FLOWCOMOTION: TEXT-TO-MOTION GENERATION VIA TOKEN–LATENT FLOW MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT

Text-to-motion generation is driven by learning motion representations for semantic alignment with language. Existing methods rely on either continuous or discrete motion representations. However, continuous representations entangle semantics with dynamics, while discrete representations lose fine-grained motion details. In this context, we propose **FlowCoMotion**, a novel motion generation framework that unifies both treatments from a modeling perspective. Specifically, FlowCoMotion employs token-latent coupling to capture both semantic content and high-fidelity motion details. In the latent branch, we apply multi-view distillation to regularize the continuous latent space, while in the token branch we use discrete temporal resolution quantization to extract high-level semantic cues. The motion latent is then obtained by combining the representations from the two branches through a token-latent coupling network. Subsequently, a velocity field is predicted based on the textual conditions. An ODE solver integrates this velocity field from a simple prior, thereby guiding the sample to the potential state of the target motion. Extensive experiments show that FlowCoMotion achieves competitive performance on text-to-motion benchmarks, including HumanML3D and SnapMoGen.

1 INTRODUCTION

Text-to-motion generation, which diverse, temporally coherent human motions from natural language descriptions, has experienced notable progress in computer vision, graphics, and robotics. These advances have been enabled by large-scale text–motion datasets (Guo et al., 2022a; 2025; Fan et al., 2025) and a variety of deep generative models, including VAEs (Guo et al., 2022a; Petrovich et al., 2022), autoregressive models (Lu et al., 2025; Huang et al., 2025; Zhang et al., 2023a; Jiang et al., 2023), diffusion models (Tevet et al., 2022; Bian et al., 2025; Zhang et al., 2023b; Hosseyni et al., 2024; Zhu et al., 2025; Huang et al., 2024), and mask generative models (Guo et al., 2024; 2025). However, existing models exhibit substantial limitations when handling complex cues. They often struggle to achieve **fine-grained control** over motion extent and intensity (e.g.,"*clockwise half circle*" or "*ninety degrees*"), and to capture subtle variations in human motion. Autoregressive models use vector quantization, which introduces approximation errors, thus limiting the quality of motion generation. In autoregressive generation, tokens at each step is generated conditionally on previously generated context, and accumulate errors over time. While diffusion models can mitigate approximation error, it typically requires much denoising steps while inferring, making long-horizon motion synthesis computationally expensive and latency-intensive. Moreover, repeatedly refining high-dimensional trajectories can lead to temporal drift, so achieving high-fidelity often incurs substantial sampling cost.

Based on these observations, we propose a novel framework called **FlowCoMotion** which can perform high-quality text-motion generation and understand precise text. FlowCoMotion combines discrete vectors obtained through residual vector quantization (RVQ) with continuous vectors and performs conditional generation via the flow matching strategy. Our framework is based on two components. First, we introduce a VAE with two branches, Token and Latent, to establish an accurate mapping between 3D motion and its corresponding token-latent coupling vector. Unlike previous methods that typically use hierarchical RVQ alone for iterative residual quantization or latent vector representation, our VAE design combines two distinct representations: the discrete tokens and the continuous latent, which are generated from the token and latent branch, respectively. Second, the Transformer-based generation head parameterizes a text-conditional velocity field that

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.



Figure 1: We propose a novel framework called **FlowCoMotion**, which can handle fine-grained semantic content to guide the generation of high-quality motion. This method is particularly suitable for handling various forms of input, including but not limited to numbers, shapes and directions.

transfers samples from a simple prior distribution to the target motion latent by flow matching. While diffusion methods also start with noise, they typically require multiple denoising iterations during the inference phase, increasing computational cost and potentially leading to long-term time drift accumulation. Therefore, an ODE solver integrates the predicted velocity field, mapping the prior to the conditional motion latent. Since the model predicts sequence-level dynamics rather than subsequent autoregressive updates, it leverages a full-context attention mechanism to better maintain long-range consistency while avoiding the multi-step sampling of diffusion methods.

Our main contributions can be summarized into three parts: (a). We propose FlowCoMotion, a framework for text-motion generation that improves fine-grained text alignment. Our method shows enhanced performance on text descriptions containing numeric values, directional cues, and geometric shapes. (b). We introduce the token-latent coupling method to simultaneously guarantee the semantic content and dynamic features of motion representation. Besides, we also introduce a flow matching generation strategy for accurate and efficient generation. (c). Experimental results demonstrate that our FlowCoMotion achieves competitive performance in text-to-motion generation tasks. On the HumanML3D dataset, our R-precision values reach 0.55, 0.74, and 0.83, respectively.

2 Related Work

Text-to-Motion Generation. Recently, text-motion generation has made significant progress, improving the realism of motion and its alignment with text description. Early methods used variational autoencoders to model motion (Guo et al., 2022a; Petrovich et al., 2022). Vector quantization represents motion into discrete tokens, which can be generated using autoregressive methods (Lu et al., 2025; Huang et al., 2025; Zhang et al., 2023a; Jiang et al., 2023; Shi et al., 2025) or mask-based methods (Guo et al., 2024; 2025). Meanwhile, the diffusion methods (Tevet et al., 2022; Zhang et al., 2024; Chen et al., 2023; Zhang et al., 2023b) denoise the motion samples iteratively, making them better match the text description. Furthermore, some researchers have achieved more advanced semantic alignment capabilities using autoregressive diffusion models (Zhu et al., 2025). Despite these advances, most work still relies on continuous or discrete representations, making it difficult to simultaneously utilize labeled semantic structure and continuous temporal dynamics. It is precisely because of this gap that we propose a labeled-latent coupling method.

Hybrid Representations in Generative Modeling. High-fidelity visual synthesis increasingly leverages the synergy between discrete and continuous representations. In image generation, VQ-GAN (Esser et al., 2021) established a powerful hybrid paradigm: using codebooks to compress structural priors while performing continuous optimization in latent space. Similarly, video models like MAGVIT (Yu et al., 2023) and VideoLDM (Blattmann et al., 2023) employ 3D tokens for temporal consistency alongside continuous transitions for inter-frame smoothness. By bridging discrete semantic synthesis and continuous texture synthesis, these frameworks alleviate high-dimensional complexity while preserving fine perceptual details. OneVAE and VAEVQ use continuous representations to improve discrete visual tokenization, whereas DisCo-Diff conditions continuous diffusion on separately generated discrete latents (Yang et al., 2025; Xu et al., 2024). FlowCoMotion instead generates time-aligned RVQ and continuous motion features with a single text-conditioned flow.

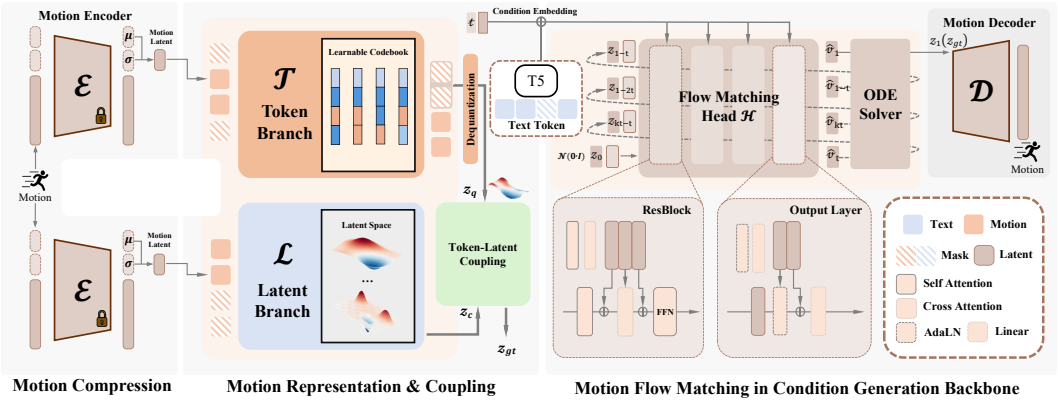


Figure 2: **Method Overview.** FlowCoMotion mainly includes three key components: (1) Motion compression (Section 3.1): The motion sequence is compressed into the latent space using VAE to obtain $z \in \mathbb{R}^{n \times b}$. (2) Motion representation & Coupling (Sections 3.2 and 3.3): The Token Branch further uses RVQ-VAE to extract discrete Token representations and then dequantizes them to obtain z_q. The Latent Branch directly uses the continuous latent space vector z_c. Finally, the final motion representation z_{gt} is obtained through the token-latent coupling network. (3) Motion Flow Matching in Condition Generation (Section 3.4): The flow velocity field $\hat{v}_t$ is predicted through the text condition c and time t. The predicted latent vector $\hat{z}_1$ is solved by the ODE solver and finally handed over to the Decoder $\mathcal{D}$.

3 Methodology

We propose **FlowCoMotion**, a framework for efficient text-to-motion generation. As shown in Figure 2, FlowCoMotion consists of three components: (1) Motion Compression (Section 3.1): A shared VAE encoder $\mathcal{E}$ maps motion sequences to a compact latent space z. (2) Motion Representation & Coupling (Sections 3.2 and 3.3): The latent z is processed by two parallel branches, a Token Branch that applies RVQ quantization to obtain discrete semantic tokens z_q, and a Latent Branch that preserves continuous dynamics z_c. A coupling network $\mathcal{C}$ fuses (z_q, z_c) into a unified representation z_{gt}. (3) Motion Flow Matching (Section 3.4): A text-conditional flow matching head $\mathcal{H}$ predicts the velocity field that maps noise to z_{gt}. We use a VAE with encoder $\mathcal{E}$ and decoder $\mathcal{D}$ to map an M-frame motion sequence $m^{1:M}$ to a compact continuous latent $z \in \mathbb{R}^d$, and reconstruct it via $\mathcal{D}$. Given a text input with N tokens $w^{1:N}$ which can be mapped into condition C, FlowCoMotion samples a noise latent conditioned on C and applies the flow matching head $\mathcal{H}$ to predict the motion latent $\hat{z}$. The synthesized motion is then obtained by decoding $\hat{m}^{1:M} = \mathcal{D}(\hat{z})$.

3.1 Motion Compression

We employ a VAE to compress motion into a continuous latent space. Our transformer-based VAE (Guo et al., 2024), comprising an encoder $\mathcal{E}$ and decoder $\mathcal{D}$ with long-stage skip connections, encodes the motion sequence into a compressed, information-rich latent space, preserving fine-grained dynamics and kinematic trends, and supporting high-fidelity, diverse motion generation. For an input motion sequence $x^{1:L}$ of arbitrary length L, the encoder processes the motion vector and learnable distribution labels to infer a Gaussian distribution with mean μ and standard deviation σ.

3.2 Motion Representation in Discrete Token

Accurately representing motion sequences is essential for motion generation tasks. We propose a **dual-representation** framework that combines token and latent representations, encoding motion sequences through two branches: one for discrete tokens and the other for continuous latent space to capture temporal details. Discretization inherently leads to information loss, reducing expressive power during quantization. To mitigate this, our framework integrates continuous latent space and discrete representations, enabling the simultaneous capture of subtle temporal variations and semantic content in motions.

For the **Token Branch**, we treat the token distribution as semantic anchors, performing quantization and dequantization. To represent motion semantics with discrete tokens, we train a motion quantizer

$\mathcal{T}$ based on the VQ-VAE architecture, as applied in (Zhang et al., 2023a; Jiang et al., 2023; Guo et al., 2024; Xiao et al., 2025). These approaches enable efficient condensation of motion semantics, facilitating fusion with continuous temporal details for motion generation tasks.

To enable the Token Branch as a semantic anchor, we employ an RVQ-VAE quantizer architecture. Multiple quantization layers are introduced to perform quantization at different temporal resolutions, with all layers sharing a single codebook (Guo et al., 2025). For a motion latent feature sequence $z \in \mathbb{R}^{n \times d}$, the quantizer applies a series of residual quantization operations with progressively increasing temporal resolutions $\{h^v\}_{v=0}^{V}$, where $h^0 < \cdots < h^V = n$. The quantized features $z_q \in \mathbb{R}^{h_0 \times d}$ at the coarse-grained level are restored to full resolution via bilinear interpolation: $\hat{f}_{\uparrow}^{v} = \mathcal{I}(f^v, h^v)$. Residuals are computed and downsampled to the next scale for quantization by subsequent layers.

$$\hat{z}_q = \sum_{v=0}^{V} \mathcal{I}(\hat{f}^v, h^v) \tag{1}$$

Overall, this Quantizer trained using latent embedding loss (commit loss) on each quantization scale $h \in \{h_0, \ldots, h_V\}$:

$$\mathcal{L}_{\text{commit}}^{(h)} = \frac{1}{|\mathcal{M}_h|} \sum_{m \in \mathcal{M}_h} \left\| z_e^{(m,h)} - \mathbf{e}_{k^{(m,h)}} \right\|_2^2 \tag{2}$$

Where $z_e^{(m,h)}$ represents the continuous latent variable of the encoder at scale h for motion sample m, and $\mathbf{e}_{k^{(m,h)}}$ is the selected vector from the codebook, determined by nearest neighbor quantization. $\mathcal{M}_h$ denotes the set of valid locations at scale h, and $|\mathcal{M}_h|$ is the number of its elements. By employing temporal resolution sampling, our Token Branch effectively samples semantic anchors at different time intervals, optimizing its fidelity and versatility in motion generation tasks.

