EFFICIENT 3D MOTION GENERATION FROM VISUAL DEMONSTRATIONS

Anonymous authors
Paper under double-blind review

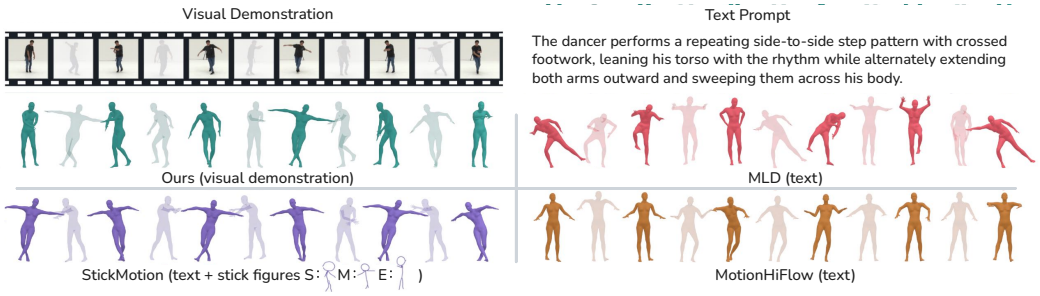

Figure 1: Our method directly conditions on a demonstration video and more faithfully follows its motion details and timing than alternative conditioning paradigms. Text-based methods, MLD (Chen et al., 2023) and MotionHiFlow (Li et al., 2026), capture only coarse motion semantics and fail to reproduce the intended fine-grained motion, whereas StickMotion (Wang et al., 2025) improves pose-level control using additional manually specified start/middle/end stick-figure sketches (S/M/E), but still lacks precise temporal control. Opaque frames denote sparse reference observations and translucent frames show intermediate motion. Additional visualizations: iclr-demo.pages.dev.

ABSTRACT

Existing 3D motion generators face a trade-off between convenient but coarse text-based control and precise but manually specified pose guidance. Moreover, diffusion-based methods further incur substantial iterative sampling costs. We propose FlowMimic, an efficient framework for generating complete 3D human motion from sparse, timestamped 2D pose observations extracted from videos or image sequences. FlowMimic consists of a Motion Latent Codec (MLC), a Time-Aware Pose Encoder (TAPE) for sparse spatiotemporal conditioning, and a Conditional Motion Flow (CMF) that performs Flow Matching directly in the learned motion latent space. The resulting flow supports single-step motion generation while retaining effective control from sparse pose observations. Experiments on AIST++ show that FlowMimic faithfully follows sparse pose observations while maintaining realistic motion quality, and substantially improves inference efficiency over iterative diffusion-based generation.

1 INTRODUCTION

In recent years, 3D human motion generation has advanced rapidly, with growing applications in filmmaking, game development, and virtual reality. Currently, diffusion models have become widely adopted due to their strong generative performance. The availability of large-scale text-motion datasets has further made text-conditioned diffusion the dominant paradigm for controllable motion generation (Plappert et al., 2016; Guo et al., 2022). Despite substantial improvements in generation quality, several important challenges remain (Tevet et al., 2023; Zhang et al., 2024).

Human motion is naturally high-dimensional and temporally structured, making direct generative modeling in the 3D motion space computationally expensive (Chen et al., 2023). This cost is further amplified by diffusion models, whose iterative denoising process requires repeated network evaluations during inference. Consequently, although diffusion-based methods achieve strong generation quality, their generation efficiency remains a practical bottleneck.

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.

In addition, current motion generation methods rely heavily on textual descriptions as the input condition for controlling generated results. Natural language provides semantic intent, but struggles to precisely describe fine-grained motion details and their temporal relationships. Achieving the desired motion often requires iterative refinement of the textual description, yet the generated results may still fail to fully meet user requirements (Wang et al., 2025). Thus, text remains an indirect way of specifying precise spatiotemporal motion. Recent multimodal approaches such as StickMotion (Wang et al., 2025) improve controllability by allowing users to specify three key poses directly through hand-drawn stick figures. However, manually constructing such guidance requires additional user effort and drawing skills, while text is still needed to provide the primary semantic description. Moreover, manually specified key poses constrain only selected moments of a motion, making it difficult to precisely describe fine-grained motion dynamics and temporal transitions.

In this work, we consider a more direct setting in which a visual motion demonstration is already available. Instead of requiring users to describe the desired motion through text or manually construct pose guidance, the observed human demonstrations can directly provide both semantic and spatiotemporal cues for motion generation. Under this circumstance, we propose FlowMimic, a framework for efficient 3D human motion generation conditioned on sparse human pose observations extracted from demonstration videos.

FlowMimic addresses motion representation, visual conditioning, and efficient generation through three main components. First, the Motion Latent Codec (MLC) compresses high-dimensional motion sequences into a compact set of latent tokens that capture their underlying motion dynamics. Second, the Time-Aware Pose Encoder (TAPE) encodes sparse 2D pose observations derived from demonstration videos together with their timestamps, allowing the conditioning signal to explicitly represent both what pose is observed and when it occurs. Finally, to improve generation efficiency, we formulate motion generation with Flow Matching, which avoids the long iterative denoising process of diffusion models and enables generation with fewer network evaluations. The resulting Conditional Motion Flow (CMF) generates motion in the learned latent space under the guidance encoded by TAPE. Our main contributions are summarized as follows:

(1) We formulate visual motion demonstration as a direct conditioning modality for 3D human motion generation, using sparse 2D human pose observations extracted from videos to provide explicit semantic and spatiotemporal guidance without relying on textual descriptions.

(2) We propose a compact latent-space motion generation framework based on Flow Matching, where the Motion Latent Codec (MLC) compresses long motion sequences into a small set of latent tokens and the Conditional Motion Flow (CMF) enables efficient single-step generation in the learned latent space.

(3) We introduce the Time-Aware Pose Encoder (TAPE) to encode sparse 2D pose observations extracted from demonstration videos together with their timestamps, providing temporally explicit conditioning for controllable motion generation.

(4) Extensive experiments on the AIST++ dataset demonstrate competitive generation quality and substantially improved condition controllability and sampling efficiency; ablation studies further validate the effectiveness of the proposed method.

2 Related Work

Diffusion-based Motion Generation. Diffusion models are widely used for text-conditioned human motion generation. MotionDiffuse (Zhang et al., 2024) introduced diffusion-based text-to-motion generation with body-part and temporally varying textual control, while MDM (Tevet et al., 2023) established a widely adopted Transformer-based diffusion framework with geometric motion losses. MLD (Chen et al., 2023) subsequently moved generation into a compact latent space to reduce computational cost. Later work focused on stronger control: ReMoDiffuse (Zhang et al., 2023) augments text conditioning through semantic and kinematic retrieval, GraphMotion (Jin et al., 2023) hierarchically decomposes motion semantics, and GMD (Karunratanakul et al., 2023) and OmniControl (Xie et al., 2024) introduce explicit spatial constraints. FlowMDM (Barquero et al., 2024) and AToM (Han et al., 2025) further improve temporal composition and event-level correspondence. Despite these advances, language typically specifies actions, attributes, and event ordering rather than complete spatiotemporal trajectories, leaving ambiguity in precise kinematic control.

Fine-grained Motion Conditioning. Beyond language, motion generation has been conditioned on music, speech, environments, and partial body observations. Music-driven methods such as EDGE (Tseng et al., 2023) and Lodge (Li et al., 2024) exploit rhythmic cues for beat alignment and long-term dance generation, but music does not uniquely determine choreography. AGRoL (Du et al., 2023) instead uses measurements from sparse wearable trackers to synthesize unobserved body motion. More directly, CondMDI (Cohan et al., 2024) performs motion in-betweening from text and sparse full-body 3D keyframes, while StickMotion (Wang et al., 2025) uses user-drawn stick figures for explicit kinematic guidance, but requires manual specification and can introduce ambiguity in the intended motion cues. Thus, more explicit conditions improve fine-grained control, but may require 3D motion cues, specialized sensing or additional manual input.

Flow-based Motion Generation. Flow Matching (Lipman et al., 2023) learns a continuous transport vector field through simulation-free regression, while Rectified Flow (Liu et al., 2023) straightens transport trajectories to enable coarser numerical integration. Flow Matching itself, however, does not imply one-step synthesis and still requires ODE integration. For motion generation, Motion Flow Matching (Hu et al., 2023) demonstrated generation with roughly ten sampling steps. FlowMotion (Canales Cuba et al., 2025) addressed motion-specific temporal artifacts through a target-predictive formulation, while MotionHiFlow (Li et al., 2026) introduced Hierarchical Flow Matching across temporal scales to capture both global structure and high-frequency details. These results demonstrate the efficiency of flow-based motion generation, but preserving fine-grained conditional fidelity and realistic temporal dynamics under very few function evaluations remains challenging.

3 METHOD

Given a series of images or video clips containing human motion demonstrations, our goal is to extract time-sparse 2D pose observations, $\mathcal{O}_{1:K} = \{(\mathbf{o}_k, \tau_k)\}_{k=1}^{K}$, and use them as reference conditions to generate a temporally continuous 3D pose sequence $\hat{\mathbf{x}}_{1:T} = (\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_T)$, where K is the number of observations, τ_k denotes the normalized timestamp of each observation, and T is the output sequence length. The input frames do not need to be temporally consecutive, as long as they specify which poses should be achieved at selected moments throughout the motion sequence. The resulting sparse pose observations guide the generation of a continuous 3D motion, while the conditioning strength can be adjusted to control generation diversity. To achieve this efficiently, we perform conditional Flow Matching in a compact motion latent space and employ a dedicated time-aware 2D pose encoder to inject sparse control signals. Fig. 2 shows the overall architecture.

