla L(z)$ is orthogonal to every redundant direction.

Hessian claim. At a critical point of L, the chain rule gives

$$D\pi(z)^* \mathrm{d}\ell_{\pi(z)} = 0.$$

Because π is a submersion, $D\pi(z)$ is surjective and its adjoint is injective. Hence $\mathrm{d}\ell_{\pi(z)} = 0$. At such a point, the Hessian chain rule reduces to

$$\nabla^2 L(z) = D\pi(z)^* \operatorname{Hess} \ell(\pi(z)) D\pi(z).$$

Therefore, for every $v \in \ker(D\pi(z))$, we have $\nabla^2 L(z)v = 0$.

Zero eigenvalue count. Since π is a submersion, $\operatorname{rank} D\pi(z) = m$, and rank-nullity gives $\dim \ker(D\pi(z)) = d - m$. Thus the null space of $\nabla^2 L(z)$ has dimension at least $d - m$, so $\nabla^2 L(z)$ has at least $d - m$ zero eigenvalues. □

This proposition shows that when a representation contains redundant coordinates, the loss landscape contains intrinsically flat directions. These directions correspond to variations that do not change the underlying state on the manifold, which may lead to an ill-posed optimization problem and unstable training dynamics. On the contrast, a compact representation aligned with the intrinsic manifold removes these redundant degrees of freedom.

E.2 Intrinsic statistical complexity of Riemannian Flow Matching

Let $(\mathcal{M}, g)$ be a compact d-dimensional Riemannian manifold embedded in $\mathbb{R}^D$, and suppose the interpolation satisfies $X_t \in \mathcal{M}$ and $\dot{X}_t \in T_{X_t}\mathcal{M}$. Define

$$\rho_t = \mathrm{Law}(X_t), \qquad u_t^\star(x) = \mathbb{E}[\dot{X}_t \mid X_t = x].$$

The flow matching population loss satisfies the regression identity

$$\mathcal{L}(u) - \mathcal{L}(u^\star) = \int_0^1 \|u_t - u_t^\star\|^2_{L^2(\rho_t; T\mathcal{M})} \,\mathrm{d}t.$$

Thus learning the velocity field is a regression problem whose covariates (t, X_t) lie on $[0,1] \times \mathcal{M}$ rather than filling $[0,1] \times \mathbb{R}^D$.

For example, under the usual regularity assumptions for nonparametric regression on a compact manifold—including an s-smooth regression function jointly in (t, x), a regular design density, controlled noise, and an estimator matched to this smoothness class—the canonical integrated squared-error rate is

$$n^{-\frac{2s}{2s+d+1}}.$$

Different assumptions on the time dependence, such as a fixed finite-dimensional time parameterization, can remove the additional time dimension and yield a rate involving d alone.

This intrinsic-dimension argument does not by itself imply that every ambient Euclidean flow-matching estimator must incur a rate depending on D. Because its inputs are also supported on $\mathcal{M}$, an ambient estimator that adapts to the support geometry may attain an intrinsic-dimension rate. The benefit of the Riemannian formulation is therefore a direct geometric inductive bias: it restricts predictions to the tangent bundle and builds the known support geometry into interpolation and integration, instead of requiring the estimator to learn these constraints from data.

Under additional flow-stability conditions, vector-field estimation error also controls terminal distribution error. In particular, if the true and estimated vector fields generate unique flows, are uniformly Lipschitz in space, and satisfy a stability inequality of the form

$$W_2^2(\hat{\rho}_1, \rho_1) \lesssim \int_0^1 \|\hat{u}_t - u_t^\star\|^2_{L^2(\rho_t; T\mathcal{M})} \,\mathrm{d}t,$$

then any valid regression rate transfers to the terminal Wasserstein error. This observation motivates the Riemannian parameterization but does not, without specifying estimators and function classes, establish a universal statistical separation from Euclidean Flow Matching.

F Additional Experiment Results

F.1 Full Results

We report the full results of our method and all baselines on HumanML3D in both the original H3D format (Table 4) and the MotionStreamer format (Table 5), together with the full results on MotionMillion (Table 6). For HumanML3D, each experiment is repeated over 20 independent runs, and we

Table 4: Full results on the HumanML3D dataset in H3D format.