3.3 Motion Representation in Continuous Latent

In the **Latent Branch**, a latent vector $z \in \mathbb{R}^m$ is extracted from the distribution using a reparameterization technique. This latent vector is fused with the Token Branch output and fed into the decoder $\mathcal{D}$ to generate the reconstructed motion sequence. The reconstruction training process is detailed in Section 3.5.

Multi-view Distillation. Relying solely on reconstruction and quantization constraints may cause latent space inconsistencies under varying pruning/enhancement conditions, reducing generalization for downstream generation. To address this, we propose a multi-view teacher-student self-distillation framework (Caron et al., 2021; Zhang et al., 2019), constructing multi-view set v for each sequence. The teacher processes the global views v_g, while the student processes v. After projection heads g_t and g_s and decentralization, we obtain the probability distribution.

$$\mathbf{p}_t(v) = \text{Softmax}\left(\frac{g_t(\mathcal{E}_t(v)) - \mathbf{c}}{\tau_t}\right) \quad \mathbf{p}_s(v) = \text{Softmax}\left(\frac{g_s(\phi(\mathcal{E}_s(v)))}{\tau_s}\right) \tag{3}$$

Where $\phi(\cdot)$ denotes the polymerization operator for distillation, and $\mathbf{c}$ represents the center vector, updated using exponential moving average to prevent characterization collapse. The distillation loss Hinton et al. (2015) is computed on a global view using the KL divergence between two probability distributions. The final latent sequence approximation $\hat{z}_q$ is the sum of all up-interpolated quantized sequences, which is passed through a **token-latent coupling network** $\mathcal{C}$ with the latent branch prediction $\hat{z}_c$, and subsequently fed into the Decoder $\mathcal{D}$ to predict the motion sequence.

$$\hat{\mathbf{m}} = \mathcal{D}(\mathcal{C}(\hat{z}_q, \hat{z}_c)) \tag{4}$$

Through multi-view distillation, our Latent Branch is effectively characterized, enhancing its capacity to capture motion details and optimize motion generation performance. Appendix Section B gives a representation-level interpretation of token-latent coupling.

3.4 Motion Flow Modeling in Condition Generation

To improve fidelity and diversity in motion generation, we propose a text-conditioned flow matching strategy that maps random noise to a token-latent coupled latent space. Building on recent advances

in flow matching generative models, our approach integrates Rectified Flow (Lipman et al., 2023; Hertrich et al., 2025) with a conditional Transformer backbone. The flow matching head predicts the velocity field in the motion latent space with high accuracy, introducing minimal inference overhead.

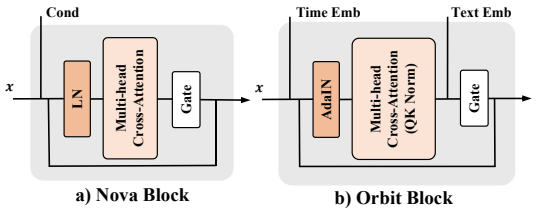


Figure 3: **Two Conditional Transformer Blocks for Motion Generation.** (a) The **Nova** Block integrates conditioning via LN-modulated cross-attention and uses a learnable gate to control the conditioned update to hidden state x. (b) The **Orbit** Block separates conditioning into time and text, performing cross-attention with QK normalization for stability. It uses the same gate to modulate the conditioned residual.

Flow Matching Process. We apply flow matching principles to represent complex probability distributions within the motion latent space. Instead of adding Gaussian noise over discrete time steps as in diffusion, we construct a continuous probability path connecting a simple prior to the real data latent variables over time $t \in [0, 1]$, resulting in a sequence of intermediate states z_t. Given a real latent variable z_1 from encoding real motions and a random sample $z_0 \sim \mathcal{N}(0, I)$ from a standard Gaussian distribution, we generate intermediate samples via a simple linear path.

$$z_t = tz_1 + (1 - t)z_0 \tag{5}$$

t controls the interpolation position. Unlike Diffusion process, flow matching requires specifying a target instantaneous velocity for each z_t, which can be viewed as a shortest straight line in the path.

$$u_t = \frac{dz_t}{dt} = z_1 - z_0 \tag{6}$$

The conditional flow model $v_\theta(z_t, t, c)$, where c is the auxiliary input, is trained to predict the instantaneous velocity u_t, learning the transformation from the prior to the data distribution over time. In our setup, c is the output hidden state of the Transformer backbone, providing conditional information that modulates the vector field. We use `T5-base` (Raffel et al., 2020) to extract word-level features from textual descriptions. As shown in Figure 3, we present two Transformers for conditional generation, each injecting distinct conditions. More model details are provided in Appendix Section E.4. The objective is based on the flow matching loss:

$$\mathcal{L} = \mathbb{E}_{z_0, z_1, t}\left[\|(z_1 - z_0) - v_\theta(z_t, t, c)\|_2^2\right] \tag{7}$$

where z_1 is the concatenated token-latent endpoint encoded from the ground-truth motion.

Inference Procedure. We begin by sampling latent variables $z_0 \sim \mathcal{N}(0, I)$ from random noise and providing conditional input c. Next, we define a continuous-time generative dynamical system using the learned conditional velocity field $v_\theta(z, t, c)$. We obtain the generated latent variables $\hat{z}_1$ by solving ordinary differential equations (ODEs) to integrate the state from $t = 0$ to $t = 1$. Finally, $\hat{z}_1$ is fed into the motion decoder $\mathcal{D}$ to reconstruct the motion sequence. For efficient conditional generation, we use a fixed-step **ODE solver** with small sampling steps to map the data distribution.

3.5 Training Procedure Overview

Fully end-to-end training can destabilize the latent space and weaken kinematic constraints. We therefore adopt a three-stage pipeline that progressively learns motion representations, stabilizes kinematic reconstruction, and enables diverse generation. As shown in Figure 4, **Motion Representation & Coupling** learns representations supporting reconstruction and generation. **Decoder Fine-tuning with Dynamic Constraints** freezes the learned motion representation and fine-tunes the Decoder to satisfy kinematic requirements. **Text-to-Motion Training with Flow Matching** trains the generation head to predict the velocity field that samples into the learned motion representation.

Motion Representation & Coupling. To map motions into the predefined latent space and achieve accurate reconstruction, we first train the VAE, Token Branch, and Latent Branch. This stage enriches

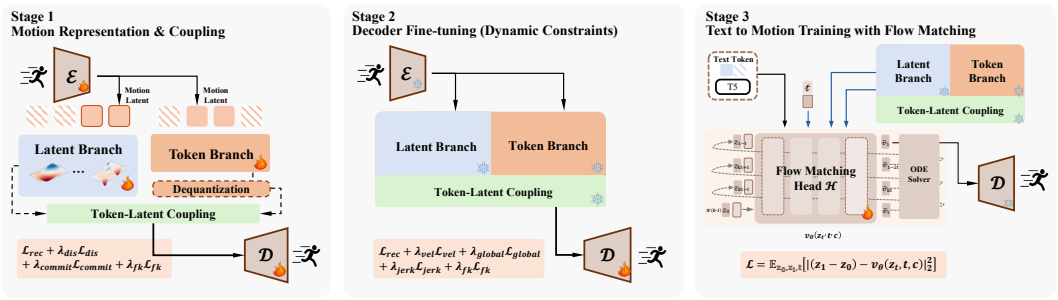


Figure 4: We propose a **three-stage training** strategy for the motion-language model: **Stage 1:** Train the Encoder $\mathcal{E}$, Decoder $\mathcal{D}$, and learnable latent space to refine latent representations. **Stage 2:** Freeze the Encoder and latent space, and fine-tune the Decoder using kinematic losses. **Stage 3:** Train the flow matching head $\mathcal{H}$ to predict the velocity flow from noise to the motion latent space. **Inference process:** Conditions are injected at each block, and the latent vector is obtained by solving the ODE through N steps from random noise. Finally, it is fed into the Decoder to obtain the predicted motion sequence.

the latent representation while minimizing kinematic constraints. Using multi-view distillation, we encourage richer mapping from motion sequences to the latent space. Following (Zhang et al., 2023a; Harvey et al., 2020), we optimize the motion tokenizer with four losses: $\mathcal{L} = \mathcal{L}_{rec} + \lambda_{dis}\mathcal{L}_{dis} + \lambda_{commit}\mathcal{L}_{commit} + \lambda_{fk}\mathcal{L}_{fk}$, where $\mathcal{L}_{rec}$ is the reconstruction loss, $\mathcal{L}_{commit}$ is the commitment loss, $\mathcal{L}_{dis}$ is the multi-view distillation loss, and $\mathcal{L}_{fk}$ is the forward kinematics (FK) loss.

Decoder Fine-tuning with Dynamic Constraints. In this stage, we freeze the encoder, token & latent branches, and introduce kinematic constraints to fine-tune the decoder. Specifically, we apply a series of kinematic constraints, including jerk loss and global loss. This stage preserves the effective motion sequence representation from the first stage, enabling the decoder to better reconstruct the motion sequence. In summary, this stage performs rigorous kinematic fine-tuning on the decoder, with a loss of $\mathcal{L} = \mathcal{L}_{rec} + \lambda_{vel}\mathcal{L}_{vel} + \lambda_{global}\mathcal{L}_{global} + \lambda_{jerk}\mathcal{L}_{jerk} + \lambda_{fk}\mathcal{L}_{fk}$, where the jerk loss $\mathcal{L}_{jerk}$, the global loss $\mathcal{L}_{global}$, the velocity loss $\mathcal{L}_{vel}$.

Text-to-Motion Training with Flow Matching. In the final stage, we encode the caption with `T5-base` and feed it to the flow matching head. We freeze all motion reconstruction networks. Given time and text, the head models the sampled path by predicting the velocity field. We train it with an `L2` loss between the predicted and ground truth velocity. This formulation leverages the sequence reconstruction capability from earlier stages while preserving inference efficiency.

4 Experiment

Extensive experiments demonstrate the superior performance of our model, **FlowCoMotion**, in motion generation tasks. Details of the dataset settings, evaluation metrics, and implementation specifics are provided in Section 4.1. We first present the performance of text-to-motion generation in Section 4.2, through both quantitative and qualitative experiments. Next, in Section 4.3, we will present more ablation experiments to demonstrate the feasibility of our method. The supplements include additional results and further implementation details.

4.1 Experimental Setup

Datasets. Our experiments are based on the HumanML3D and SnapMoGen (Guo et al., 2022a; 2025) datasets. HumanML3D is a large-scale benchmark for text-to-motion generation and understanding, containing 14,616 motion sequences from AMASS (Mahmood et al., 2019) and HumanAct12, annotated with 44,970 sequence-level natural language descriptions. SnapMoGen includes 43.7 hours of high-quality motion data at 30 frames per second, with each motion segment accompanied by 6 detailed text descriptions, including daily tasks, fitness, social interactions, and dance.

Evaluation Metrics. We evaluate motion quality using Fréchet Inception Distance (FID), which measures how closely generated motions align with ground truth motions in feature space, reflecting overall motion quality. Diversity quantifies feature space variation across generated samples, while MultiModality captures variation among motions generated from the same text description. To assess

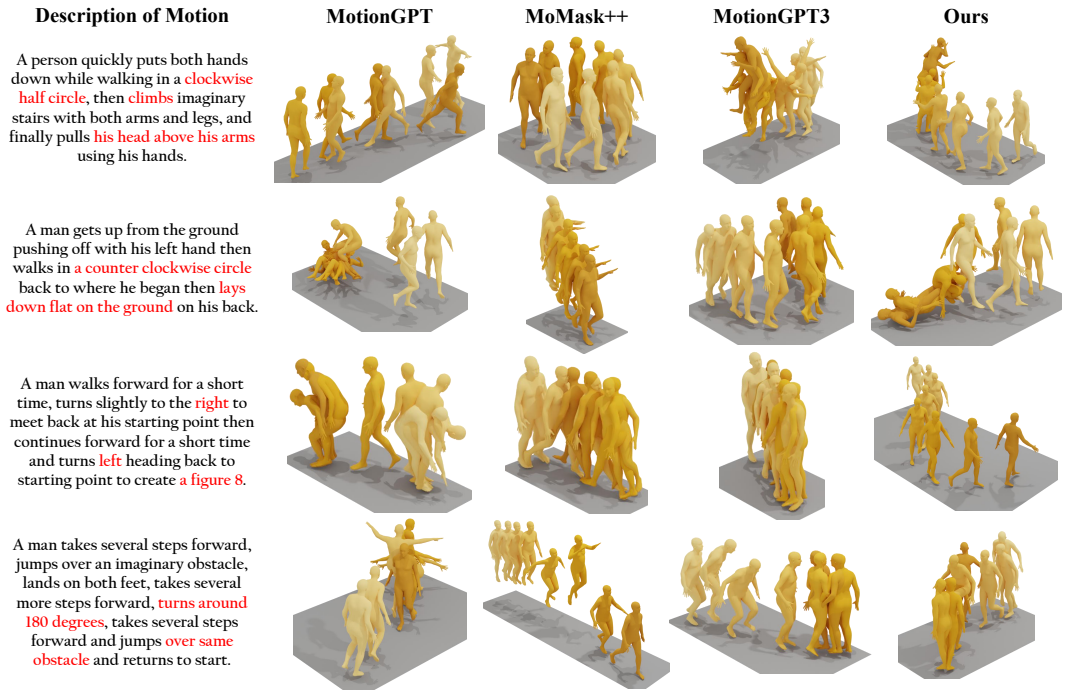


Figure 5: **Qualitative evaluation on text-to-motion generation.** We qualitatively compared the visualizations generated by our method with those produced by MotionGPT Jiang et al. (2023), MoMask++ Guo et al. (2025), and MotionGPT3 Zhu et al. (2025). Misaligned motions of other methods are marked with red text. The results show that **FlowCoMotion** demonstrates powerful text understanding capabilities, enabling it to generate more accurate and coherent motions.

text-motion alignment, we use motion-text retrieval R-Precision and Multimodal Distance, which measures the embedding-space distance between paired motion and text embeddings.