3.1 PRELIMINARIES: FLOW MATCHING AND RECTIFICATION

Let $\mathbf{z}_1 \sim p_z$ denote a data sample and $\mathbf{z}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ a Gaussian source sample. Following Rectified Flow (Liu et al., 2023), we use the linear interpolation path

$$\mathbf{z}_t = (1-t)\mathbf{z}_0 + t\mathbf{z}_1, \qquad \mathbf{v}^\star = \mathbf{z}_1 - \mathbf{z}_0, \qquad t \sim \mathcal{U}(0,1), \tag{1}$$

and train a velocity field $\mathbf{v}_\theta(\mathbf{z}_t, t)$ to regress the constant target velocity $\mathbf{v}^\star$. At inference, samples are obtained by integrating the corresponding ordinary differential equation (ODE) $\mathrm{d}\mathbf{z}_t/\mathrm{d}t = \mathbf{v}_\theta(\mathbf{z}_t, t)$ from $t = 0$ to $t = 1$.

Rectification (reflow) further straightens the learned transport trajectories. After training an initial flow, each source sample is re-coupled with the endpoint generated by the initial flow model under the same condition, and the model is retrained on the resulting pairs using the same linear-path objective. Straighter trajectories reduce numerical integration error and enable accurate sampling with fewer solver steps. Our CFG-aware reflow implementation is detailed in Appendix A.5.

3.2 MOTION LATENT CODEC (MLC)

We represent motion using 22 SMPL joints (Loper et al., 2015) and the 263-dimensional kinematic representation introduced by Guo et al. (2022), consisting of root angular and planar velocities, root height, local joint positions, 6D joint rotations, local joint velocities, and foot-contact labels. We use fixed-length sequences of $T = 196$ frames for motion sequences (Chen et al., 2023).

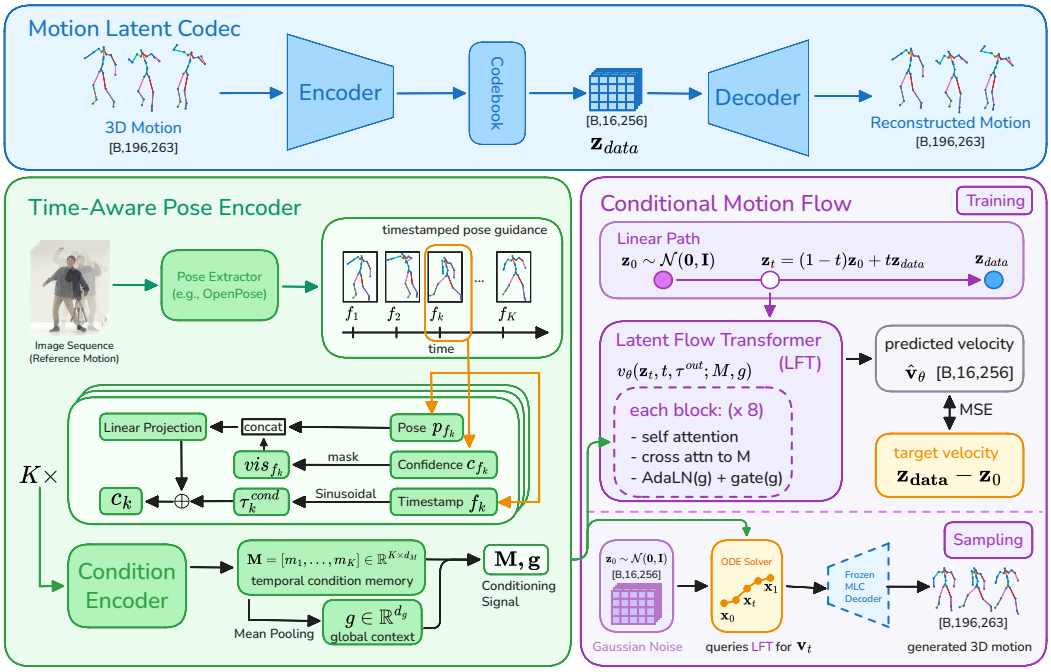

Figure 2: Architecture overview. The system consists of three main components: the Motion Latent Codec (blue) encodes 3D motion into a compact discrete latent representation and decodes generated latents back into motion; the Time-Aware Pose Encoder (green) extracts sparse 2D pose observations and corresponding timestamps from demonstration videos and encodes them into temporally informed conditioning signals; and the Conditional Motion Flow (purple) learns a conditional flow from noise to motion latents, enabling efficient generation guided by the encoded condition signals.

A Transformer-based VQ-VAE (van den Oord et al., 2017) compresses each $\mathbf{x} \in \mathbb{R}^{196\times263}$ into 16 latent tokens. Each frame is projected to 256 dimensions, 16 learned latent tokens are prepended, and the sequence is processed by a Transformer encoder $\mathcal{E}$. The 16 token outputs form $\mathbf{z}_e \in \mathbb{R}^{16\times256}$, which is quantized to the nearest entry of an EMA-updated codebook $\mathrm{E} \in \mathbb{R}^{1024\times256}$, yielding $\mathbf{z}_q \in \mathbb{R}^{16\times256}$. A straight-through estimator propagates gradients through quantization, and the Transformer decoder concatenates 196 learned frame queries with the 16 latent tokens and jointly processes them using full self-attention, after which the frame-query states are projected to reconstruct the 196×263 motion sequence.

The MLC is trained with feature-space reconstruction, joint-space reconstruction, temporal consistency, and VQ commitment objectives; full loss definitions and weights are provided in Appendix A.2. The commitment term $\mathcal{L}_{\text{commit}} = \beta \|\mathbf{z}_e - \mathrm{sg}(\mathbf{z}_q)\|_2^2$ encourages encoder outputs to remain close to their selected codebook entries, preventing the encoder representation from drifting away from the discrete latent space; $\mathrm{sg}(\cdot)$ denotes stop-gradient operation. After pretraining, the MLC is frozen. Quantized latents are normalized using training set statistics, $\mathbf{z} = (\mathbf{z}_q - \boldsymbol{\mu}_z)/\boldsymbol{\sigma}_z$, and the normalized 16×256 representation defines the data space for the flow model in the CMF module (Sec. 3.4). Generated latents are denormalized and decoded to obtain the final motion.

3.3 Time-Aware Pose Encoder (TAPE)

Given input images or video frames, we extract 2D human poses using OpenPose (Cao et al., 2021). For each frame f, OpenPose provides $J = 25$ joints with image-plane coordinates $\mathbf{p}_{f,j} \in \mathbb{R}^2$ and confidence $c_{f,j} \in [0, 1]$. Low-confidence joints are masked with a binary visibility indicator and their coordinates are set to zero. During training, each joint is additionally dropped with probability $p_{\text{joint}} = 0.05$ to improve robustness to occlusion and incomplete detections.

To reduce variation caused by image translation and body scale while preserving relative image-plane motion, poses are normalized using a fixed sequence-level root reference and a per-frame

body scale, followed by per-joint normalization using training set statistics. To make each pose observation time-aware, we additionally assign it a temporal location within the motion sequence. For a motion sequence of length T, we select K reference frames. If $f_k \in \{0, \dots, T-1\}$ is the frame index of observation k, its normalized temporal location is $\tau_k = f_k/(T-1) \in [0,1]$.

For each observation, the normalized 2D joint coordinates and visibility indicators are flattened into a 75-dimensional pose feature $\mathbf{u}_k$. The feature is projected to the Transformer hidden dimension by a learned linear projection $\mathbf{W}_c$ and explicitly combined with a sinusoidal timestamp encoding,

$$\mathbf{c}_k = \mathbf{W}_c \mathbf{u}_k + \mathbf{e}_\tau(\tau_k). \tag{2}$$

The sequence $[\mathbf{c}_1, \dots, \mathbf{c}_K]$ is processed by a Transformer encoder with hidden dimension $d = 512$, producing time-aware condition memory $\mathbf{M} \in \mathbb{R}^{K \times 512}$. Self-attention allows information exchange across all selected reference poses while each memory token remains tied to its timestamp. Mean-pooling over $\mathbf{M}$ gives a sequence-level global context $\mathbf{g} \in \mathbb{R}^{512}$. Thus, $\mathbf{M}$ provides local, time-specific pose constraints, while $\mathbf{g}$ summarizes the overall conditioning sequence.

3.4 Conditional Motion Flow (CMF)

We parameterize the conditional velocity field as $\mathbf{v}_\theta(\mathbf{z}_t, t, \boldsymbol{\tau}^{\text{out}}; \mathbf{M}, \mathbf{g})$ using a Transformer operating on the 16 motion latent slots. We assign the i-th latent slot a normalized temporal location $\tau_i^{\text{out}} = i/(L-1)$, where $L = 16$. Each latent token is embedded as

$$\mathbf{h}_i = \mathbf{W}_z \mathbf{z}_t^{(i)} + \mathbf{e}_{\text{slot}}(\tau_i^{\text{out}}) + \mathbf{e}_t(t) \in \mathbb{R}^{512}. \tag{3}$$

Here, $\mathbf{W}_z$ projects each motion latent slot into the Transformer feature space, $\mathbf{e}_{\text{slot}}(\tau_i^{\text{out}})$ encodes its temporal position and $\mathbf{e}_t(t)$ encodes the current flow time. Each Transformer block first applies self-attention across the 16 latent tokens, then injects the time-aware condition memory $\mathbf{M}$ through cross-attention. In parallel, the global context $\mathbf{g}$ modulates the latent features through adaptive layer normalization (AdaLN) and a learned conditioning gate:

$$\mathbf{a} = \text{sigmoid}(\mathbf{W}_g \mathbf{g}), \qquad \mathbf{h} \leftarrow \mathbf{h} + \mathbf{a} \odot \text{CrossAttn}\big(\text{AdaLN}(\mathbf{h}; \mathbf{g}), \mathbf{M}\big), \tag{4}$$

where $\mathbf{W}_g$ is a learned projection from the global context to the feature-wise gating coefficients. Each block is followed by a feed-forward network and residual connections. The final features are projected to $\hat{\mathbf{v}}_t \in \mathbb{R}^{16 \times 256}$, yielding the conditional velocity prediction.