Method	HumanML3D Format						
	FID↓	R@1↑	R@2↑	R@3↑	MM-Dist↓	Div→	MM↑
GT	0.002	0.511	0.703	0.797	2.974	9.503	2.799
MLD	0.473	0.481	0.673	0.772	3.196	9.724	2.413
T2M-GPT	0.116	0.492	0.679	0.775	3.121	9.761	1.856
MotionGPT	0.232	0.492	0.681	0.733	3.096	**9.528**	2.008
MoMask	0.045	0.521	0.713	0.807	2.958	–	1.241
MotionGPT-2	0.191	0.496	0.691	0.782	3.080	9.860	2.137
MotionLCM	0.304	0.505	0.705	0.805	2.986	9.607	2.259
MotionCLR	0.269	**0.542**	**0.732**	**0.831**	**2.806**	9.607	1.985
MotionLab	0.167	–	–	0.810	2.830	9.593	**2.912**
MARDM	0.114	0.500	0.695	0.795	–	–	2.231
Ours	**0.043**$^{\pm.002}$	0.525$^{\pm.002}$	0.711$^{\pm.002}$	0.805$^{\pm.002}$	2.930$^{\pm.007}$	9.555$^{\pm.060}$	2.748$^{\pm.023}$

Table 5: Full results on the HumanML3D dataset in MotionStreamer format.

Method	MotionStreamer Format						
	FID↓	R@1↑	R@2↑	R@3↑	MM-Dist↓	Div→	MM↑
GT	0.002	0.702	0.864	0.914	15.151	27.492	–
MLD	18.236	0.546	0.730	0.792	16.638	26.352	–
T2M-GPT	12.475	0.606	0.774	0.838	16.812	27.275	–
MotionGPT	14.375	0.456	0.598	0.628	17.892	27.114	–
MoMask	12.232	0.621	0.784	0.846	16.138	27.127	–
MotionStreamer	11.790	0.631	0.802	0.859	16.081	27.284	–
Ours	**5.835**$^{\pm.060}$	**0.710**$^{\pm.002}$	**0.850**$^{\pm.002}$	**0.899**$^{\pm.001}$	**15.683**$^{\pm.022}$	**27.672**$^{\pm.076}$	**14.906**$^{\pm.208}$

Table 6: Full results of MotionMillion dataset.

Method	Guidance Scale	FID↓	R@1↑	R@2↑	R@3↑
ScaMo	–	89.0	0.67	0.81	0.87
MotionMillion-3B	–	10.8	0.79	0.91	0.94
MotionMillion-7B	–	10.3	0.79	0.90	0.94
Ours (0.5B+1.7B)	2.0	**5.6**	0.81	0.92	0.95
Ours (0.5B+1.7B)	3.0	7.8	**0.86**	**0.95**	**0.97**

report the mean with 95% confidence intervals computed using the t-distribution. For MotionMillion, the dataset scale and our compute budget limit evaluation to a single run.

F.2 Training Stability

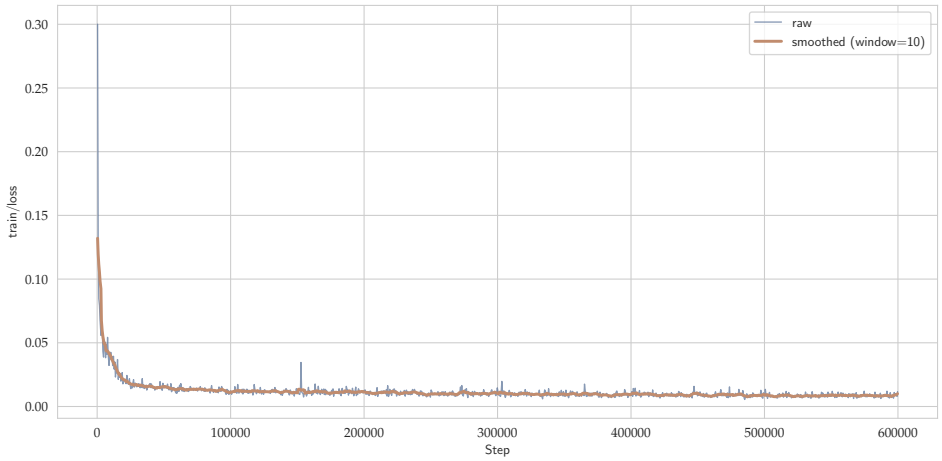


Figure 4: Training loss curves for MotionMillion.

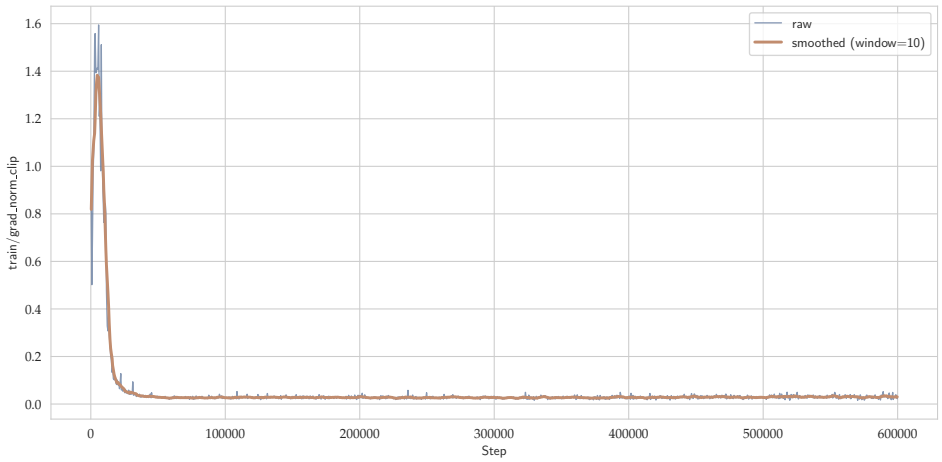


Figure 5: Gradient norm curves for MotionMillion.