Implementation Details. FlowCoMotion consists of a motion VAE and a Transformer-based flow matching head. The VAE has 6 layers and encodes each sequence into a 256 dimension latent, split into a 64 dimension token branch and a 192 dimension continuous branch. The flow matching head uses 15 Transformer layers with 8 attention heads and a 512 hidden size. We use 40 Euler steps and the reference motion length for benchmark evaluation; practical inference accepts user-specified or externally predicted durations. More details are reported in Appendix Section E.

Table 1: **Quantitative evaluation on HumanML3D test set.**

Methods	R-precision↑			FID↓	MultiModal Dist↓	Diversity→	MModality↑
	Top 1	Top 2	Top 3				
Real motions	$0.511^{\pm.003}$	$0.703^{\pm.003}$	$0.797^{\pm.002}$	$0.002^{\pm.000}$	$2.974^{\pm.008}$	$9.503^{\pm.065}$	-
TM2D	$0.319^{\pm.000}$	-	-	$1.021^{\pm.000}$	$4.098^{\pm.000}$	$\mathbf{9.513}^{\pm.000}$	$\mathbf{4.139}^{\pm.000}$
MotionCraft	$0.501^{\pm.003}$	$0.697^{\pm.003}$	$0.796^{\pm.002}$	$0.173^{\pm.002}$	$3.025^{\pm.008}$	$9.543^{\pm.098}$	-
ReMoDiffuse	$0.510^{\pm.005}$	$0.698^{\pm.006}$	$0.795^{\pm.004}$	$0.103^{\pm.004}$	$2.974^{\pm.016}$	$9.018^{\pm.075}$	$1.795^{\pm.043}$
MMM	$0.504^{\pm.003}$	$0.696^{\pm.003}$	$0.794^{\pm.002}$	$0.080^{\pm.003}$	$2.998^{\pm.007}$	$9.411^{\pm.058}$	$1.164^{\pm.041}$
MotionGPT	$0.492^{\pm.003}$	$0.681^{\pm.003}$	$0.778^{\pm.002}$	$0.232^{\pm.008}$	$3.096^{\pm.008}$	$9.528^{\pm.071}$	$2.008^{\pm.084}$
MotionGPT-2	$0.496^{\pm.002}$	$0.691^{\pm.003}$	$0.782^{\pm.004}$	$0.191^{\pm.004}$	$3.080^{\pm.013}$	$9.860^{\pm.026}$	$2.137^{\pm.022}$
DiverseMotion	$0.515^{\pm.003}$	$0.706^{\pm.002}$	$0.802^{\pm.002}$	$0.072^{\pm.004}$	$2.941^{\pm.007}$	$9.683^{\pm.102}$	$1.869^{\pm.089}$
BAD	$0.517^{\pm.002}$	$0.713^{\pm.003}$	$0.808^{\pm.003}$	$0.065^{\pm.003}$	$2.901^{\pm.008}$	$9.694^{\pm.068}$	$1.194^{\pm.044}$
MoMask	$0.521^{\pm.002}$	$0.713^{\pm.002}$	$0.807^{\pm.002}$	$\mathbf{0.045}^{\pm.002}$	$2.958^{\pm.008}$	$9.620^{\pm.064}$	$1.241^{\pm.040}$
MoMask++	$0.528^{\pm.003}$	$0.718^{\pm.003}$	$0.811^{\pm.002}$	$0.072^{\pm.003}$	$2.912^{\pm.008}$	$9.573^{\pm.062}$	$1.227^{\pm.046}$
MotionGPT3	$\underline{0.543}^{\pm.003}$	$\underline{0.735}^{\pm.002}$	$\underline{0.828}^{\pm.002}$	$0.217^{\pm.010}$	$\mathbf{2.793}^{\pm.007}$	$9.662^{\pm.072}$	$1.366^{\pm.046}$
Ours$^{\text{nova}}$	$0.532^{\pm.002}$	$0.725^{\pm.002}$	$0.819^{\pm.002}$	$0.072^{\pm.003}$	$2.887^{\pm.006}$	$9.377^{\pm.078}$	$1.875^{\pm.088}$
Ours$^{\text{orbit}}$	$\mathbf{0.550}^{\pm.003}$	$\mathbf{0.740}^{\pm.003}$	$\mathbf{0.832}^{\pm.002}$	$\underline{0.061}^{\pm.003}$	$\underline{2.852}^{\pm.008}$	$\underline{9.457}^{\pm.067}$	$\underline{2.071}^{\pm.093}$

Table 2: **Quantitative evaluation on SnapMoGen test set.**

Methods	R-precision↑			FID↓	CLIP Score↑	MModality↑
	Top 1	Top 2	Top 3			
Real motions	$0.940^{\pm.001}$	$0.976^{\pm.001}$	$0.985^{\pm.001}$	$0.001^{\pm.000}$	$0.837^{\pm.000}$	-
MDM	$0.503^{\pm.002}$	$0.653^{\pm.002}$	$0.727^{\pm.002}$	$57.783^{\pm.092}$	$0.481^{\pm.001}$	$\mathbf{13.412}^{\pm.231}$
T2M-GPT	$0.618^{\pm.002}$	$0.773^{\pm.002}$	$0.812^{\pm.002}$	$32.629^{\pm.087}$	$0.573^{\pm.001}$	$9.172^{\pm.181}$
StableMoFusion	$0.679^{\pm.002}$	$0.823^{\pm.002}$	$0.888^{\pm.002}$	$27.801^{\pm.063}$	$0.605^{\pm.001}$	$9.064^{\pm.138}$
MARDM	$0.659^{\pm.002}$	$0.812^{\pm.002}$	$0.860^{\pm.002}$	$26.878^{\pm.131}$	$0.602^{\pm.001}$	$9.812^{\pm.287}$
MoMask	$0.777^{\pm.002}$	$0.888^{\pm.002}$	$0.927^{\pm.002}$	$17.404^{\pm.051}$	$0.664^{\pm.001}$	$8.183^{\pm.184}$
MoMask++	$\mathbf{0.805}^{\pm.002}$	$\mathbf{0.904}^{\pm.002}$	$\mathbf{0.938}^{\pm.001}$	$15.560^{\pm.071}$	$\mathbf{0.684}^{\pm.001}$	$6.556^{\pm.178}$
Ours$^{\text{nova}}$	$0.737^{\pm.002}$	$0.873^{\pm.002}$	$0.929^{\pm.002}$	$17.544^{\pm.068}$	$0.654^{\pm.001}$	$9.190^{\pm.240}$
Ours$^{\text{orbit}}$	$0.776^{\pm.002}$	$0.890^{\pm.002}$	$0.934^{\pm.001}$	$\mathbf{14.678}^{\pm.061}$	$0.672^{\pm.001}$	$8.553^{\pm.230}$

4.2 Comparison to State-of-the-art Approaches

We compared our method with a range of existing state-of-the-art works, including autoregressive models (Gong et al., 2023; Pinyoanuntapong et al., 2024; Jiang et al., 2023; Wang et al., 2024; Zhang et al., 2023a; Meng et al., 2025), diffusion models (Tevet et al., 2022; Bian et al., 2025; Zhang et al., 2023b; Hosseyni et al., 2024; Huang et al., 2024), mask generation models (Guo et al., 2024; 2025), and autoregressive diffusion model (Zhu et al., 2025).

Qualitative Comparison. Figure 5 presents a qualitative comparison between our method, MotionGPT, MoMask++, and MotionGPT3. While MotionGPT generates semantically correct motions, it struggles with direction- and degree-related descriptions, such as "clockwise half circle" and "in figure 8". Although MoMask++ and MotionGPT3 show improvements, they still fail to fully align motions with the text descriptions. Additionally, MoMask++ occasionally produces stiff and repetitive motions. In contrast, our method generates high-quality motions that are highly consistent with the input text. Further comparative results can be found in Appendix Section A.1.

Quantitative Comparisons. Following prior work, we repeat each experiment 20 times and report the mean with a 95% confidence interval. We evaluate two conditional Transformer blocks, termed **Nova** and **Orbit**. Results on HumanML3D and SnapMoGen are reported in Tables 1 and 2. FlowCoMotion achieves competitive or better performance on the core metrics, indicating robust text-to-motion generation. On HumanML3D, the **Nova** transformer already matches strong baselines, and **Orbit** further improves performance. Besides, we report a full comparison with prior T2M approaches in Appendix Section A.2.

Table 3: **Ablation analysis of motion representation on Text-to-Motion Tasks.**

	VQ Config.				VQ Reconstruction		T2M Generation	
	#Codes	#Quant.	F/M	w/ Att.	FID ↓	Joint Pos. Err. ↓	FID ↓	R-Precison@3 ↑
HumanML3D	1024	4	M	✓	0.0622	0.0488	$0.122^{\pm.002}$	$0.809^{\pm.003}$
	512	4	M	✓	0.0859	0.0514	$0.218^{\pm.003}$	$0.771^{\pm.002}$
	Representation		Dimension		VQ Reconstruction		T2M Generation	
	Type		Dim.		FID ↓	Joint Pos. Err. ↓	FID ↓	R-Precison@3 ↑
	Latent		256		0.0054	0.0302	$0.087^{\pm.004}$	$0.796^{\pm.003}$
	Hybrid		64, 192		0.0036	0.0234	$0.061^{\pm.003}$	$0.832^{\pm.002}$
	VQ Config.				VQ Reconstruction		T2M Generation	
	#Codes	#Quant.	F/M	w/ Att.	FID ↓	Joint Pos. Err. ↓	FID ↓	R-Precison@3 ↑
SnapMoGen	1024	2	M	✓	4.26	8.48	$26.248^{\pm.060}$	$0.911^{\pm.002}$
	512	2	M	✓	4.41	8.62	$28.013^{\pm.072}$	$0.903^{\pm.001}$
	Representation		Dimension		VQ Reconstruction		T2M Generation	
	Type		Dim.		FID ↓	Joint Pos. Err. ↓	FID ↓	R-Precison@3 ↑
	Latent		256		0.1650	4.3650	$19.465^{\pm.037}$	$0.895^{\pm.002}$
	Hybrid		64, 192		0.0503	4.4234	$14.678^{\pm.061}$	$0.934^{\pm.001}$

4.3 Ablation Studies

Our research proposes a new strategy for integrating discrete motion representation and continuous motion, and using a flow matching head for conditional generation. This strategy consists of two

key components: 1) Token-Latent Coupling, 2) Conditional generation using flow matching. More ablation studies are shown in Appendix Section B.

Token-Latent Coupling. To evaluate the utility of Token-Latent Coupling, we compared our model with variants of motion tokens that were discretized using vector quantization alone (*token dimension = 256*), and with variants that used only continuous motion Latents (*latent dimension = 256*). As shown in Table 3, under the same training iterations, the mixed representation significantly outperformed the other variants in both the motion reconstruction and generation tasks. The discrete-only model favors R-Precision, whereas the continuous-only model yields better reconstruction. Token-attribute associations and dimension-matched probes further show stronger semantic prediction from the token branch and stronger speed/jerk prediction from the continuous branch (Equation (16)).

Table 4: **Ablation analysis on HumanML3D test set of motion representation and generation heads.**

Representation	Dimension	VQ Reconstruction		T2M Generation		
Type	Dim.	FID ↓	Joint Pos. Err. ↓	FID ↓	R-Precison@3 ↑	Inference time ↓
Latent + Flow Matching	256	0.0054	0.0302	$0.087^{\pm.004}$	$0.796^{\pm.003}$	0.49
Hybrid + Flow Matching	64, 192	0.0036	0.0234	$0.061^{\pm.003}$	$0.832^{\pm.002}$	0.52
Latent + Diffusion	256	0.0054	0.0302	$0.078^{\pm.003}$	$0.781^{\pm.002}$	7.66
Hybrid + Diffusion	64, 192	0.0036	0.0234	$0.075^{\pm.003}$	$0.801^{\pm.002}$	8.02

Flow Modeling Generation. To assess the benefits of the flow modeling approach, we compared our model with a variant generated using the standard diffusion method. As shown in Table 4, under the same training iterations, the method using flow modeling significantly outperformed the diffusion model. Additionally, the inference speed of the flow modeling method was also significantly faster than that of the diffusion method.