3.5 Training and Inference

Training. We optimize CMF with a weighted combination of the standard Flow Matching objective and a reprojection loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{flow}} + \lambda_{\text{reproj}} \mathcal{L}_{\text{reproj}}. \tag{5}$$

Following standard Flow Matching (Lipman et al., 2023), the linear path defined in Sec. 3.1 gives

$$\mathcal{L}_{\text{flow}} = \mathbb{E}_{\mathbf{z}_0, \mathbf{z}_1, t} \left[\left\| \mathbf{v}_\theta(\mathbf{z}_t, t, \boldsymbol{\tau}^{\text{out}}; \mathbf{M}, \mathbf{g}) - (\mathbf{z}_1 - \mathbf{z}_0) \right\|_2^2 \right]. \tag{6}$$

We additionally use $\mathcal{L}_{\text{reproj}}$ to enforce consistency with the sparse visual observations. Starting from a Gaussian source $\mathbf{z}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, we integrate the predicted vector field from $t = 0$ to $t = 1$ with a differentiable ODE solver and decode the resulting latent using the frozen MLC. The generated 3D motion is mapped to 14 BODY25-compatible joints and projected to the image plane using a sequence-level weak-perspective camera fitted on the conditioning frames. We then minimize the confidence-weighted 2D joint error at these frames. Its detailed formulation and weight ablation are provided in Appendix A.4.

To enable controllable conditioning strength at inference, we adopt classifier-free guidance (CFG) (Ho & Salimans, 2021). During training, the visual conditioning is independently replaced for each training example by learned null embeddings representing the unconditional condition, $(\mathbf{M}_\varnothing, \mathbf{g}_\varnothing)$. The null rate is linearly warmed up from 0 to $p_{\text{null}} = 0.05$ over the first 5 epochs and remains fixed thereafter. We also employ a condition-density curriculum, where the number of reference observations K gradually decreases from dense conditioning to a mixture of sparse settings. After initial flow training, we perform one additional reflow stage following the rectification principle in Sec. 3.1; our CFG-aware reflow implementation is detailed in Appendix A.5.

Inference. Generation starts from $\mathbf{z}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Given guidance scale s, the conditional and unconditional predictions are combined as

$$\mathbf{v}_{\mathrm{CFG}} = \mathbf{v}_{\varnothing} + s\big(\mathbf{v}_{\mathrm{cond}} - \mathbf{v}_{\varnothing}\big), \tag{7}$$

and the resulting ODE is integrated from $t = 0$ to $t = 1$ using the Heun solver. After one round of rectification, the resulting CMF module supports generation with a single solver step. The endpoint $\hat{\mathbf{z}}_1 \in \mathbb{R}^{16\times 256}$ is denormalized using the latent statistics in Sec. 3.2 and decoded by the frozen MLC module to obtain the final 196-frame motion sequence.

4 Experiments

We evaluate FlowMimic on the AIST++ dataset along three aspects: generation quality and physical plausibility, fidelity to the visual conditions, and inference efficiency.

4.1 Experimental Setup

4.1.1 Datasets

We evaluate FlowMimic on the AIST++ (Li et al., 2021) dance dataset, which provides synchronized multi-view RGB videos and 3D motion annotations. Dance motions provide a suitable testbed for visual demonstration-based control, as their fine-grained movements and temporal dynamics are difficult to specify with natural language alone. During training, frontal, left-frontal, right-frontal, and low-angle frontal views are treated as independent visual-condition samples. For text-conditioned baselines, Qwen3-VL-32B-Instruct (Bai et al., 2025) generates diverse textual descriptions from the dance videos. MVHumanNet (Xiong et al., 2024) is combined with AIST++ for MLC pretraining.

4.1.2 Baselines and Evaluation Protocol

We compare with MLD (Chen et al., 2023), FloodDiffusion (Cai et al., 2026), MotionHiFlow (Li et al., 2026), StickMotion (Wang et al., 2025), OmniControl (Xie et al., 2024) and CondMDI (Cohan et al., 2024). The first three are text-conditioned, StickMotion adds stickman sketches, OmniControl combines text with spatiotemporal joint control, and CondMDI performs motion in-betweening from text and sparse full-body 3D keyframes. All methods use the same AIST++ split and generate 30-fps motion. VAE-based models use the combined AIST++–MVHumanNet dataset for VAE pretraining. FlowMimic is evaluated using only the frontal view for a matched protocol.

For FlowMimic, we evaluate sparse ($K = 7$) and dense ($K = 196$) conditioning at evenly spaced timestamps, while OmniControl and CondMDI use the sparse setting only. Conditioning signals differ across methods: FlowMimic extracts full-body 2D poses from demonstration videos, OmniControl uses text with one-joint 3D control, CondMDI uses text with target-derived full-body 3D keyframes, and StickMotion uses three target-motion sketches. The same $K = 7$ partition is used for baseline RPD. Results are averaged over 10 matched-seed runs with 95% confidence intervals.

4.1.3 Evaluation Metrics

Generation Quality and Physical Plausibility. We evaluate generation quality using Fréchet Inception Distance (FID) and Diversity with a T2M motion-feature extractor (Guo et al., 2022) trained on AIST++ and frozen for evaluation. FID measures distributional distance between generated and real motions in the feature space, while Diversity measures variation among generated motions and is compared with the Real reference. We further assess motion dynamics and physical plausibility using Acceleration P90, the 90th percentile of global acceleration magnitude across 22 joints, and foot skating, measured as mean ankle speed during predicted foot contact. The former captures high-magnitude motion dynamics, and lower foot skating indicates more stable foot–ground contact.

Condition Fidelity. Paired Motion Feature Distance (PMFD) measures the mean L_2 distance between each generated motion and its paired target in the same T2M feature space, with lower values indicating greater motion-level similarity. Mean Per Joint Position Error (MPJPE) measures global 3D joint-position error against the paired target motion, reported in millimeters. We report full-sequence MPJPE in the main comparison and MPJPE on observed condition frames, denoted

$\mathrm{MPJPE_o}$, in conditioning analyses. Reprojection Distance (RPD) measures frame-level geometric alignment between generated 3D joints and the corresponding 2D demonstration poses, where lower is better. Let $\mathcal{T}_{\mathrm{all}} = \{0, \ldots, T-1\}$ denote all sequence frames, $\mathcal{C} \subset \mathcal{T}_{\mathrm{all}}$ the conditioning frames, and $\mathcal{J}$ the generated joints. For any frame set $\mathcal{S} \subseteq \mathcal{T}_{\mathrm{all}}$, we define

$$\mathrm{RPD}(\mathcal{S}) = \frac{1}{|\mathcal{S}|} \sum_{t \in \mathcal{S}} \frac{\sum_{j \in \mathcal{J}} c_{t,j} \left\| \Pi_{\hat{\phi}}(\hat{\mathbf{J}}_{t,j}) - \mathbf{p}_{t,j} \right\|_2}{\sum_{j \in \mathcal{J}} c_{t,j}}. \tag{8}$$

Here, $c_{t,j}$ denotes the OpenPose confidence of BODY25 joint j at frame t. The weak-perspective camera $\hat{\phi} = (s, b_x, b_y)$, with scale s and image-plane translation (b_x, b_y), is fitted once per sequence using only $\mathcal{C}$, without rotational alignment, and fixed for all evaluation frames. We report

$$\mathrm{RPD}_{\mathrm{unobserved}} = \mathrm{RPD}(\mathcal{T}_{\mathrm{all}} \setminus \mathcal{C}), \qquad \mathrm{RPD}_{\mathrm{all}} = \mathrm{RPD}(\mathcal{T}_{\mathrm{all}}). \tag{9}$$

These measure pose fidelity on non-conditioning frames and the complete sequence, respectively.

Inference Efficiency. We report end-to-end latency and solver-level Number of Function Evaluations (NFE) to characterize the computational cost of motion generation. NFE measures the number of guided denoiser or velocity-field evaluations required for sampling, while end-to-end latency measures the corresponding practical runtime. Runtime is measured on one NVIDIA RTX A6000, excluding data loading but including condition encoding, sampling, and latent decoding.

4.1.4 Implementation Details

Both MLC and initial CMF use AdamW with a learning rate of 2×10^{-4} and weight decay 0.01. MLC is trained with a global batch size of 96 for 92,400 updates and then frozen. CMF is trained on random 196-frame AIST++ clips with a global batch size of 896 for approximately 69,200 initial-flow updates. The condition-density curriculum uses $K \in \{196, 98, 49, 28, 14, 7\}$ with even, random, and boundary-gap sampling. We use CFG scale 5.0 and reprojection-loss weight $\lambda = 0.001$. Reflow rectification follows Appendix A.5 and is trained for 33,334 updates. The full training pipeline takes about 70 hours on two NVIDIA RTX A6000 GPUs with 48GB of VRAM each.