We monitor the training loss and gradient norm curves for our MotionMillion experiments, as shown in Figure 4 and Figure 5. The loss curve shows a smooth and steady decrease throughout training. The gradient norm curve also remains stable with only one spike in the beginning of training, and that's why we adopt a warm-up learning rate strategy and a single gradient clipping threshold for the entire training process. And at the end of training, both the loss and gradient norm curves are still stable, which suggests that the training process is well-behaved and does not diverge or collapse.

G Implementation Details

G.1 Model Architecture

Our model uses a Diffusion Transformer backbone, while the text encoder varies across datasets.

For HumanML3D, we use Qwen3-Embedding-0.6B (Zhang et al., 2025) to extract a 1024-dimensional text feature vector. We fuse the text features with the time embedding through an MLP and use the fused representation as the conditioning input to the Diffusion Transformer.

For MotionMillion, we use Qwen3-1.7B (Yang et al., 2025) to extract text features. Unlike Qwen3-Embedding-0.6B, Qwen3-1.7B is a decoder-only large language model. Following recent advances in image generation (Cai et al., 2025; Esser et al., 2024), we therefore adopt a single-stream multi-modal Diffusion Transformer (MM-DiT), where text and motion tokens are concatenated along the sequence dimension and processed as a unified input stream.

G.2 Training Details

Table 7: Training hyperparameters for different model scales.

Hyperparameter	**RMG-base**	**RMG**
Input Dim.	91	91
Hidden Dim.	384	1024
No. of Layers	6	24
No. of Heads	8	8
FFN Multiplier	8	4
Max Learning Rate	1e-4	1e-4
Learning Rate Scheduler	Cosine w/ warmup	Cosine w/ warmup
Ratio of Warmup Steps	0.08	0.08
Effective Batch Size	$32 \times 8 \times 1$	$16 \times 8 \times 2$
Training Steps	150k	600k
Gradient Clipping	0.5	0.5

We list the training hyperparameters for different model scales in Table 7. The effective batch size is calculated as the product of the per-device batch size, the number of devices, and the gradient accumulation steps.

G.3 Conversion Functions

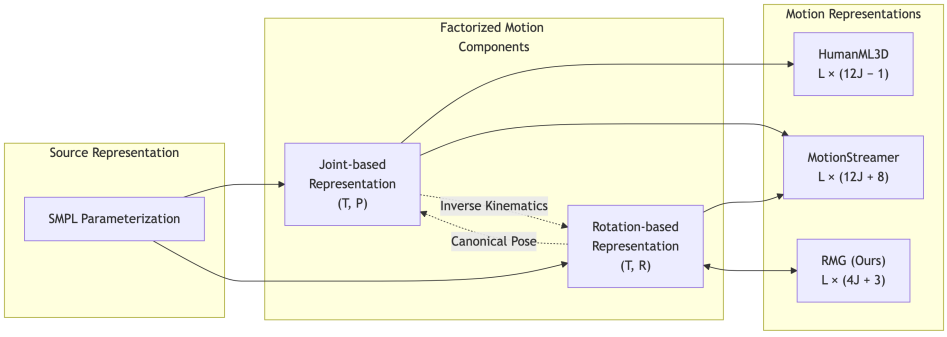


Figure 6: The conversion functions between different motion representations.

We implement two extra conversion functions to convert our representation of human motion to HumanML3D format and MotionMillion format, respectively. The core idea is to first map our representation back to either a rotation-based or a joint-based representation, and then apply the same preprocessing pipeline used by the target format. Figure 6 summarizes these conversion functions.

H LIMITATIONS

While our proposed Riemannian Motion Generation framework demonstrates promising results in motion representation and generation, there are several limitations that warrant discussion:

- The Riemannian representation has not been tested on the auto-regressive setting though it's been a popular stream in recent human motion generation works.
- More conditioning modalities (e.g., music, video) and interactive generation scenarios (e.g., human-object interaction) have not been explored.
- The generation duration is currently limited to 10 seconds (300 frames), and scaling to longer horizons may require larger data collection efforts and more computational resources.
- More body configurations (e.g., hands and face) are not included in the current framework, and extending to these richer configurations may require additional design considerations.
- Motion editing and temporal in-painting are not studied in this work, and it remains to be seen how the Riemannian representation can be adapted to these tasks.