Table 5: **Quantitative evaluation on harder test set which selected from HumanML3D.**

Methods	R-precision↑			FID↓
	Top 1	Top 2	Top 3	
MoMask	$0.467^{\pm.002}$	$0.655^{\pm.003}$	$0.748^{\pm.003}$	$\mathbf{0.231}^{\pm.012}$
MoMask++	$0.473^{\pm.003}$	$0.669^{\pm.003}$	$0.769^{\pm.002}$	$0.354^{\pm.015}$
MotionGPT3	$\underline{0.496}^{\pm.002}$	$\underline{0.685}^{\pm.002}$	$\underline{0.787}^{\pm.003}$	$0.658^{\pm.032}$
Ours^{nova}	$0.479^{\pm.002}$	$0.678^{\pm.003}$	$0.776^{\pm.002}$	$0.349^{\pm.014}$
Ours^{orbit}	$\mathbf{0.502}^{\pm.003}$	$\mathbf{0.693}^{\pm.004}$	$\mathbf{0.796}^{\pm.002}$	$\underline{0.332}^{\pm.011}$

4.4 Evaluation on Fine-Grained Text Understanding

To rigorously assess our model's capability in comprehending complex language instructions, we conduct an attribute-specific analysis on a constructed "Hard Test Set". Specifically, we filter the HumanML3D test set to isolate prompts containing fine-grained constraints (Wang et al., 2023; Huang et al., 2025; Jin et al., 2023), such as directional indicators (e.g., left/right), rotational semantics (e.g., clockwise/counterclockwise), and precise numerical or degree-based instructions. Details of the hard test split are provided in Appendix Section E.5. As shown in Table 5, FlowCoMotion demonstrates a distinct advantage in these constrained scenarios, achieving the best text-motion alignment on this subset. Additionally, FlowCoMotion maintains a competitive FID, significantly outperforming MoMask++ and MotionGPT3 in motion quality under strict textual conditions. These results confirm that our approach effectively balances high-fidelity motion synthesis with precise semantic grounding, addressing complex attribute binding that frequently challenges existing methods.

5 Conclusion

In Conclusion, we propose FlowCoMotion, a novel generative flow matching modeling framework for text-driven 3D human motion generation. FlowCoMotion integrates a label-latent coupling method for accurate motion representation, ensuring semantic and dynamic properties, and a flow matching generation strategy for fast generation of high-quality motion. However, some limitations remain, such as the need for further optimization in generating longer sequences. Our future work will explore zero-shot generalization capabilities.

REFERENCES

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 *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence*, 2025. doi: 10.1609/aaai.v39i2.32183.

Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jegou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9630–9640, 2021. doi: 10.1109/ICCV48922.2021.00951.

Ling-Hao Chen, Wenxun Dai, Xuan Ju, Shunlin Lu, and Lei Zhang. Motionclr: Motion generation and training-free editing via understanding attention mechanisms. *arxiv:2410.18977*, 2024a.

Ling-Hao Chen, Shunlin Lu, Ailing Zeng, Hao Zhang, Benyou Wang, Ruimao Zhang, and Lei Zhang. Motionllm: Understanding human behaviors from human motions and videos. *arXiv preprint arXiv:2405.20340*, 2024b.

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 *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18000–18010, 2023. doi: 10.1109/CVPR52729.2023.01726.

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12873–12883, June 2021.

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 (ICCV)*, pp. 13336–13348, October 2025.

Xuehao Gao, Yang Yang, Zhenyu Xie, Shaoyi Du, Zhongqian Sun, and Yang Wu. Guess: Gradually enriching synthesis for text-driven human motion generation. *IEEE Transactions on Visualization and Computer Graphics*, 30(12):7518–7530, 2024. doi: 10.1109/TVCG.2024.3352002.

Kehong Gong, Dongze Lian, Heng Chang, Chuan Guo, Zihang Jiang, Xinxin Zuo, Michael Bi Mi, and Xinchao Wang. TM2D: Bimodality Driven 3D Dance Generation via Music-Text Integration . In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9908–9918, Los Alamitos, CA, USA, October 2023. IEEE Computer Society. doi: 10.1109/ICCV51070.2023.00912.

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 *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5142–5151, 2022a. doi: 10.1109/CVPR52688.2022.00509.

Chuan Guo, Xinxin Zuo, Sen Wang, and Li Cheng. Tm2t: Stochastic and tokenized modeling for the reciprocal generation of 3d human motions and texts. In *European Conference on Computer Vision*, pp. 580–597, 2022b.

Chuan Guo, Yuxuan Mu, Muhammad Gohar Javed, Sen Wang, and Li Cheng. Momask: Generative masked modeling of 3d human motions. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1900–1910, 2024. doi: 10.1109/CVPR52733.2024.00186.

Chuan Guo, Inwoo Hwang, Jian Wang, and Bing Zhou. Snapmogen: Human motion generation from expressive texts. *arXiv preprint arXiv:2507.09122*, 2025.

Félix G. Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher Pal. Robust motion in-betweening. *ACM Trans. Graph.*, 39(4), August 2020. ISSN 0730-0301. doi: 10.1145/3386569.3392480.

Johannes Hertrich, Antonin Chambolle, and Julie Delon. On the relation between rectified flows and optimal transport. *CoRR*, abs/2505.19712, 2025. doi: 10.48550/ARXIV.2505.19712.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

S. Rohollah Hosseyni, Ali Ahmad Rahmani, S. Jamal Seyedmohammadi, Sanaz Seyedin, and Arash Mohammadi. Bad: Bidirectional auto-regressive diffusion for text-to-motion generation. *arXiv preprint arXiv:2409.10847*, 2024.

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. *arXiv preprint arXiv:2405.05691*, 2024.

Yiming Huang, Weilin Wan, Yue Yang, Chris Callison-Burch, Mark Yatskar, and Lingjie Liu. Como: Controllable motion generation through language guided pose code editing. In *European Conference on Computer Vision*, pp. 180–196, 2025.

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

Peng Jin, Yang Wu, Yanbo Fan, Zhongqian Sun, Wei Yang, and Li Yuan. Act as you wish: Fine-grained control of motion diffusion model with hierarchical semantic graphs. In *Advances in Neural Information Processing Systems*, volume 36, pp. 15497–15518, 2023.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2023.

Yunhong Lou, Linchao Zhu, Yaxiong Wang, Xiaohan Wang, and Yi Yang. Diversemotion: Towards diverse human motion generation via discrete diffusion. *arXiv preprint arXiv:2309.01372*, 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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27872–27882, 2025. doi: 10.1109/CVPR52734.2025.02595.

Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael Black. Amass: Archive of motion capture as surface shapes. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5441–5450, 2019. doi: 10.1109/ICCV.2019.00554.

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. *arXiv preprint arXiv:2411.16575*, 2025.

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, 2022.

Ekkasit Pinyoanuntapong, Pu Wang, Minwoo Lee, and Chen Chen. Mmm: Generative masked motion model. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1546–1555, 2024. doi: 10.1109/CVPR52733.2024.00153.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

Junyu Shi, Lijiang Liu, Yong Sun, Zhiyuan Zhang, Jinni Zhou, and Qiang Nie. Genm3: Generative pretrained multi-path motion model for text conditional human motion generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13129–13139, October 2025.

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.

Yin Wang, Zhiying Leng, Frederick W. B. Li, Shun-Cheng Wu, and Xiaohui Liang. Fg-t2m: Fine-grained text-driven human motion generation via diffusion model. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 21978–21987, 2023. doi: 10.1109/ICCV51070.2023.02014.

Yuan Wang, Di Huang, Yaqi Zhang, Wanli Ouyang, Jile Jiao, Xuetao Feng, Yan Zhou, Pengfei Wan, Shixiang Tang, and Dan Xu. Motiongpt-2: A general-purpose motion-language model for motion generation and understanding. *arXiv preprint arXiv:2410.21747*, 2024.

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 (ICCV)*, pp. 10086–10096, October 2025.

Yilun Xu, Gabriele Corso, Tommi Jaakkola, Arash Vahdat, and Karsten Kreis. Disco-diff: Enhancing continuous diffusion models with discrete latents. In *International Conference on Machine Learning*, 2024.

Sicheng Yang, Xing Hu, Qiang Wu, and Dawei Yang. Vaevq: Enhancing discrete visual tokenization through variational modeling. *arXiv preprint arXiv:2511.06863*, 2025.

Lijun Yu, Yong Cheng, Kihyuk Sohn, José Lezama, Han Zhang, Huiwen Chang, Alex Hauptmann, Ming-Hsuan Yang, Yuan Hao, Irfan Essa, and Lu Jiang. Magvit: Masked generative video transformer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

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 (CVPR)*, pp. 14730–14740, June 2023a.

Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3712–3721, 2019. doi: 10.1109/ICCV.2019.00381.

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

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, 2024. doi: 10.1109/TPAMI.2024.3355414.

Zeyu Zhang, Akide Liu, Ian Reid, Richard Hartley, Bohan Zhuang, and Hao Tang. Motion mamba: Efficient and long sequence motion generation. In *European Conference on Computer Vision*, pp. 265–282. Springer, 2025.

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 (ICCV)*, pp. 509–519, October 2023.

Wenyang Zhou, Zhiyang Dou, Zeyu Cao, Zhouyingcheng Liao, Jingbo Wang, Wenjia Wang, Yuan Liu, Taku Komura, Wenping Wang, and Lingjie Liu. Emdm: Efficient motion diffusion model for fast, high-quality motion generation. *arXiv preprint arXiv:2312.02256*, 2023.

Bingfan Zhu, Biao Jiang, Sunyi Wang, Shixiang Tang, Tao Chen, Linjie Luo, Youyi Zheng, and Xin Chen. Motiongpt3: Human motion as a second modality. *arXiv preprint arXiv:2506.24086*, 2025.

Qiran Zou, Shangyuan Yuan, Shian Du, Yu Wang, Chang Liu, Yi Xu, Jie Chen, and Xiangyang Ji. Parco: Part-coordinating text-to-motion synthesis. *arXiv preprint arXiv:2403.18512*, 2024.

APPENDIX

The appendix is organized as follows. Section A provides additional qualitative and quantitative comparisons on HumanML3D. Section B reports ablations on ODE steps, classifier-free guidance, inference efficiency, and the token/latent dimensional split. Section C describes the human-evaluation protocol and results. Section D presents the representation-level analysis of token-latent coupling and conditional flow matching. Section E gives the remaining implementation details, including the training objectives, codebook and temporal-resolution settings, flow-matching formulation, model parameterization, and the construction of Harder-HumanML3D.

LLM Usage. We employ large language models (LLMs), such as ChatGPT, to assist in polishing our paper. Specifically, LLMs are used to correct grammatical errors, refine word choice, and improve overall fluency. We do not use LLMs to formulate our methodology or run experiments.

A MORE RESULTS

A.1 MORE QUALITATIVE RESULTS

We supplement some qualitative results of the text-to-motion comparison on HumanML3D, dividing them into shorter and longer prompts, as shown in Figure A and Figure B respectively.

For longer prompts, while MotionGPT produces semantically plausible basic motions, it struggles significantly with extended descriptions that require a series of distinct actions (e.g., knee down" $\rightarrow$ raising right arm up" $\rightarrow$ "placing its forearm in front of the face"). Although MoMask++ and MotionGPT3 show improved sequence handling, they still fail to achieve precise text-motion alignment, frequently suffering from incorrect action ordering or the insertion of irrelevant movements. Conversely, our approach excels at interpreting complex, multi-action prompts, delivering high-quality motion sequences that strictly adhere to the textual conditions.