4.2 Quantitative Results

As shown in Table 1, FlowMimic preserves Diversity close to Real and substantially reduces foot skating, indicating preserved motion variation and more stable foot–ground contact. FID is computed in the T2M feature space, which primarily captures high-level semantics and has limited

Table 1: Quantitative comparison on AIST++. MHF, SM, MLD, FD, OC and CMDI denote MotionHiFlow (Li et al., 2026), StickMotion (Wang et al., 2025), Motion Latent Diffusion (Chen et al., 2023), FloodDiffusion (Cai et al., 2026), OmniControl (Xie et al., 2024), and CondMDI (Cohan et al., 2024) respectively. S/D denote sparse ($K = 7$)/dense ($K = 196$) conditioning. Div., Accel., and FS denote Diversity, Acceleration P90 ($\mathrm{m/s^2}$), and foot-skating speed ($\mathrm{mm/s}$). $\mathrm{RPD}_{u/a}$ denote RPD on unobserved/all frames; for RPD_u of methods without sparse visual conditions, the same $K = 7$ frame partition is used. $\downarrow$ indicates lower is better and $\rightarrow$ indicates closer to Real is better. **Bold** indicates the best generated result.

Method	FID $\downarrow$	Div. $\rightarrow$	Accel. $\rightarrow$	FS $\downarrow$	PMFD $\downarrow$	MPJPE $\downarrow$	$\mathrm{RPD}_u \downarrow$	$\mathrm{RPD}_a \downarrow$
Real-S	0.00	$8.63^{\pm 0.09}$	$28.11^{\pm 0.00}$	$426^{\pm 0}$	0.00	0.00	$21.74^{\pm 0.00}$	$21.74^{\pm 0.00}$
Real-D	0.00	$8.63^{\pm 0.09}$	$28.11^{\pm 0.00}$	$426^{\pm 0}$	0.00	0.00	–	$21.52^{\pm 0.00}$
MHF	$\mathbf{1.01^{\pm 0.08}}$	$9.43^{\pm 0.16}$	$40.47^{\pm 0.48}$	$1172^{\pm 14}$	$4.83^{\pm 0.05}$	$571.98^{\pm 5.02}$	$73.47^{\pm 0.62}$	$73.46^{\pm 0.62}$
SM	$1.89^{\pm 0.11}$	$10.00^{\pm 0.14}$	$80.81^{\pm 9.91}$	$1474^{\pm 150}$	$4.63^{\pm 0.07}$	$645.88^{\pm 36.23}$	$72.31^{\pm 0.33}$	$72.31^{\pm 0.33}$
MLD	$2.05^{\pm 0.27}$	$9.87^{\pm 0.12}$	$41.79^{\pm 4.97}$	$923^{\pm 55}$	$6.02^{\pm 0.09}$	$843.92^{\pm 69.43}$	$76.17^{\pm 0.31}$	$76.15^{\pm 0.31}$
FD	$6.66^{\pm 0.25}$	$9.02^{\pm 0.15}$	$137.18^{\pm 13.34}$	$1880^{\pm 161}$	$6.89^{\pm 0.08}$	$1119.99^{\pm 89.19}$	$74.98^{\pm 0.18}$	$74.99^{\pm 0.18}$
OC	$1.81^{\pm 0.18}$	$10.01^{\pm 0.13}$	$38.68^{\pm 10.23}$	$1661^{\pm 157}$	$5.15^{\pm 0.06}$	$609.21^{\pm 48.30}$	$87.83^{\pm 0.72}$	$87.81^{\pm 0.72}$
CMDI	$1.75^{\pm 0.14}$	$9.47^{\pm 0.10}$	$125.73^{\pm 1.73}$	$1862^{\pm 19}$	$\mathbf{3.36^{\pm 0.03}}$	$397.81^{\pm 3.34}$	$56.20^{\pm 0.20}$	$56.05^{\pm 0.20}$
Ours-D	$3.12^{\pm 0.07}$	$8.81^{\pm 0.12}$	$\mathbf{17.99^{\pm 0.84}}$	$562^{\pm 7}$	$3.62^{\pm 0.02}$	$\mathbf{322.30^{\pm 3.55}}$	–	$\mathbf{41.85^{\pm 0.19}}$
Ours-S	$3.23^{\pm 0.10}$	$\mathbf{8.47^{\pm 0.12}}$	$15.82^{\pm 1.64}$	$\mathbf{541^{\pm 24}}$	$4.10^{\pm 0.04}$	$380.30^{\pm 18.07}$	$\mathbf{53.23^{\pm 0.30}}$	$53.12^{\pm 0.30}$

sensitivity to fine-grained motion details such as jitter and foot sliding. Thus, the lower FID reflects a smaller distribution-level discrepancy in the T2M feature space, but does not directly indicate better fine-grained motion realism. For conditional fidelity, FlowMimic achieves substantially lower PMFD and RPD than the text-conditioned baselines and OmniControl. CondMDI obtains lower PMFD, but requires exact full-body 3D keyframes from the target motion as input, which provide stronger target-specific conditioning for this paired metric than FlowMimic's image-derived 2D poses. Since such accurate 3D spatial information is difficult to obtain in real-world applications and typically requires multi-view reconstruction, Appendix C.1 further evaluates CondMDI using monocularly estimated 3D conditions from the demonstration videos. Despite the strong conditioning of CondMDI, FlowMimic achieves lower MPJPE and lower RPD on unobserved and all frames. The gains on $\mathrm{RPD}_{\mathrm{unobserved}}$ also suggest that sparse pose observations guide the full trajectory beyond the conditioning timestamps. Finally, Table 2 shows that FlowMimic generates motion with a single Heun step (2 NFEs) in about 31 ms per sequence, substantially reducing inference latency.

Table 2: Inference time on AIST++ (ms/sequence). NFE counts solver-level guided denoiser or vector-field evaluations; timing includes internal classifier-free guidance branches. Runtime is measured on an NVIDIA RTX A6000 GPU for 196-frame motions. **Bold** denotes the best results.

Method	Steps	NFE ↓	Time (ms) ↓
MLD	50 DDIM	50	$177^{\pm 13}$
MotionHiFlow	20 Flow	23	$293^{\pm 27}$
StickMotion	50 DDIM	50	$725^{\pm 84}$
FloodDiffusion	10 subdivisions	108	$3{,}044^{\pm 41}$
OmniControl	1000 DDPM	1000	$48{,}710^{\pm 560}$
CondMDI	1000 DDPM	1000	$1{,}861^{\pm 1}$
Ours	1 Heun	2	$\mathbf{31}^{\pm \mathbf{1}}$

4.3 Ablation and Robustness Studies

Effect of CFG Scale. Fig. 3 shows that FID, PMFD, and $\mathrm{RPD}_{\mathrm{all}}$ improve as the CFG scale increases before saturating around 5.0; we therefore use 5.0 for reflow training and the main evaluations.

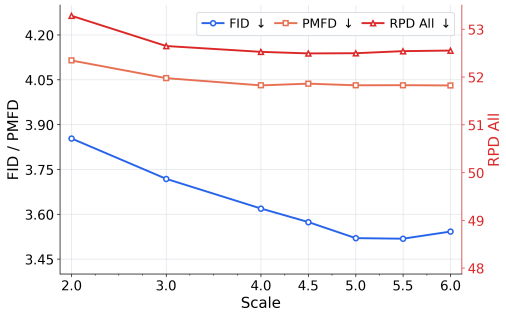

Figure 3: Effect of CFG scale on FID, PMFD, and $\mathrm{RPD}_{\mathrm{all}}$. FID and PMFD use the left axis, while $\mathrm{RPD}_{\mathrm{all}}$ uses the right axis; evaluated on initial flow; lower is better.

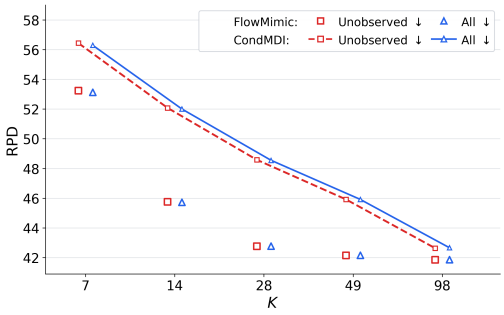

Figure 4: RPD at different K settings. The two markers represent $\mathrm{RPD}_{\mathrm{unobs.}}$ and $\mathrm{RPD}_{\mathrm{all}}$, respectively. The lines show the corresponding results of CondMDI (Cohan et al., 2024).

Robustness to Sparse Conditioning. As shown in Fig. 4, increasing K improves unobserved-frame and overall fidelity, with diminishing gains at higher densities. We omit $K = 196$ because all frames are observed and RPD_u is therefore undefined. FlowMimic remains below CondMDI across all tested K, demonstrating effective control under highly sparse conditioning.

Robustness to Condition Sampling Patterns. We evaluate $K = 7$ under even, random, and boundary-gap sampling, where the last preserves the endpoints while leaving a contiguous 25%–50% interval unobserved. Table 3 shows that even sampling gives the best full-sequence fidelity, while random and boundary-gap sampling mainly degrade the unobserved frames. Meanwhile,

Table 3: Robustness to condition sampling patterns under sparse conditioning ($K = 7$). Div., Accel., and FS denote Diversity, acceleration P90 ($\mathrm{m/s^2}$), and foot-skating speed ($\mathrm{mm/s}$), respectively. RPD_u and RPD_a denote RPD on unobserved and all frames. MPJPE_o denotes MPJPE on the observed condition frames, reported in mm. B-gap denotes boundary-gap sampling. $\downarrow$ indicates lower is better and $\rightarrow$ indicates closer to Real is better. **Bold** denotes the best result for each metric.