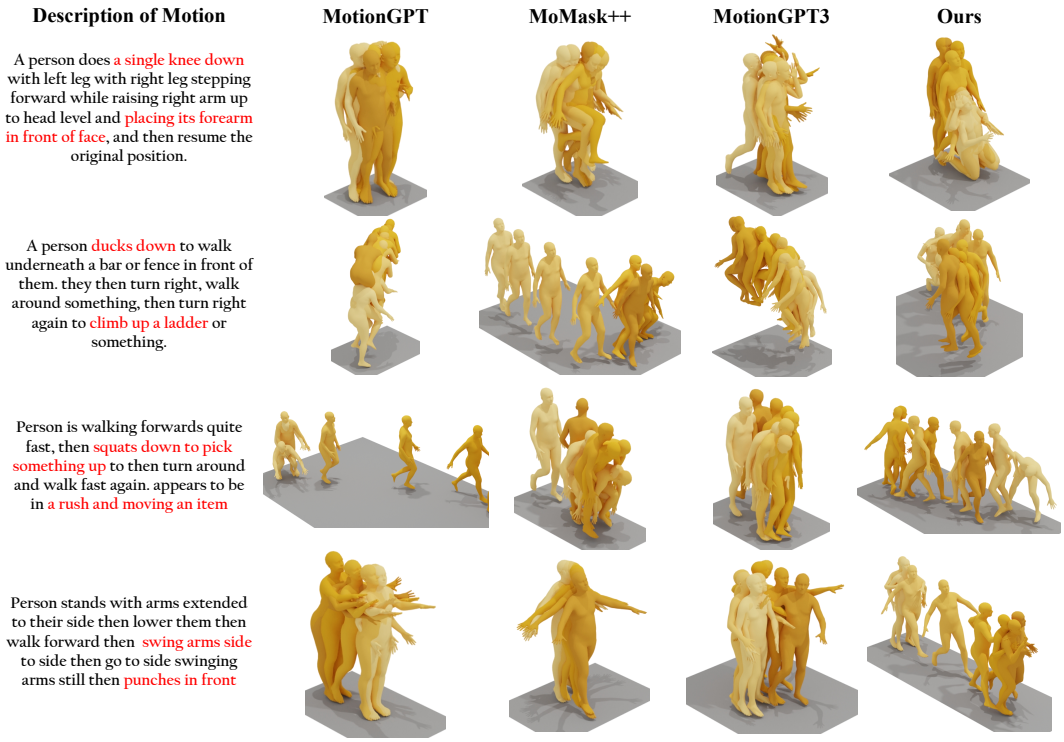


Figure A: **Qualitative evaluation on text-to-motion generation.** The provided state-of-the-art methods are under the same training and inference setting on HumanML3D. Misaligned motions are indicated by red texts. We selected some results with longer prompts. The results indicate that our motion-language framework exhibits strong text comprehension capabilities, leading to more accurate and coherent motion generation.

For shorter prompts, MotionGPT generally generates semantically correct motions, but it struggles with direction- and plural-related descriptions (e.g., “right side”, “arms” and “left/right arm”). While MoMask++ and MotionGPT3 improve in this regard, they still struggle to generate motions that perfectly match the text descriptions. Furthermore, the motions generated by MoMask++ are largely identical and lack dynamism. In contrast, our method is able to generate high-quality motions that are highly consistent with the input text.

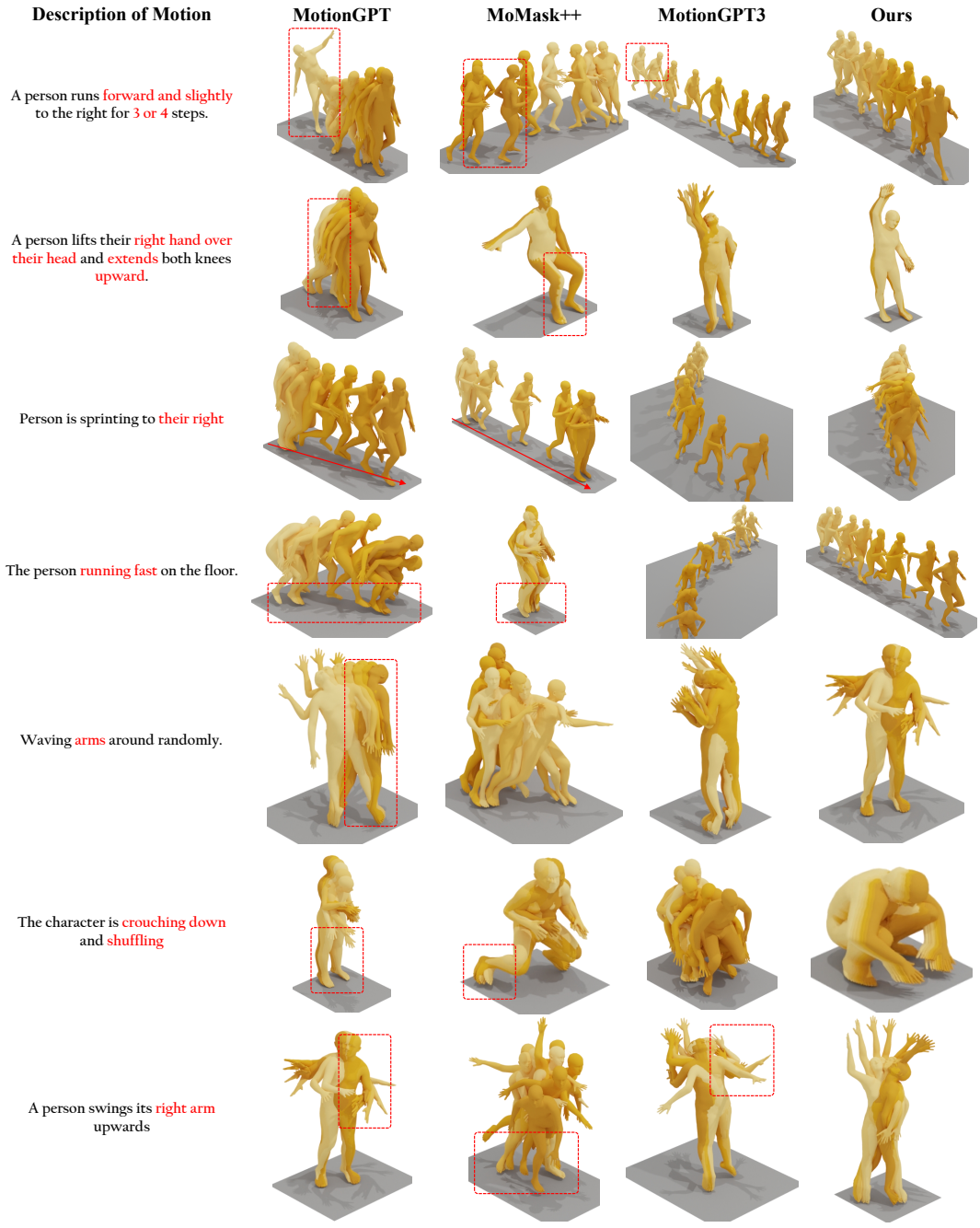


Figure B: **Qualitative evaluation on text-to-motion generation.** The state-of-the-art methods presented in this paper are all implemented using the same inference and rendering settings on HumanML3D. Misaligned motion is indicated by red text and bounding boxes. We selected some results with shorter prompts. The results demonstrate that our motion-language framework exhibits strong text understanding capabilities, enabling the generation of more accurate and coherent motion.

Table A: **Quantitative evaluation on HumanML3D test set.**

Methods	R-precision↑			FID↓	MultiModal Dist↓	Diversity→	MModality↑
	Top 1	Top 2	Top 3				
Real motions	$0.511^{\pm.003}$	$0.703^{\pm.003}$	$0.797^{\pm.002}$	$0.002^{\pm.000}$	$2.974^{\pm.008}$	$9.503^{\pm.065}$	-
TEMOS	$0.424^{\pm.002}$	$0.612^{\pm.002}$	$0.722^{\pm.002}$	$3.734^{\pm.028}$	$3.703^{\pm.008}$	$8.973^{\pm.071}$	$0.368^{\pm.018}$
TM2T	$0.424^{\pm.003}$	$0.618^{\pm.003}$	$0.729^{\pm.002}$	$1.501^{\pm.017}$	$3.467^{\pm.011}$	$8.589^{\pm.076}$	$2.424^{\pm.093}$
T2M	$0.457^{\pm.002}$	$0.639^{\pm.003}$	$0.740^{\pm.003}$	$1.067^{\pm.002}$	$3.340^{\pm.008}$	$9.188^{\pm.002}$	$2.090^{\pm.083}$
TM2D	$0.319^{\pm.000}$	-	-	$1.021^{\pm.000}$	$4.098^{\pm.000}$	$\mathbf{9.513}^{\pm.000}$	$\mathbf{4.139}^{\pm.000}$
MotionDiffuse	$0.491^{\pm.001}$	$0.681^{\pm.001}$	$0.782^{\pm.001}$	$0.630^{\pm.001}$	$3.113^{\pm.001}$	$9.410^{\pm.049}$	$1.553^{\pm.042}$
MDM	$0.320^{\pm.005}$	$0.498^{\pm.004}$	$0.611^{\pm.007}$	$0.544^{\pm.044}$	$5.566^{\pm.027}$	$9.559^{\pm.086}$	$2.799^{\pm.072}$
MotionLLM	$0.482^{\pm.004}$	$0.672^{\pm.003}$	$0.770^{\pm.002}$	$0.491^{\pm.019}$	$3.138^{\pm.010}$	$9.838^{\pm.244}$	-
MLD	$0.481^{\pm.003}$	$0.673^{\pm.003}$	$0.772^{\pm.002}$	$0.473^{\pm.013}$	$3.196^{\pm.010}$	$9.724^{\pm.082}$	$2.413^{\pm.079}$
Motion Mamba	$0.502^{\pm.003}$	$0.693^{\pm.002}$	$0.792^{\pm.002}$	$0.281^{\pm.009}$	$3.060^{\pm.058}$	$9.871^{\pm.084}$	$2.294^{\pm.058}$
MotionCraft	$0.501^{\pm.003}$	$0.697^{\pm.003}$	$0.796^{\pm.002}$	$0.173^{\pm.002}$	$3.025^{\pm.008}$	$9.543^{\pm.098}$	-
ReMoDiffuse	$0.510^{\pm.005}$	$0.698^{\pm.006}$	$0.795^{\pm.004}$	$0.103^{\pm.004}$	$2.974^{\pm.016}$	$9.018^{\pm.075}$	$1.795^{\pm.043}$
Fg-T2M	$0.492^{\pm.002}$	$0.683^{\pm.003}$	$0.783^{\pm.002}$	$0.243^{\pm.019}$	$3.109^{\pm.007}$	$9.278^{\pm.072}$	$1.614^{\pm.049}$
T2M-GPT	$0.492^{\pm.003}$	$0.679^{\pm.002}$	$0.775^{\pm.002}$	$0.141^{\pm.005}$	$3.121^{\pm.009}$	$9.722^{\pm.082}$	$1.831^{\pm.048}$
EMDM	$0.498^{\pm.007}$	$0.684^{\pm.006}$	$0.786^{\pm.006}$	$0.112^{\pm.019}$	$3.110^{\pm.027}$	$9.551^{\pm.078}$	$1.641^{\pm.078}$
GraphMotion	$0.504^{\pm.003}$	$0.699^{\pm.002}$	$0.785^{\pm.002}$	$0.116^{\pm.007}$	$3.070^{\pm.008}$	$9.692^{\pm.067}$	$2.766^{\pm.096}$
AttT2M	$0.499^{\pm.003}$	$0.690^{\pm.002}$	$0.786^{\pm.002}$	$0.112^{\pm.006}$	$3.038^{\pm.007}$	$9.700^{\pm.090}$	$2.452^{\pm.051}$
GUESS	$0.503^{\pm.003}$	$0.688^{\pm.002}$	$0.787^{\pm.002}$	$0.109^{\pm.007}$	$3.006^{\pm.007}$	$9.826^{\pm.104}$	$2.430^{\pm.100}$
MMM	$0.504^{\pm.003}$	$0.696^{\pm.003}$	$0.794^{\pm.002}$	$0.080^{\pm.003}$	$2.998^{\pm.007}$	$9.411^{\pm.058}$	$1.164^{\pm.041}$
MotionCLR	$0.542^{\pm.001}$	$0.733^{\pm.002}$	$0.827^{\pm.003}$	$0.099^{\pm.003}$	$2.981^{\pm.011}$	-	$2.145^{\pm.043}$
MotionGPT	$0.492^{\pm.003}$	$0.681^{\pm.003}$	$0.778^{\pm.002}$	$0.232^{\pm.008}$	$3.096^{\pm.008}$	$9.528^{\pm.071}$	$2.008^{\pm.084}$
MotionGPT-2	$0.496^{\pm.002}$	$0.691^{\pm.003}$	$0.782^{\pm.004}$	$0.191^{\pm.004}$	$3.080^{\pm.013}$	$9.860^{\pm.026}$	$2.137^{\pm.022}$
ParCo	$0.515^{\pm.003}$	$0.706^{\pm.003}$	$0.801^{\pm.002}$	$0.109^{\pm.005}$	$2.927^{\pm.008}$	$9.576^{\pm.088}$	$1.382^{\pm.060}$
DiverseMotion	$0.515^{\pm.003}$	$0.706^{\pm.002}$	$0.802^{\pm.002}$	$0.072^{\pm.004}$	$2.941^{\pm.007}$	$9.683^{\pm.102}$	$1.869^{\pm.089}$
BAD	$0.517^{\pm.002}$	$0.713^{\pm.003}$	$0.808^{\pm.003}$	$0.065^{\pm.003}$	$2.901^{\pm.008}$	$9.694^{\pm.068}$	$1.194^{\pm.044}$
MoMask	$0.521^{\pm.002}$	$0.713^{\pm.002}$	$0.807^{\pm.002}$	$\mathbf{0.045}^{\pm.002}$	$2.958^{\pm.008}$	$9.620^{\pm.064}$	$1.241^{\pm.04}$
MoMask++	$0.528^{\pm.003}$	$0.718^{\pm.003}$	$0.811^{\pm.002}$	$0.072^{\pm.003}$	$2.912^{\pm.008}$	$9.573^{\pm.062}$	$1.227^{\pm.046}$
MotionGPT3	$\underline{0.543}^{\pm.003}$	$\underline{0.735}^{\pm.002}$	$\underline{0.828}^{\pm.002}$	$0.217^{\pm.010}$	$\mathbf{2.793}^{\pm.007}$	$9.662^{\pm.072}$	$1.366^{\pm.046}$
Ours$^{\text{nova}}$	$0.532^{\pm.002}$	$0.725^{\pm.002}$	$0.819^{\pm.002}$	$0.072^{\pm.003}$	$2.887^{\pm.006}$	$9.377^{\pm.078}$	$1.875^{\pm.088}$
Ours$^{\text{orbit}}$	$\mathbf{0.550}^{\pm.003}$	$\mathbf{0.740}^{\pm.003}$	$\mathbf{0.832}^{\pm.002}$	$\underline{0.061}^{\pm.003}$	$\underline{2.852}^{\pm.008}$	$\underline{9.457}^{\pm.067}$	$\underline{2.071}^{\pm.093}$

A.2 More Quantitative Evaluation

To comprehensively evaluate our method on text-to-motion generation, we report a full comparison with prior T2M approaches in Table A. The main methods include the following: TEMOS (Petrovich et al., 2022), TM2T (Guo et al., 2022b), T2M (Guo et al., 2022a), TM2D (Gong et al., 2023), MotionDiffuse (Zhang et al., 2024), MDM (Tevet et al., 2022), MotionLLM (Chen et al., 2024b), MLD (Chen et al., 2023), Motion Mamba (Zhang et al., 2025), ReMoDiffuse (Zhang et al., 2023b), Fg-T2M (Wang et al., 2023), T2M-GPT (Zhang et al., 2023a), EMDM (Zhou et al., 2023), GraphMotion (Jin et al., 2023), AttT2M (Zhong et al., 2023), GUESS (Gao et al., 2024), MMM (Pinyoanuntapong et al., 2024), MotionCLR (Chen et al., 2024a), MotionGPT (Jiang et al., 2023), MotionGPT-2 (Wang et al., 2024), Parco (Zou et al., 2024), DiverseMotion (Lou et al., 2023), BAD (Hosseyni et al., 2024), MoMask (Guo et al., 2024), Momask++ (Guo et al., 2025), MotionGPT3 (Zhu et al., 2025). The results show that our method consistently outperforms prior approaches and achieves competitive performance on HumanML3D.

B More Ablation Studies

In this section, we have made additional supplementary information as follows: (1). The impact of the number of ODE sampling steps and the size of the CFG model during the flow matching process on the generated results. (2). We compared the reasoning speed of our framework with that of other competitive frameworks. (3). The influence of the proportions of Tokens and Latents on motion reconstruction.

Inference Hyperparameters. During the inference process of flow matching, the scaling coefficient s of CFG and the number of ODE sampling L are two crucial hyperparameters. In Figure C, we present the performance curves of FID and R-Precision@3 by changing different values of s and L. Several key observations emerge. Firstly, more sampling times do not necessarily lead to better results. As L increases, FID quickly converges to a minimum value, typically within approximately 40 iterations. After more than 40 iterations, there is no further performance improvement in FID,

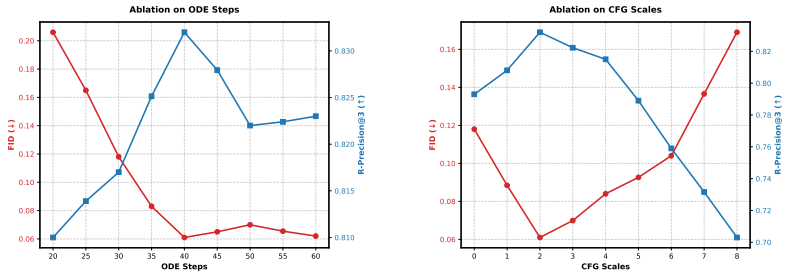


Figure C: Evaluation sweep over ODE numbers L (left) and guidance scale s (right) in inference. We find a accuracy-fidelity sweep spot around s = 2, meanwhile 40 iterations (L = 40) for flow matching yield sufficiently good results.

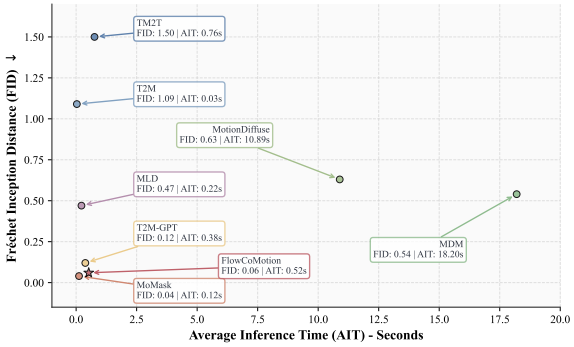


Figure D: **Comparison of inference time costs.** All tests were conducted on a single NVIDIA RTX 2080 Ti. Models closer to the origin indicate better performance. For non-autoregressive models, our method significantly improves inference speed, thanks to the flow matching directional velocity field.

but R-Precision@3 will experience a certain decline. Secondly, the optimal guidance coefficient s for inference is approximately around s = 2. Over-guided decoding may even lead to a decrease in performance. Our FlowCoMotion requires fewer inference steps than most diffusion models.

Inference Time. In Figure D, we compare inference efficiency and quality across methods, measuring runtime as the average latency over 100 samples on an NVIDIA RTX 2080 Ti. FlowCoMotion provides a stronger quality–efficiency trade-off than prior approaches. Specifically, by leveraging a Flow Matching framework rather than traditional diffusion models, our approach significantly accelerates the generation process. The optimal transport trajectories inherent in Flow Matching allow for a drastic reduction in the number of sampling steps required without compromising motion fidelity. Consequently, FlowCoMotion substantially decreases inference latency, achieving generation speeds that closely approach those of fast autoregressive-based methods, while still maintaining superior motion quality.

Representation Dimension. We study how allocating capacity between the token and latent branches affects reconstruction quality. Keeping the total dimensionality fixed, we vary the dimensions of both branches and summarize the results in Figure E. Performance is best when one branch receives most of the capacity while the other retains a non-trivial share; overly skewed allocations lead to clear degradation. This trend supports the complementarity of the two representations: tokens capture semantic structure, whereas continuous latent preserve geometric and temporal detail. A primary–secondary allocation thus provides the strongest hybrid representation under a fixed budget. Conversely, forcing an equal division restricts the representational power of both branches, preventing either from fully modeling its target features. Based on these empirical observations, we adopt this optimal asymmetric allocation as our default configuration for all subsequent experiments.

Dimension Matched Probes on Semantics of Tokens. We additionally fit identical probes to frozen 64D token representations and 64D PCA projections of the continuous representations; PCA and all probe hyperparameters are fitted on the training split only. As shown in Table B, the token

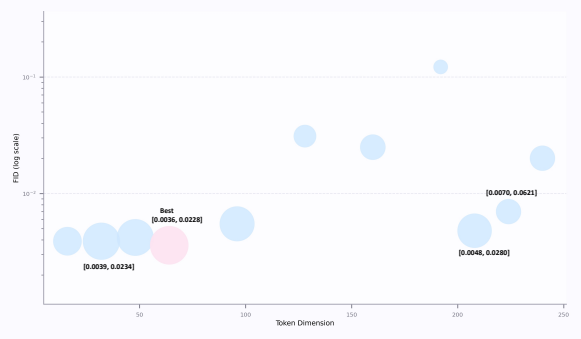


Figure E: **Ablation analysis on representation dimension choices.** We report reconstruction quality using FID and MPJPE. The larger of the circles represents a better MPJPE.

branch better predicts semantic attributes (**0.40 vs. 0.33** mAP), whereas the continuous branch better predicts speed and jerk. Together, the token-attribute association, aligned coarse-code trajectories, and dimension-matched probes support complementary encoding: RVQ tokens preferentially capture semantic motion structure, while continuous latents preserve motion dynamics.

Table B: **Semantic analysis of RVQ tokens and dimension-matched branch probes on HumanML3D.** Speed and Jerk denote mean root speed and mean joint jerk.

Token–Attribute Association (mAP ↑)			
Representation	Action	Direction	Body Part
RVQ Token IDs	0.43	0.34	0.40
Shuffled Token IDs	0.15	0.14	0.17
Dimension-Matched Frozen Probes			
Representation	Semantic mAP ↑	Speed R^2 ↑	Jerk R^2 ↑
Token Branch (64-D)	0.40	0.41	0.30
Continuous Branch (PCA-64)	0.33	0.63	0.51

C Human Evaluation on Text-to-Motion Quality

This evaluation presents a comprehensive user evaluation of our motion generation framework FlowCoMotion by conducting a Google Forms survey as shown in Figure F. Participants were asked to evaluate the effectiveness and realism of the generated motions based on accuracy, smoothness, realism, and overall user experience.

Evaluation Setup. 1. For the individual evaluation, we provided 44 distinct motion sequences. To mitigate potential biases introduced by user selection inertia, 6 additional sequences were randomly inserted into the set but excluded from the final analysis. Participants rated these motions on a three-point scale (scored from 1 to 3), and we report the final average rating. 2. For the comparative evaluation, we curated 50 sets of motion sequences generated by different models based on identical textual descriptions. Participants were asked to select the best-performing model within each set. This allowed for a direct assessment of our model's realism, coherence, and engagement against existing alternatives. We report the final average win rate (out of 50) based on how often users felt our method achieved the best results.

Baseline. We selected three equally competitive models: MotionGPT, Momask++, and MotionGPT3. We evaluated them using high-quality rendered sequence images and videos.

Human Evaluation Results Analysis. As illustrated in Figure G, FlowCoMotion demonstrates clear superiority over baselines including MoMask++ , MotionGPT3, and MotionGPT. In the individual evaluation, our method achieved high satisfaction rates of 86.0% , 88.0% , and 89.0% for Motion Quality , Motion Diversity , and Text-Motion Alignment, respectively. Furthermore, the comparative evaluation reveals a strong human preference for FlowCoMotion, which was selected as the top-performing model by 44.8% , 48.8% , and 52.3% of participants across the three respective metrics.

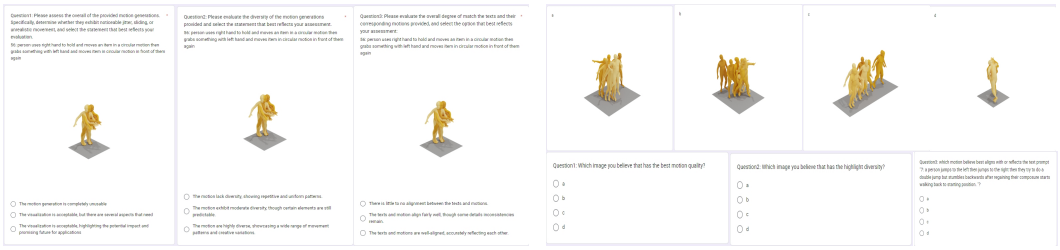


Figure F: **User study form.** The User Interface (UI) was used in our user study. Our user study consists of 50 individual evaluations and 50 comparative evaluations. We incorporate elements from other methods to prevent the review results from becoming invalid due to user inertia.

These findings confirm that our model generates more realistic, diverse, and well-aligned motions than existing approaches.

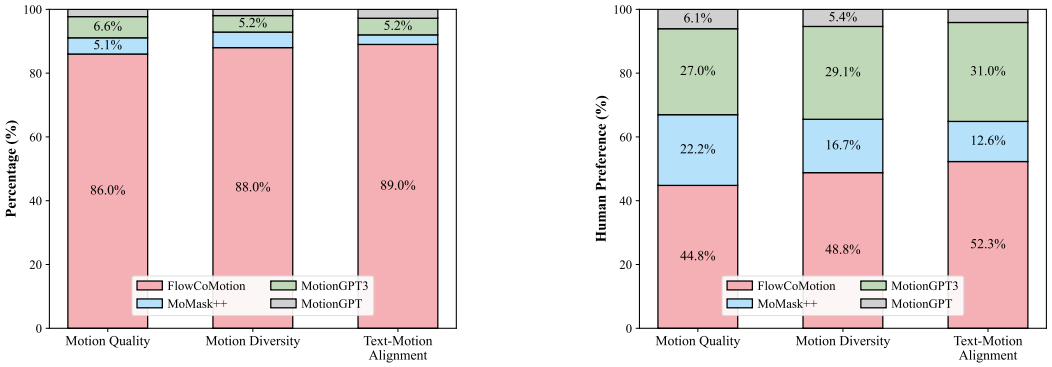


Figure G: Human preference on Text-to-Motion Generation by comparing FlowCoMotion with MotionGPT (Jiang et al., 2023), Momask++ (Guo et al., 2025) and MotionGPT3 (Zhu et al., 2025). We report the scores on three generation aspects.

D Why Token-Latent Coupling Helps: A Theoretical View

Let X represent a motion sequence, modeled as a vector in $\mathbb{R}^{T\times d}$, where T denotes the sequence length and d is the dimension of each representation. Additionally, let C be the textual condition that drives the motion generation process.