Pattern	FID $\downarrow$	Div. $\rightarrow$	Accel. $\rightarrow$	FS $\downarrow$	PMFD $\downarrow$	$\mathrm{MPJPE}_o \downarrow$	$\mathrm{RPD}_u \downarrow$	$\mathrm{RPD}_a \downarrow$
Real	0.00	$8.63^{\pm 0.09}$	$28.11^{\pm 0.00}$	$426^{\pm 0}$	0.00	0.00	$21.74^{\pm 0.00}$	$21.74^{\pm 0.00}$
Even	$\mathbf{3.23^{\pm 0.10}}$	$\mathbf{8.47^{\pm 0.12}}$	$15.82^{\pm 1.64}$	$\mathbf{541^{\pm 24}}$	$\mathbf{4.10^{\pm 0.04}}$	$364.94^{\pm 13.66}$	$\mathbf{53.23^{\pm 0.30}}$	$\mathbf{53.12^{\pm 0.30}}$
Random	$3.34^{\pm 0.08}$	$8.38^{\pm 0.12}$	$15.12^{\pm 1.27}$	$542^{\pm 19}$	$4.30^{\pm 0.04}$	$388.94^{\pm 14.18}$	$56.21^{\pm 0.30}$	$55.91^{\pm 0.29}$
B-gap	$3.37^{\pm 0.07}$	$8.45^{\pm 0.12}$	$\mathbf{16.01^{\pm 1.77}}$	$545^{\pm 19}$	$4.29^{\pm 0.05}$	$\mathbf{363.18^{\pm 11.66}}$	$59.78^{\pm 0.28}$	$59.35^{\pm 0.28}$

$\mathrm{MPJPE_o}$ remains broadly comparable, suggesting that irregular observations mainly make unconstrained motion harder to interpolate rather than limiting pose adherence.

Ablation of Pose Conditioning Paths. Table 4 examines the contributions of temporal condition memory $\mathbf{M}$ and global context $\mathbf{g}$. Both conditioning paths substantially improve over the no-pose setting, with $\mathbf{M}$ providing the stronger contribution to pose fidelity. Adding g yields modest additional gains overall, with the clearest improvement under random sampling. This suggests that global context provides limited additional benefit for regular sampling but can improve robustness to some irregular sampling patterns, particularly random sampling.

Additional analyses of MLC latent capacity, solver and reflow behavior, robustness to pose extraction and camera view are provided in Appendix C.

Table 4: Ablation of TAPE conditioning paths under sparse conditioning ($K = 7$). *No cond.* indicates that no condition signal is passed to the model. $\mathbf{M}$ denotes temporal condition memory and $\mathbf{g}$ denotes global context. MPJPE_o denotes MPJPE on observed condition frames. All ablation variants, including the $M{+}g$ branch, are measured using the initial trained flow, without reflow rectification. **Bold** indicates the best result within each sampling pattern.

Pattern	Cond. Path	PMFD $\downarrow$	$\mathrm{MPJPE}_o \downarrow$	$\mathrm{RPD}_{\mathrm{unobs.}} \downarrow$
Even	*No cond.*	$7.13^{\pm 0.06}$	$445.07^{\pm 1.14}$	$64.32^{\pm 0.18}$
	g-only	$4.10^{\pm 0.03}$	$391.71^{\pm 1.27}$	$58.88^{\pm 0.27}$
	M-only	$\mathbf{3.88^{\pm 0.02}}$	$364.99^{\pm 2.38}$	$\mathbf{52.63^{\pm 0.29}}$
	$M{+}g$	$3.95^{\pm 0.04}$	$\mathbf{351.35^{\pm 2.12}}$	$52.75^{\pm 0.24}$
Random	*No cond.*	$7.13^{\pm 0.06}$	$460.64^{\pm 1.30}$	$65.58^{\pm 0.18}$
	g-only	$4.27^{\pm 0.04}$	$406.22^{\pm 2.32}$	$59.85^{\pm 0.20}$
	M-only	$4.13^{\pm 0.03}$	$393.43^{\pm 2.00}$	$56.16^{\pm 0.35}$
	$M{+}g$	$\mathbf{4.09^{\pm 0.05}}$	$\mathbf{373.46^{\pm 1.05}}$	$\mathbf{55.57^{\pm 0.21}}$
Boundary-gap	*No cond.*	$7.13^{\pm 0.06}$	$430.76^{\pm 1.66}$	$67.90^{\pm 0.21}$
	g-only	$4.24^{\pm 0.02}$	$384.58^{\pm 1.26}$	$63.68^{\pm 0.31}$
	M-only	$\mathbf{4.14^{\pm 0.03}}$	$\mathbf{352.87^{\pm 2.11}}$	$59.47^{\pm 0.37}$
	$M{+}g$	$4.19^{\pm 0.04}$	$359.27^{\pm 2.16}$	$\mathbf{59.45^{\pm 0.34}}$

5 Conclusion

In this work, we introduce FlowMimic, a visually conditioned framework for 3D human motion generation that requires only a demonstration video as input. Visual demonstrations naturally capture detailed motion patterns that are difficult to specify precisely with text, making them particularly suitable for fine-grained control of complex motions such as dance. FlowMimic uses sparse, timestamped 2D pose observations extracted from the input video as the sole conditioning signal, without auxiliary text or manually specified pose guidance. To our knowledge, FlowMimic is the first framework to generate complete 3D human motion from sparse visual demonstrations while enabling efficient single-step generation through Flow Matching. Experiments on AIST++ show that FlowMimic achieves strong conditional fidelity while maintaining physically plausible motion.

AI USE STATEMENT

In this work, we used generative AI tools to assist with textual label generation and preprocessing for the AIST++ dataset, code prototyping and editing, and proofreading and language polishing of the manuscript. Generative AI was not used to formulate the research hypotheses, design the methodology or experiments, interpret or analyze experimental results, or develop the scientific claims and conclusions of the paper. The technical content and core manuscript were developed by the authors. All AI-assisted code was manually reviewed and tested for correctness, dataset-related outputs were checked before use, and all AI-assisted textual edits were reviewed and revised by the authors to ensure technical accuracy and consistency with the intended meaning. We take full responsibility for the final content of this work, including text, claims, code and data artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Detailed implementation, training, preprocessing, baseline construction, and evaluation settings are provided in the appendix. Additional qualitative results and motion visualizations are available on our anonymous project page at `https://iclr-demo.pages.dev`. Source code will be publicly released.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

German Barquero, Sergio Escalera, and Cristina Palmero. Seamless human motion composition with blended positional encodings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 457–469, 2024.

Valentin Bazarevsky, Ivan Grishchenko, Karthik Raveendran, Tyler Zhu, Fan Zhang, and Matthias Grundmann. Blazepose: On-device real-time body pose tracking. *arXiv preprint arXiv:2006.10204*, 2020.

Yiyi Cai, Yuhan Wu, Kunhang Li, You Zhou, Bo Zheng, and Haiyang Liu. Flooddiffusion: Tailored diffusion forcing for streaming motion generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2295–2304, 2026.

Manolo Canales Cuba, Vinícius do Carmo Melício, and João Paulo Gois. Flowmotion: Target-predictive conditional flow matching for jitter-reduced text-driven human motion generation. *Computers & Graphics*, 132:104374, 2025. doi: 10.1016/j.cag.2025.104374.

Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 43(01):172–186, 2021. doi: 10.1109/TPAMI.2019.2929257.

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 (CVPR)*, pp. 18000–18010, 2023.

Setareh Cohan, Guy Tevet, Daniele Reda, Xue Bin Peng, and Michiel van de Panne. Flexible motion in-betweening with diffusion models. In *ACM SIGGRAPH 2024 Conference Papers*, 2024.

Yuming Du, Robin Kips, Albert Pumarola, Sebastian Starke, Ali Thabet, and Artsiom Sanakoyeu. Avatars grow legs: Generating smooth human motion from sparse tracking inputs with diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 481–490, 2023.

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 (CVPR)*, pp. 5152–5161, 2022.

Haonan Han, Xiangzuo Wu, Huan Liao, Zunnan Xu, Zhongyuan Hu, Ronghui Li, Yachao Zhang, and Xiu Li. Atom: Aligning text-to-motion model at event-level with gpt-4vision reward. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22746–22755, 2025.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

Vincent Tao Hu, Wenzhe Yin, Pingchuan Ma, Yunlu Chen, Basura Fernando, Yuki M Asano, Efstratios Gavves, Pascal Mettes, Bjorn Ommer, and Cees G. M. Snoek. Motion flow matching for human motion synthesis and editing. *arXiv preprint arXiv:2312.08895*, 2023.

Peng Jin, Yang Wu, Yanbo Fan, Zhongqian Sun, Yang Wei, and Li Yuan. Act as you wish: Fine-grained control of motion diffusion model with hierarchical semantic graphs. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Korrawe Karunratanakul, Konpat Preechakul, Supasorn Suwajanakorn, and Siyu Tang. Guided motion diffusion for controllable human motion synthesis. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2151–2162, 2023.

Heng Li, Xiaotong Lin, Ling-An Zeng, Yulei Kang, Shuai Li, and Jian-Fang Hu. Motionhiflow: Text-to-motion via hierarchical flow matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9352–9363, 2026.

Ronghui Li, YuXiang Zhang, Yachao Zhang, Hongwen Zhang, Jie Guo, Yan Zhang, Yebin Liu, and Xiu Li. Lodge: A coarse to fine diffusion network for long dance generation guided by the characteristic dance primitives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1524–1534, June 2024.

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 (ICCV)*, pp. 13401–13412, 2021.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Smpl: a skinned multi-person linear model. *ACM Trans. Graph.*, 34(6), 2015. doi: 10.1145/2816795.2818013.

Matthias Plappert, Christian Mandery, and Tamim Asfour. The kit motion-language dataset. *Big Data*, 4(4):236–252, 2016. doi: 10.1089/big.2016.0028. PMID: 27992262.

Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-Or, and Amit Haim Bermano. Human motion diffusion model. In *The Eleventh International Conference on Learning Representations*, 2023.

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 (CVPR)*, pp. 448–458, 2023.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Tao Wang, Zhihua Wu, Qiaozhi He, Jiaming Chu, Ling Qian, Yu Cheng, Junliang Xing, Jian Zhao, and Lei Jin. Stickmotion: Generating 3d human motions by drawing a stickman. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pp. 12370–12379, 2025.

Yiming Xie, Varun Jampani, Lei Zhong, Deqing Sun, and Huaizu Jiang. Omnicontrol: Control any joint at any time for human motion generation. In *The Twelfth International Conference on Learning Representations*, 2024.

Zhangyang Xiong, Chenghong Li, Kenkun Liu, Hongjie Liao, Jianqiao Hu, Junyi Zhu, Shuliang Ning, Lingteng Qiu, Chongjie Wang, Shijie Wang, Shuguang Cui, and Xiaoguang Han. Mvhumannet: A large-scale dataset of multi-view daily dressing human captures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19801–19811, 2024.

Mingyuan Zhang, Xinying Guo, Liang Pan, Zhongang Cai, Fangzhou Hong, Huirong Li, Lei Yang, and Ziwei Liu. Remodiffuse: Retrieval-augmented motion diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 364–373, 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 & Machine Intelligence*, 46(06):4115–4128, 2024. doi: 10.1109/TPAMI.2024.3355414.

A Additional Implementation Details

A.1 Dataset Preprocessing and Motion Representation

AIST++. The 3D target motion is obtained from the released AIST++ SMPL annotations. We load the SMPL pose, translation, and scaling parameters, run the male SMPL model with the default body shape, and retain the first 22 joints. The pelvis trajectory is corrected for the released scale factor and expressed relative to the first pelvis position. AIST++ motion and the camera-specific OpenPose detections are treated as 60-fps sequences and converted to 30 fps by retaining every second frame. During training, long sequences are randomly cropped to 196 consecutive frames; shorter sequences are zero-padded and accompanied by a valid-frame mask. Matched benchmark evaluation uses the deterministic first 196-frame crop.

MVHumanNet. For MVHumanNet, we directly use the first 22 joints from the released per-frame joint arrays. Before common canonicalization, we apply fixed dataset-specific axis transforms, $(x, y, z) \mapsto (x, -z, y)$ for AIST++ and $(x, y, z) \mapsto (-y, -x, -z)$ for MVHumanNet. The source 5-fps MVHumanNet motion is then upsampled to 30 fps with PCHIP interpolation independently over joint coordinates. MVHumanNet is used only for MLC pretraining.

Common motion representation. Both datasets are converted to the same 263-D HumanML3D-style representation. Before feature extraction, each sequence is placed on the floor, horizontally centered at its initial pelvis, and rotated such that the initial body-facing direction points toward the canonical forward axis. The 263-D vector contains 4 root-motion channels, 63 root-invariant joint-position channels, 126 6D rotation channels, 66 local joint-velocity channels, and 4 foot-contact channels. The conversion produces transition features for adjacent frames and repeats the final row so that 196 SMPL22 frames remain a 196×263 sequence. The first 259 continuous channels are normalized with statistics computed from all valid training frames of the combined AIST++–MVHumanNet MLC training set; the four binary contact channels are not standardized.

A.2 Motion Latent Codec Training

The MLC uses 16 latent tokens of dimension 256 and a 1,024-entry EMA codebook. Each encoder token is quantized to its nearest codebook entry, and decoder gradients are passed to the encoder with

Table 5: Loss weights used for MLC pretraining. Temporal weights are linearly ramped from zero to their listed final values during the last 20% of training.

Group	Loss term	Weight
Feature-space	Continuous reconstruction $\mathcal{L}_{\text{cont}}$	1.0
	Foot contact $\mathcal{L}_{\text{contact}}$	1.0
	Additional root reconstruction $\mathcal{L}_{\text{root}}$	2.0
Joint-space	Full-joint reconstruction $\mathcal{L}_{\text{joint}}$	0.5
	Distal-joint reconstruction $\mathcal{L}_{\text{distal}}$	1.0
	Root trajectory $\mathcal{L}_{\text{root-traj}}$	1.0
	Terminal root drift $\mathcal{L}_{\text{root-drift}}$	1.0
	Distal-joint velocity $\mathcal{L}_{\text{distal-vel}}$	1.0
Quantization	VQ commitment	0.25
Temporal	Velocity matching $\mathcal{L}_{\text{vel}}$	0.25
	Acceleration matching $\mathcal{L}_{\text{acc}}$	0.02

the straight-through estimator. The codebook is updated with EMA decay 0.99, while the encoder is regularized by the commitment objective

$$\mathcal{L}_{\text{VQ}} = \beta_{\text{VQ}} \, \text{MSE}\big(\mathbf{z}_e, \text{sg}(\mathbf{z}_q)\big), \quad \beta_{\text{VQ}} = 0.25. \tag{10}$$

The reconstruction objective combines feature-space and joint-space supervision. Most regression terms use Smooth-L1 loss with $\beta = 1$ and are averaged over valid frames only. In the 263-D motion feature space, we use

$$\mathcal{L}_{\text{recon}} = \mathcal{L}_{\text{cont}} + \mathcal{L}_{\text{contact}} + 2\mathcal{L}_{\text{root}}, \tag{11}$$

where $\mathcal{L}_{\text{cont}}$ is applied to the 259 continuous channels, $\mathcal{L}_{\text{contact}}$ is binary cross-entropy with logits on the four foot-contact channels, and $\mathcal{L}_{\text{root}}$ adds extra supervision to the first four root-motion channels.

The decoded motion is additionally supervised in physical SMPL22 joint coordinates:

$$\begin{aligned}\mathcal{L}_{\text{visible}} = 0.5\mathcal{L}_{\text{joint}} + \mathcal{L}_{\text{distal}} + \mathcal{L}_{\text{root-traj}} \\ + \mathcal{L}_{\text{root-drift}} + \mathcal{L}_{\text{distal-vel}}.\end{aligned} \tag{12}$$

Here, $\mathcal{L}_{\text{joint}}$ compares all 22 joints without root alignment; $\mathcal{L}_{\text{distal}}$ emphasizes the two feet and two wrists (SMPL22 indices $\{10, 11, 20, 21\}$); $\mathcal{L}_{\text{root-traj}}$ supervises the pelvis trajectory in the horizontal plane; $\mathcal{L}_{\text{root-drift}}$ penalizes start-to-end horizontal displacement error; and $\mathcal{L}_{\text{distal-vel}}$ matches frame-to-frame displacement of the distal joints.

We further match first- and second-order temporal differences of all 259 continuous IK263 channels using $\mathcal{L}_{\text{vel}}$ and $\mathcal{L}_{\text{acc}}$. Their weights are zero during the first 80% of training and then linearly increase to 0.25 and 0.02, respectively, over the final 20%. The complete objective is

$$\begin{aligned}\mathcal{L}_{\text{MLC}} = \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{visible}} + \mathcal{L}_{\text{VQ}} \\ + \lambda_v \mathcal{L}_{\text{vel}} + \lambda_a \mathcal{L}_{\text{acc}}.\end{aligned} \tag{13}$$

For reproducibility, Table 5 summarizes the final loss weights. These values were empirically set during MLC development to obtain stable training and satisfactory reconstruction quality, and the same configuration is kept fixed for all subsequent experiments.

The MLC is trained with a global batch size of 96 for 92,400 optimizer updates and is frozen before CMF training.

A.3 Visual Condition Preprocessing and Sampling

For each OpenPose BODY25 detection, confidence is thresholded at $\delta = 0.4$, slightly below the conventional 0.5 threshold to retain more potentially informative joint observations. This yields a binary visibility indicator,

$$m_{t,j} = \mathbf{1}[c_{t,j} \geq \delta]. \tag{14}$$

The condition encoder receives normalized coordinates and this binary visibility mask; the original confidence values are not directly provided as condition features. During training, each visible joint is additionally dropped independently with probability 0.05.

The 2D coordinate origin is BODY25 MidHip (joint 8) from the first frame of the full resampled source sequence, before selecting the 196-frame crop or the observed condition frames. If MidHip is invisible in that frame, we use the mean of the visible joints; if no joint is visible, the origin is set to zero. This fixed origin is retained for all crops so that image-plane displacement is preserved. For each frame, body scale is computed from the visible-joint bounding box,

$$s_t = \max(x_{\max,t} - x_{\min,t},\ y_{\max,t} - y_{\min,t}), \tag{15}$$

with a minimum value of 10^{-6}. Only visible joints contribute to the bounding box. After root-and-scale normalization, each joint coordinate is standardized using training-set per-joint statistics. Invisible coordinates are set to zero after standardization, and normalized coordinates are clipped to $[-10, 10]$. The flattened 50 coordinate values and 25 visibility indicators form the 75-D pose feature used by TAPE.

For $T = 196$, we use $K \in \{196, 98, 49, 28, 14, 7\}$. The three sampling patterns used in training and robustness evaluation are implemented as follows.

- **Even:** indices are $\mathrm{round}(\mathrm{linspace}(0, T-1, K))$, which includes both endpoints for $K \geq 2$.
- **Random:** K distinct frame indices are sampled uniformly without replacement from $\{0, \ldots, T-1\}$ and then sorted.
- **Boundary-gap:** frames 0 and $T-1$ are retained for $K \geq 2$. A contiguous interior gap is sampled with a preferred length 25%–50% of the sequence, capped by the available unobserved-frame budget. The remaining $K-2$ observations are sampled uniformly without replacement from interior frames outside the gap.