Our framework constructs two distinct representations from X: (1). A discrete token representation Z_q, derived from the Residual Vector Quantization tokenizer. (2). A continuous latent representation Z_c, obtained from a VAE's encoder.

To integrate both modalities, we first extract a dequantized multi-scale token feature $\hat{z}_q = \sum_{v=0}^{V} \mathcal{I}(\hat{f}^v, h^v)$, where $\hat{f}^v$ represents the token features at scale v, and h^v denotes the associated multi-scale feature for each token. We can combine $\hat{z}_q$ with $\hat{z}_c$ using a coupling network, producing a fused representation $\mathcal{C}$ defined as:

$$\hat{\mathbf{z}}_{gt} = \mathcal{C}(\hat{z}_q, \hat{z}_c) \quad \hat{\mathbf{m}} = \mathcal{D}(\hat{\mathbf{z}}_{gt}) \tag{8}$$

In Stage 3, we define the flow matching data endpoint as $z_1 = z_{gt}$, and train a condition velocity field to map $z_0 \sim \mathcal{N}(0, I)$ to z_1, where z_0 is drawn from a standard Gaussian distribution.

D.1 Coupling improves the best achievable reconstruction risk.

We quantify the utility of token-latent coupling through the population Bayes risk under squared reconstruction loss. For any representation S of X, define the optimal reconstruction risk,

$$\mathcal{R}^\star(S) \triangleq \inf_g \mathbb{E}\left[\|X - g(S)\|_2^2\right], \tag{9}$$

where g ranges over all measurable decoders.

Proposition D.1 (Bayes risk monotonicity and a joint-representation limit)**.** *Let S_1, S_2 be two representations of X. If $\sigma(S_1) \subseteq \sigma(S_2)$ (i.e., S_2 is at least as informative as S_1), then*

$$\mathcal{R}^\star(S_2) \le \mathcal{R}^\star(S_1). \tag{10}$$

In particular, for the joint-representation $S_{joint} \triangleq (\hat{z}_c, \hat{z}_q)$, we have

$$\mathcal{R}^\star(\hat{z}_c, \hat{z}_q) \;\le\; \min\Big\{\mathcal{R}^\star(\hat{z}_c),\; \mathcal{R}^\star(\hat{z}_q)\Big\}. \tag{11}$$

Moreover, the inequality in Equation (11) is typically strict whenever the two branches are complementary, i.e., one branch reduces the conditional reconstruction uncertainty left by the other on a set of nonzero probability mass.

Proof. Under squared loss, the Bayes-optimal decoder is the conditional expectation $g^\star(S) = \mathbb{E}[X \mid S]$. Hence

$$\mathcal{R}^\star(S) = \mathbb{E}\big[\|X - \mathbb{E}[X \mid S]\|_2^2\big] = \mathbb{E}[\mathrm{Var}(X \mid S)]\,. \tag{12}$$

If $\sigma(S_1) \subseteq \sigma(S_2)$, conditioning on S_2 cannot increase conditional variance, implying $\mathrm{Var}(X \mid S_2) \le \mathrm{Var}(X \mid S_1)$ almost surely, and thus $\mathcal{R}^\star(S_2) \le \mathcal{R}^\star(S_1)$. Applying this with $S_2 = (\hat{z}_c, \hat{z}_q)$ and $S_1 = \hat{z}_c$ (or $S_1 = \hat{z}_q$) yields Equation (11). □

Proposition D.1 should be read as a representational *limit statement*: the joint statistic $(\hat{z}_c, \hat{z}_q)$ admits a Bayes-optimal reconstruction risk no worse than either branch alone. Our coupling network $\hat{z}_{gt} = \mathcal{C}(\hat{z}_q, \hat{z}_c)$ can be viewed as a learnable *compression* of this joint statistic. In Stage 1 and 2, we train $\mathcal{C}$ and the decoder $\mathcal{D}$ to reconstruct X from $\hat{z}_{gt}$, which encourages $\hat{z}_{gt}$ to preserve the reconstruction-relevant information in $(\hat{z}_c, \hat{z}_q)$. Consequently, as the capacity of $\mathcal{C}$ and $\mathcal{D}$ increases and optimization succeeds, the achievable reconstruction risk using $\hat{z}_{gt}$ can approach the joint-representation limit in Equation (11). DisCo-Diff studies how separately generated discrete latents simplify continuous diffusion. Proposition D.1 instead states only the reconstruction information available from a joint representation; we do not claim novelty for this general monotonicity result.

D.2 Coupling Reduces the Irreducible Error of Flow Matching.

We next connect coupling to conditional flow matching. Let z_1 denote the target endpoint latent (in our method, $z_1 = \hat{z}_{gt}$), and draw $z_0 \sim \mathcal{N}(0, I)$. Consider the standard linear interpolation path,

$$z_t \;=\; (1-t)z_0 + tz_1, \qquad u_t \;\triangleq\; \frac{dz_t}{dt} \;=\; z_1 - z_0, \qquad t \sim \mathcal{U}(0,1). \tag{13}$$

The population conditional flow matching objective is

$$\mathcal{L}_{\mathrm{FM}}(v) \triangleq \mathbb{E}\big[\|u_t - v(z_t, t, C)\|_2^2\big]\,, \tag{14}$$

where the expectation is over (t, z_0, z_1, C). By the optimality of L_2 regression,

$$v^\star(z_t, t, C) = \mathbb{E}[u_t \mid z_t, t, C], \tag{15}$$

and the minimum achievable loss equals the irreducible conditional variance

$$\mathcal{L}^\star_{\mathrm{FM}}(z_1) \;\triangleq\; \inf_v \mathcal{L}_{\mathrm{FM}}(v) \;=\; \mathbb{E}[\mathrm{Var}(u_t \mid z_t, t, C)]\,. \tag{16}$$

We compare two endpoint constructions under the same text condition C: a coupled endpoint z_1^{cp} (e.g., $z_1^{\mathrm{cp}} = \hat{z}_{gt} = \mathcal{C}(\hat{z}_q, \hat{z}_c)$) and a single-branch endpoint z_1^{sb}. These two choices induce different random variables $z_t^{\mathrm{cp}} = (1-t)z_0 + tz_1^{\mathrm{cp}}$ and $z_t^{\mathrm{sb}} = (1-t)z_0 + tz_1^{\mathrm{sb}}$, and their irreducible errors are compared through their respective conditionings. In addition, throughout Stage 1 and 2 we regularize the latent scales (e.g., VAE KL regularization), so z_1^{cp} and z_1^{sb} are comparable in magnitude; hence the variance change discussed below is attributed to reduced conditional ambiguity (mode dispersion) rather than trivial rescaling.

Proposition D.2 (Reduction of an irreducible error upper bound via endpoint concentration)**.** *Assume that coupling makes the endpoint distribution more concentrated under the text condition, in the sense that*

$$\mathbb{E}\big[\|z_1^{\mathrm{cp}} - \mathbb{E}[z_1^{\mathrm{cp}} \mid C]\|_2^2\big] \;\le\; \mathbb{E}\big[\|z_1^{\mathrm{sb}} - \mathbb{E}[z_1^{\mathrm{sb}} \mid C]\|_2^2\big]\,. \tag{17}$$

Then, for any fixed $t \in (0,1)$, the irreducible conditional variance term admits the bound

$$\mathbb{E}[\mathrm{Var}(u_t^{\mathrm{cp}} \mid z_t^{\mathrm{cp}}, t, C)] \;\le\; \frac{1}{(1-t)^2}\,\mathbb{E}\big[\|z_1^{\mathrm{cp}} - \mathbb{E}[z_1^{\mathrm{cp}} \mid C]\|_2^2\big]\,, \tag{18}$$

and analogously

$$\mathbb{E}\big[\mathrm{Var}(u_t^{\mathrm{sb}} \mid z_t^{\mathrm{sb}}, t, C)\big] \;\le\; \frac{1}{(1-t)^2}\,\mathbb{E}\big[\|z_1^{\mathrm{sb}} - \mathbb{E}[z_1^{\mathrm{sb}} \mid C]\|_2^2\big]\,. \tag{19}$$

Consequently, under Equation (17), coupling reduces an upper bound *on the irreducible variance term in Equation (16). (The factor $\frac{1}{(1-t)^2}$ diverges as $t \to 1$, which is expected: near the endpoint, the uncertainty of the velocity field is highly sensitive to the endpoint distribution.)*

Proof. Fix $t \in (0,1)$ and condition on (t, C). Since $z_t = (1-t)z_0 + tz_1$ and $u_t = z_1 - z_0$, we eliminate z_0 via

$$z_0 = z_1 - u_t \quad\Longrightarrow\quad z_t = (1-t)(z_1 - u_t) + tz_1 = z_1 - (1-t)u_t, \tag{20}$$

equivalently $z_1 = z_t + (1-t)u_t$. Conditioning on (z_t, t, C) yields

$$\mathrm{Var}(z_1 \mid z_t, t, C) = (1-t)^2\,\mathrm{Var}(u_t \mid z_t, t, C), \tag{21}$$

and hence

$$\mathbb{E}[\mathrm{Var}(u_t \mid z_t, t, C)] = \frac{1}{(1-t)^2}\,\mathbb{E}[\mathrm{Var}(z_1 \mid z_t, t, C)]\,. \tag{22}$$

By the law of total variance,

$$\begin{aligned}\mathbb{E}[\mathrm{Var}(z_1 \mid z_t, t, C)] &\le \mathbb{E}[\mathrm{Var}(z_1 \mid C)] \\ &= \mathbb{E}\big[\|z_1 - \mathbb{E}[z_1 \mid C]\|_2^2\big] - \mathbb{E}\big[\|\mathbb{E}[z_1 \mid z_t, t, C] - \mathbb{E}[z_1 \mid C]\|_2^2\big] \\ &\le \mathbb{E}\big[\|z_1 - \mathbb{E}[z_1 \mid C]\|_2^2\big]\,.\end{aligned} \tag{23}$$

Combining the above gives the generic bound

$$\mathbb{E}[\mathrm{Var}(u_t \mid z_t, t, C)] \le \frac{1}{(1-t)^2}\,\mathbb{E}\big[\|z_1 - \mathbb{E}[z_1 \mid C]\|_2^2\big]\,, \tag{24}$$

which, applied to z_1^{cp} and z_1^{sb}, yields Equations (18) and (19). Finally, Equation (17) transfers the improvement to the corresponding upper bound. □

Equation (16) shows that conditional flow matching is fundamentally limited by the conditional dispersion of the target velocity u_t given (z_t, t, C). In text-to-motion, the mapping $C \mapsto X$ is typically one-to-many, so a single-branch endpoint may aggregate multiple semantic modes under the same caption, increasing the variability of z_1 and injecting additional uncertainty into the regression target u_t. Coupling injects discrete semantic anchors into the endpoint construction, which can make $p(z_1 \mid C)$ more concentrated in the sense of Equation (17), thereby reducing an upper bound on the irreducible variance term in Equation (16) and easing the conditional regression task learned by v_θ.

D.3 Discussion on assumptions and how our design satisfies them.

The above analysis relies on two practical assumptions. First, the coupled representation should be sufficiently *information-preserving* with respect to the two branches, so that fusing $(\hat{z}_q, \hat{z}_c)$ into $\hat{z}_{gt}$ does not discard reconstruction-relevant cues. In our design, this is supported by an expressive coupling network $\mathcal{C}$ (e.g., residual MLP/Transformer blocks) and the reconstruction-driven supervision in Stage 1 and 2, where $\mathcal{C}$ and $\mathcal{D}$ are optimized to minimize $\mathbb{E}\|X - \mathcal{D}(\mathcal{C}(\hat{z}_q, \hat{z}_c))\|_2^2$. This training objective encourages $\hat{z}_{gt}$ to behave as a faithful compression of the joint statistic, aligning with the representational limit in Equation (11).

Second, the variance-bound improvement in Proposition D.2 presumes that coupling makes the endpoint distribution $p(z_1 \mid C)$ more concentrated in a scale-comparable space. Our implementation promotes this comparability through latent regularization and normalization (e.g., the VAE prior/KL regularization for $\hat{z}_c$ and codebook/embedding normalization for the RVQ branch), which prevents the bound reduction from being explained by trivial rescaling. Moreover, the RVQ commitment loss stabilizes token assignments across time scales, reducing semantic drift of $\hat{z}_q$ and improving the predictability between discrete anchors and continuous motion dynamics. Empirically, this reduces conditional ambiguity of the endpoint under the same caption, making the conditional regression target in flow matching less noisy and easier to learn with finite-capacity vector fields.

E Details on Our Work

E.1 Training Objectives

Reconstruction Loss. As the primary objective, this loss ensures the accurate recovery of motion sequences by minimizing the L2 distance between predicted ($\hat{X}$) and ground truth (X) motions. Its weight is fixed at $\lambda_{rec} = 1.0$ across both stages:

$$L_{rec} = \mathbb{E}[\|X - \hat{X}\|_2^2] \tag{25}$$

Commitment Loss. To stabilize the quantization process, the commitment loss ($\lambda_{commit} = 0.01$) minimizes the distance between the encoder's continuous latent output and its nearest codebook entry. Coupled with temporal resolution sampling, this guarantees that the model effectively samples semantic anchors at varying time intervals without needing explicit formulation here.

Forward Kinematics (FK) Loss. Acting as a structural regularization term to ensure physically plausible motions, the FK loss compares predicted and expected joint positions:

$$L_{fk} = \mathbb{E}[\|X_{\text{pred}} - X_{\text{FK}}\|_2^2] \tag{26}$$

We set $\lambda_{fk} = 0.5$ in Stage 1 and reduce it to 0.1 in Stage 2.

Dynamic Distillation Loss. In Stage 1, a multi-view teacher-student alignment regularizes the latent space. The weight λ_{dis} linearly warms up to 0.02, holds constant, then decays to 0.0. *View generation.* Sequences yield two global views and four local views (random crops $\in [0.3, 0.7]$). Local views receive temporal augmentations: time-warp, Gaussian noise, and temporal masking. *Aggregation.* A single per-view representation is obtained via temporal masked mean pooling:

$$\phi(F, M) = \frac{\sum_{t=1}^{T} M_t \, F_t}{\sum_{t=1}^{T} M_t + \varepsilon} \tag{27}$$

Temperature and centering. We use separate temperatures ($\tau_{\text{teacher}} = 0.04$, $\tau_{\text{student}} = 0.07$) and center teacher logits via an EMA buffer. The L2 distillation objective minimizes the error between student outputs and teacher targets.

Velocity, Acceleration, and Jerk Losses. Applied in Stage 2 to enhance temporal consistency, these losses penalize deviations in the first three derivatives of motion: velocity ($\lambda_{vel} = 0.2$), acceleration ($\lambda_{acc} = 0.05$), and jerk ($\lambda_{jerk} = 0.01$). Balancing these higher-order kinematic constraints ensures smooth, natural transitions and overall generation stability.

E.2 Codebook Size and Resolution Segmentation

The codebook size and resolution segmentation significantly impact quantization and generation quality. Our model fixes the codebook size at 1024 with an entry dimension of 64. Combined with a continuous latent dimension of 192, the total latent dimension per sample is 256. To optimize quantization, we adapt the resolution segmentation per dataset. For HumanML3D, we apply a four-layer scale $\mathcal{S}_h = [8, 4, 2, 1]$. In SnapMoGen, we utilize a two-layer scale $\mathcal{S}_h = [2, 1]$. Through temporal resolution sampling, the Token Branch extracts semantic anchors at varying intervals to enhance fidelity. Combining these varying scales with the commitment loss enables the model to capture multiple levels of motion granularity, producing detailed and coherent sequences across diverse datasets.

E.3 Flow Matching Details

In our implementation, we define a continuous path between a prior latent variable $z_0 \sim \mathcal{N}(0, I)$ and the real data latent z_1, leveraging a velocity field predicted by the flow model.

Latent Variable Interpolation and Velocity. During training and inference, the latent variable z_t is linearly interpolated as $z_t = tz_1 + (1-t)z_0$ for $t \in [0, 1]$, smoothly transitioning from Gaussian noise to the data distribution. The conditional flow model $v_\theta(z_t, t, c)$ predicts the instantaneous velocity $u_t = z_1 - z_0$ at each timestep, utilizing z_t and an auxiliary textual condition c.

Flow Matching Loss. The training objective minimizes the mean squared error between the predicted and true velocities:

$$\mathcal{L} = \mathbb{E}_{z_0, z_1, t}\left[\|(z_1 - z_0) - v_\theta(z_t, t, c)\|_2^2\right], \tag{28}$$

where z_1 is the ground truth motion latent from the token-latent coupling network. This enables the model to accurately bridge the prior and data distributions.

Inference and ODE Solver. Inference begins with $z_0 \sim \mathcal{N}(0, I)$. We solve the equation $\frac{dz_t}{dt} = v_\theta(z_t, t, c)$ over $t \in [0, 1]$ using a fixed-step Euler solver to output the final latent $\hat{z}_1$. We set the total steps to 40, updating the latent iteratively via $z_{t+1} = z_t + \Delta t \cdot v_\theta(z_t, t, c)$. This step configuration provides an optimal trade-off between high-fidelity motion quality and computational efficiency.

Classifier-Free Guidance (CFG). We apply CFG with a fixed scale $s = 2.0$. By concurrently predicting conditional (v_{cond}) and unconditional (v_{uncond}, via zeroed text embeddings) velocities, we guide the generation:

$$v_{\text{cfg}} = v_{\text{uncond}} + s \cdot (v_{\text{cond}} - v_{\text{uncond}}). \tag{29}$$

Noise Schedule. By modulating the velocity field $v_\theta(z_t, t, c)$ with time t, the noise schedule ensures the generated sequence is progressively refined.

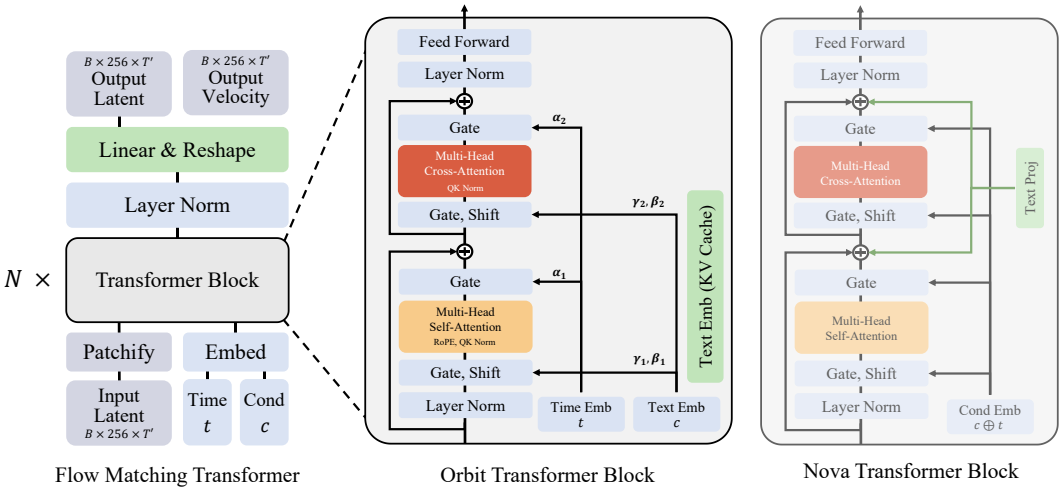


Figure H: **Details of our Flow Matching Transformer model.** Given an Input Latent of shape $B \times 256 \times T'$, alongside Time Cond and Text Emb, the initial features are processed through Patchify and Embed layers. These tokens are then passed through N Transformer Blocks, which incorporate Orbit Transformer Block and Nova Transformer Block designs utilizing Multi-Head Self-Attention and Multi-Head Cross-Attention. After passing through the N layers, latent generation proceeds via a Linear & Reshape layer, ultimately yielding the Output Latent and Output Velocity in their original $B \times 256 \times T'$ dimensions.

E.4 Model Parameterization

Backbone configuration. We parameterize the conditional vector field using a Transformer-based flow matching network which shown in Figure H. The input latent representations are first linearly projected to a hidden dimension $d_{\text{model}} = 512$. The network architecture is composed of 15 Transformer blocks, each utilizing 8 attention heads and applying a dropout rate of 0.1. To ensure training

stability, we adopt a pre-normalization layout, applying LayerNorm prior to the sub-modules and employing residual connections afterward.

Time embedding parameterization. The time step t is embedded using sinusoidal Fourier features with a maximum period of $10{,}000$. The resulting concatenated $\sin(\cdot)$ and $\cos(\cdot)$ components are then mapped to the hidden dimension d_{model} through a two-layer multi-layer perceptron (MLP) with an expansion ratio of 4 (i.e., $512 \to 2048 \to 512$). This resulting time embedding serves as a crucial conditioning signal injected into each block, preserving the latent sequence length throughout the network.

Condition builder and injection. Outputs derived from the text encoder are first projected to d_{model} utilizing a sequence of Linear and LayerNorm layers. Subsequently, a global text summary is extracted via masked mean pooling. This summary representation is then transformed and additively combined with the time step embedding t. To mitigate the risk of early over-conditioning during training, the text contribution is scaled by a learnable gating parameter, initialized at $\alpha_{\text{init}} = 0.02$. The resulting combined condition vector is injected into every Transformer block, while the fine-grained, token-level text features are retained to serve as memory for the cross-attention mechanisms.

Positional encoding and attention masks. For the latent sequence, we incorporate learned positional embeddings accommodating a maximum length of 1024. Conversely, the text memory stream does not require additional positional encoding, as the encoder outputs are inherently position-aware. Both the latent and text streams employ key-padding masks to handle variable sequence lengths. To ensure the integrity of the generation process, a latent validity mask is systematically reapplied following each ODE solver update, guaranteeing that the padded positions remain strictly unchanged.

Stability-oriented block design. To further enhance numerical stability within the network, the Orbit blocks apply QK normalization and facilitate the injection of the combined conditioning signal via adaptive layer normalization (AdaLN). Finally, to optimize inference efficiency, we introduce an option to cache the key-value (KV) tensors associated with the frozen text memory. This efficiently circumvents the need to recompute text-side attention across successive ODE solver steps.

E.5 Construction of the Harder HumanML3D Test Set

Overview. Harder-HumanML3D is a diagnostic subset of the official HumanML3D test split, selected from captions with complex, multi-step, or spatially precise constraints. Our methodology combines heuristic linguistic rules with large language model (LLM) reasoning to ensure both structural complexity and semantic depth.

Table C: **Keyword Lexicon for Multi-Dimensional Difficulty Rules.** We define eight linguistic dimensions ($D_1 - D_8$) to quantify the complexity of motion prompts. Keywords are matched using heuristic rules to compute the initial difficulty score.

Dimension	Matching Keywords / Patterns
D_1 **Direction**	left, right, forward, backward, clockwise, counterclockwise, sideways, diagonal, north ,south, east, westward, …
D_2 **Body Parts**	arm, leg, hand, foot, head, torso, hip, shoulder, elbow, knee, wrist, ankle, spine, neck, finger, chest, back, …
D_3 **Modifiers**	quickly, slowly, slightly, sharply, gently, rapidly, suddenly, gradually, briefly, continuously, repeatedly, …
D_4 **Temporal**	then, while, before, after, simultaneously, followed by, at the same time, next, finally, meanwhile, …
D_5 **Count**	twice, three times, several times, multiple times, once, again, number of, …
D_6 **Verbs**	kick, punch, bend, twist, spin, crawl, kneel, wave, crouch, squat, hop, leap, lunge, stomp, clap, point, rotate, …
D_7 **Constraint**	only, specific, particular, certain, exactly, precisely, must, specifically, …
D_8 **Spatial**	upward, downward, inward, outward, horizontal, vertical, perpendicular, parallel, above, below, overhead, …

Multi-Dimensional Difficulty Rules. We define eight linguistic dimensions (D_1 to D_8) to quantify the complexity of motion prompts, targeting specific challenges in motion synthesis such as directional precision, temporal composition, and spatial relations. The detailed keyword lexicon and the assigned heuristic weight w_i for each dimension are summarized in Table C. Based on this lexicon, the initial heuristic score is calculated as:

$$S_{rule} = \sum_{i=1}^{8} w_i \cdot \text{count}(D_i) \tag{30}$$



You are an expert in evaluating the difficulty of text-to-motion generation tasks. Given a natural language description of a human body motion, rate its generation difficulty on a scale from 1 to 10. Difficulty criteria: Higher scores (7-10) for prompts with: fine-grained directional cues (left/right), specific body part references, temporal sequences, spatial constraints, combined or simultaneous actions, speed/manner modifiers, counting/repetition. Lower scores (1-3) for simple, vague, single-action descriptions. Respond **ONLY** with a **JSON** object in this exact format (no extra text): **{"score": <int 1-10>, "reason": "<one sentence explanation>"}**



Motion description: **"{caption}"**

Figure I: **Prompt template for Difficulty Oracle.** The system prompt (top) defines linguistic evaluation criteria and enforces JSON output, while the user prompt (bottom) supplies the target caption to enable structured semantic scoring.

LLM-Assisted Semantic Scoring. Recognizing that word counts alone cannot fully capture logical complexity, we employ `DeepSeek-R1` to oracle the difficulty of the prompts. For candidates with $S_{rule} \geq 4$, the LLM evaluates the prompt's difficulty on a scale of 1–10 and provides a natural language justification. The complete prompt template designed for this evaluation is illustrated in Figure I. This crucial step filters out prompts that contain difficulty keywords but remain semantically simple (e.g., "a person walks left").

Final Selection and Dataset Statistics. The final difficulty score is a fusion of the heuristic and LLM scores:

$$S_{final} = S_{rule} + \alpha \cdot S_{llm} \tag{31}$$

where we set $\alpha = 1.0$. Following the protocol of Fg-T2M (Wang et al., 2023), we select the top 2,582 unique motion sequences with the highest S_{final}.

The subset contains more directional and compositional captions than the full split and is used only as a diagnostic complement to the standard evaluation.