The main sparse evaluation uses $K = 7$ with even sampling.

A.4 Initial-Flow Training and Reprojection Objective

The primary Flow Matching loss is evaluated directly on sampled interpolation states and does not require numerical integration. The auxiliary reprojection objective is evaluated periodically by rolling a subset of Gaussian sources to $t = 1$, decoding the predicted endpoint with the frozen MLC decoder, recovering SMPL22 joints, and comparing the projected joints with the observed 2D poses. The generated endpoint is decoded as a continuous latent and is not requantized through the MLC codebook.

At each conditioning timestamp, the corresponding generated frame is selected using

$$i_k = \mathrm{round}\big(\tau_k(T-1)\big). \tag{16}$$

We use the following 14 BODY25-compatible joints:

BODY25 joint	BODY25 index	SMPL22 joint
Neck	1	Avg of shoulders (16, 17)
Right shoulder	2	Right shoulder (17)
Right elbow	3	Right elbow (19)
Right wrist	4	Right wrist (21)
Left shoulder	5	Left shoulder (16)
Left elbow	6	Left elbow (18)
Left wrist	7	Left wrist (20)
MidHip	8	Avg of hips (1, 2)
Right hip	9	Right hip (2)
Right knee	10	Right knee (5)
Right ankle	11	Right ankle (8)
Left hip	12	Left hip (1)
Left knee	13	Left knee (4)
Left ankle	14	Left ankle (7)

A single weak-perspective camera with scale s and image-plane translation $\mathbf{b} = (b_x, b_y)$ is fitted jointly over all observed condition frames. The fitting target is the root-centered OpenPose coordinate system with the image y axis flipped upward; the per-frame bounding-box normalization used by TAPE is not used for this loss. Camera fitting initially uses compatible joints with confidence at least 0.4 and falls back to all positive-confidence joints if fewer than six valid joint observations are available. No rotation or perspective parameters are fitted, and the shared scale is not constrained to be positive. For valid frame i, the reprojection error is

$$e_i = \frac{\sum_j c_{i,j} \left\| s\widehat{\mathbf{J}}^{xy}_{i,j} + \mathbf{b} - \mathbf{p}_{i,j} \right\|_2}{\sum_j c_{i,j} + 10^{-6}}, \tag{17}$$

where $\mathbf{b} = (b_x, b_y)$. The loss is the unweighted mean of e_i over valid conditioning frames. Thus, the reprojection discrepancy is a confidence-weighted Euclidean distance.

The reprojection term is activated after 20,760 initial-flow updates and ramps to $\lambda_{\text{reproj}} = 0.001$ by update 27,680. Because differentiable rollout is expensive, it is evaluated once every eight optimizer updates on a random sub-batch of 32 samples per GPU. The rollout uses Euler integration: 16 steps before update 27,680 and a random choice of 8 or 16 steps afterward. The conditional velocity field is used directly during this training regularizer, without inference-time classifier-free guidance.

Selection of the reprojection-loss weight. We further examine the effect of the reprojection-loss weight λ_{reproj}. As shown in Fig. 5, adding reprojection supervision substantially improves both PMFD and FID compared with training without this regularizer. A smaller weight of 5×10^{-4} gives the best PMFD, while $\lambda_{\text{reproj}} = 10^{-3}$ achieves a slightly better FID with nearly the same PMFD. We therefore choose $\lambda_{\text{reproj}} = 10^{-3}$ as a balanced setting for the main experiments, since it provides strong motion fidelity while also slightly improving overall generation quality. A larger weight of 2×10^{-3} begins to degrade PMFD, suggesting that overly strong reprojection supervision can over-constrain the model.

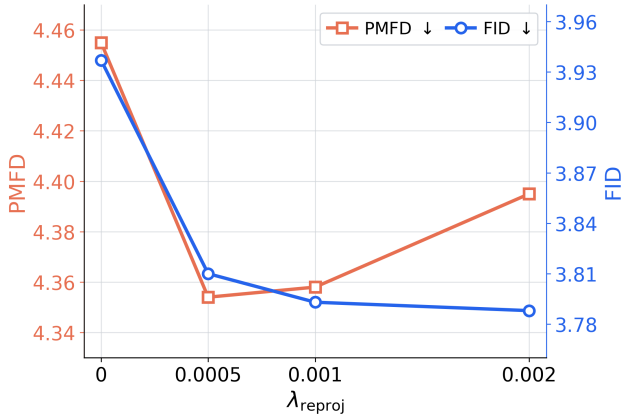

Figure 5: Effect of λ_{reproj} on PMFD and FID. Reprojection supervision improves both metrics, and $\lambda_{\text{reproj}} = 10^{-3}$ is selected as a balanced setting for the main experiments. The ablation branches are evaluated at update 41,520 under $K = 7$ boundary-gap conditioning, without reflow training. Lower is better.

A.5 Reflow Training

The final rectification stage uses online teacher–student reflow. For each training sample, a fresh Gaussian source $\mathbf{z}_0$ is paired with the endpoint $\mathbf{z}_1^T$ produced by the frozen Round-0 EMA teacher at update 68,220 under the same visual condition, where the superscript T denotes the teacher. Teacher endpoints are generated with an 8-step Heun solver. To make the reflow model compatible with classifier-free guidance (CFG), the teacher guidance scale is sampled as $s = 5.0$ with probability 0.70, $s = 2.5$ with probability 0.15, and $s = 1.0$ with probability 0.15. The guided teacher velocity field is

$$\mathbf{v}_s^T = \mathbf{v}_\varnothing^T + s\big(\mathbf{v}_c^T - \mathbf{v}_\varnothing^T\big). \tag{18}$$

Here, $\mathbf{v}_c^T$ and $\mathbf{v}_\varnothing^T$ denote the conditional and learned-null (unconditional) teacher velocity fields, respectively, and s is the CFG scale. The teacher endpoint $\mathbf{z}_1^T$ defines the reflow target displacement $\mathbf{z}_1^T - \mathbf{z}_0$. For $t \sim \mathcal{U}(0,1)$, the student is evaluated at $\mathbf{z}_t = (1-t)\mathbf{z}_0 + t\mathbf{z}_1^T$. Denoting its conditional and null velocity fields by $\mathbf{v}_c^S$ and $\mathbf{v}_\varnothing^S$, where S denotes the student, we optimize

$$\mathcal{L}_{\text{reflow}} = \left\| \mathbf{v}_\varnothing^S + s\big(\mathbf{v}_c^S - \mathbf{v}_\varnothing^S\big) - (\mathbf{z}_1^T - \mathbf{z}_0) \right\|_2^2 . \tag{19}$$

Teacher pairs are generated online. Unlike initial-flow CFG training, whole-condition dropout is not used: the conditional and learned-null branches are evaluated explicitly for each guided training sample. The reprojection objective is disabled during this stage.

Reflow uses the same six condition densities with sampling probabilities $\{0.30, 0.15, 0.20, 0.15, 0.12, 0.08\}$ for $K = \{196, 98, 49, 28, 14, 7\}$, respectively. Even, random, and boundary-gap patterns are sampled with probabilities 0.50, 0.30, and 0.20. Additional mini-batch quotas guarantee exposure to boundary-gap/$K = 7$ (8%), random/$K = 7$ (5%), and boundary-gap/$K = 14$ (4%) cases. Flow-time sampling uses 60% uniform samples and 10% each at the exact start, near the start, near the end, and the exact end. Detached one- or two-step student rollout states are introduced after update 10,380 and reach 20% of training states by update 17,300. The optimizer is AdamW with weight decay 0.01 and EMA decay 0.99. The learning rate warms up to 2.5×10^{-5} and later decays to 1.25×10^{-5}; the condition encoder uses 0.1 times the main learning rate. The global batch size is 896 initially and 832 after enabling the later rollout stage. The deployed checkpoint is selected at update 33,334 using one-step validation performance.

A.6 Training Hyperparameters

The main optimization and training settings for MLC, the initial CMF flow, and reflow are summarized in Table 6.

Table 6: Main optimization settings used for MLC and CMF training.

Setting	MLC	Initial CMF	Reflow
Optimizer	AdamW	AdamW	AdamW
Learning rate	2×10^{-4}	2×10^{-4}	2.5×10^{-5} peak
Weight decay	0.01	0.01	0.01
Global batch size	96	896	$896 \rightarrow 832$
Training updates	92,400	∼69,200	33,334 selected
EMA decay	0.99 (codebook)	0.99	0.99
Training GPUs	2×RTX A6000	2×RTX A6000	2×RTX A6000

B Evaluation Details

B.1 Baseline Condition Construction

AIST++ does not provide natural-language descriptions. For each dance motion, Qwen3-VL-32B-Instruct generates three semantically diverse descriptions. All text-conditioned baselines use this same three-caption pool. Each baseline selects a caption through its own seeded evaluation path; we do not enforce a cross-method caption-index manifest. Fig. 6 shows one example of the textual descriptions generated for a single video.

For StickMotion, the matched 196-frame evaluation uses three sketches at

$$\lfloor 0.125T \rfloor , \qquad \lfloor 0.5T \rfloor , \qquad \lfloor 0.875T \rfloor , \tag{20}$$

which correspond to frames $[24, 98, 171]$ for $T = 196$. These sketches are constructed from the paired ground-truth SMPL22 motion. Each selected 3D pose is pelvis-centered, normalized by body scale, and projected with StickMotion's canonical frontal projection before its native hand-drawn deformation. Therefore, the baseline receives clean pose guidance without detector noise. StickMotion uses the same AIST++ caption pool described above.

For OmniControl, spatial conditions are constructed following Xie et al. (2024) from the global SMPL-22 joint positions of the paired ground-truth motion. At each controlled timestamp, one joint



Description 1: Facing forward, step right then left with alternating weight shifts, arms moving side to side in rhythm, progressing gently forward at a steady, moderate pace.

Description 2: Positioned forward, the dancer moves laterally with alternating steps, weight shifting fluidly, arms swinging naturally alongside, advancing slightly forward at a moderate, even tempo.

Description 3: With feet alternating in a side-to-side motion while facing forward, weight transfers smoothly, arms swinging naturally at the sides, moving subtly left in a consistent, moderate rhythm.

Figure 6: Example of textual conditioning generated for an AIST++ dance sequence. Qwen3-VL-32B-Instruct produces three semantically diverse descriptions from the same video. While the captions capture coarse motion semantics such as stepping direction, weight shifts, arm movement, and tempo, they do not fully specify fine-grained motion details such as exact joint configurations, transient poses, or their precise timing. This illustrates the limited ability of text alone to provide detailed spatiotemporal motion control.

is selected from among the pelvis, left/right foot, head, and left/right wrist, while all other joint–time entries are masked. Evaluation follows the specified keyframe count and sampling pattern. OmniControl therefore receives exact 3D spatial guidance without pelvis centering, camera projection, or detector noise. Its text condition uses frozen CLIP ViT-B/32 embeddings of the same three-caption pool described above.

For CondMDI, we train the model following Cohan et al. (2024) with full-body 3D keyframe conditioning. At evaluation, sparse conditioning uses full-body 3D keyframes taken from the paired ground-truth motion at the same $K = 7$ evenly spaced timestamps used for sparse evaluation. Its text condition is drawn from the same three-caption pool described above.

Data splits are performed at the underlying-motion level, so different views of the same motion do not cross splits.

B.2 Motion Feature Evaluator

FID, Diversity, and PMFD are computed with a T2M-style motion feature extractor trained on the AIST++ training split and frozen for evaluation. We follow the original T2M training configuration, replacing the original text annotations with the three Qwen3-VL-32B-Instruct descriptions generated for each AIST++ motion. The same caption pool is used by the text-conditioned baselines. FID compares the generated and real motion-feature distributions, Diversity follows the original T2M pairwise feature-distance protocol, and PMFD is the mean L_2 distance between each generated motion feature and its paired target motion feature.

B.3 Reprojection Distance

RPD uses a single sequence-level weak-perspective camera fitted only on the observed condition frames $\mathcal{C}$. For the 14 compatible joints $\mathcal{J}$, camera parameters are obtained from the weighted least-squares problem

$$(s^\star, b_x^\star, b_y^\star) = \arg\min_{s,b_x,b_y} \sum_{t\in\mathcal{C}} \sum_{j\in\mathcal{J}} c_{t,j} \Big[(s x_{t,j} + b_x - u_{t,j})^2 + (s y_{t,j} + b_y - v_{t,j})^2 \Big], \tag{21}$$

where $c_{t,j}$ is the OpenPose joint confidence. The fitted camera is then frozen and reused for unobserved and all frames. Unlike the training reprojection loss, the RPD camera fit accepts all finite compatible joints with positive confidence; it does not apply the 0.4 threshold. The semantic joint set is the same 14-joint set used during training, although the evaluator constructs MidHip as the average of the two SMPL hip joints and Neck as the average of the two shoulder joints. No rotational alignment is performed.

B.4 Physical Plausibility Metrics

Acceleration P90. Let $\mathbf{J}_{t,j}$ denote global 3D joint j at frame t and let $f = 30$ fps. We compute

$$\mathbf{v}_{t,j} = (\mathbf{J}_{t+1,j} - \mathbf{J}_{t,j})f, \qquad \mathbf{a}_{t,j} = (\mathbf{v}_{t+1,j} - \mathbf{v}_{t,j})f. \tag{22}$$

For each sequence, acceleration magnitudes from all frames and all 22 joints are pooled and the 90th percentile is computed. The reported score is the mean of these per-sequence values in $\mathrm{m/s^2}$. No root alignment is applied.

Foot skating. For FlowMimic, we apply a sigmoid to its predicted ankle- and toe-contact logits and mark frame t as a contact frame if either probability exceeds 0.5. For baselines, foot-contact states are obtained using their native decoding procedures. We compute the global 3D ankle speed between consecutive frames and average it over the corresponding contact frames. The foot-skating score is the mean of the left- and right-foot contact-conditioned speeds. Since the SMPL joint coordinates are expressed in meters, the score is multiplied by 1000 and reported in $\mathrm{mm/s}$.

B.5 Runtime Measurement

Inference timing is measured on a single NVIDIA RTX A6000. Data loading is excluded, while condition encoding, numerical integration including all internal CFG branches, latent denormalization, and MLC decoding are included. Solver-level NFE counts each guided vector-field evaluation requested by the numerical solver once; the conditional and null backbone calls inside CFG affect inference time but do not separately increase the reported NFE.

C Additional Experiments and Analysis

C.1 CondMDI With Ground-Truth And Estimated 3D Conditions

The main comparison in Table 1 provides CondMDI with full-body 3D keyframes from the paired target motion. To examine a setting where target 3D keyframes are unavailable, we construct the same $K = 7$ conditions from the demonstration video. Specifically, we estimate full-body 3D pose and trajectory from the corresponding frontal view using MediaPipe (Bazarevsky et al., 2020) and the calibrated AIST++ camera, and convert the resulting motion to the SMPL22 coordinate convention. Table 7 shows that replacing ground-truth 3D conditions with estimated 3D substantially degrades conditional fidelity and joint-space accuracy, demonstrating CondMDI's sensitivity to the accuracy of its 3D conditioning signal. This estimated setting provides a complementary comparison closer to FlowMimic's video-input scenario, but still requires an intermediate 3D recovery stage.

Table 7: Effect of 3D condition accuracy on CondMDI. GT 3D uses keyframes from the paired target motion, while estimated 3D uses full-body pose and global trajectory estimated from the demonstration video. Both settings use the same $K = 7$ evenly spaced timestamps. Lower is better.

Condition	PMFD ↓	RPD_a ↓	MPJPE ↓
GT 3D	$3.36^{\pm 0.03}$	$56.05^{\pm 0.20}$	$397.8^{\pm 3.3}$
Estimated 3D	$5.10^{\pm 0.07}$	$67.34^{\pm 0.46}$	$531.9^{\pm 6.5}$

C.2 MLC Latent Capacity

We ablate the temporal capacity of MLC using 8, 16, and 32 latent tokens, while keeping the code dimension (256) and codebook size (1,024) fixed. Table 8 first compares reconstruction quality at the same training epoch.

Reconstruction FID and PMFD measure distributional quality and paired motion fidelity, respectively. Joint L2 is computed over all 22 SMPL joints without root alignment, whereas distal-joint L2 considers only the feet and wrists. The acceleration and high-frequency energy ratios measure preservation of motion dynamics, with an ideal value of 1. Codebook utilization reports the fraction of VQ entries activated on the test set.

Table 8: Reconstruction performance with different MLC latent capacities. Ratio-based dynamics metrics have an ideal value of 1. All variants use 256-D codes and a 1,024-entry codebook. **Bold** indicates the best result.

Metric	8 tokens	16 tokens	32 tokens
Reconstruction FID ↓	4.059	**3.787**	4.692
Reconstruction PMFD ↓	3.903	**3.421**	3.971
Joint L2 ↓	0.319	0.269	**0.255**
Distal-joint L2 ↓	0.395	0.344	**0.339**
Acceleration P90 ratio → 1	0.577	**0.672**	0.577
High-frequency energy ratio → 1	0.147	**0.253**	0.159
Codebook utilization	17.5%	56.6%	89.3%

Eight tokens exhibit a clear capacity bottleneck. Increasing the capacity to 16 substantially improves both feature-space reconstruction and preservation of high-frequency motion. Increasing it further to 32 slightly reduces coordinate-level L2 errors and increases codebook utilization, but these gains do not translate into better FID or dynamics preservation, indicating that reconstruction quality does not improve monotonically with latent length. The codebook utilization rates of 56.6% and 89.3% for the 16- and 32-token models indicate that the chosen codebook capacity is actively used.

We further compare 16- and 32-token MLCs using matched downstream flow training. Both flows are evaluated at epoch 40, by which point they already produce reliable motion under dense conditioning ($K = 196$), providing sufficiently developed and training-matched checkpoints for comparing latent capacities. Evaluation uses 50-step Heun sampling and three matched seeds.

Table 9: Downstream flow performance with different MLC latent capacities under matched training and evaluation settings. **Bold** indicates the best result.

MLC	FID ↓	PMFD ↓	Diversity
16 tokens	$\mathbf{3.788}^{\pm 0.134}$	$3.929^{\pm 0.116}$	$8.428^{\pm 0.191}$
32 tokens	$4.163^{\pm 0.417}$	$\mathbf{3.867}^{\pm 0.121}$	$8.315^{\pm 0.151}$

As shown in Table 9, the two capacities give similar downstream paired fidelity, with 32 tokens providing only a small PMFD improvement. In contrast, 16 tokens achieve lower FID and slightly higher diversity while requiring only half as many latent tokens for flow modeling. Together with the reconstruction results, this suggests that 16 tokens already provide sufficient MLC capacity, whereas doubling the latent length yields no consistent quality improvement and makes the downstream flow problem more expensive. We therefore adopt 16 tokens as the best overall quality–capacity trade-off.

C.3 Solver and Reflow Analysis

C.3.1 Initial-Flow Solver Saturation

Before reflow, we evaluate the initial flow (update 68,220) with 8, 16, and 50 Heun steps using CFG scale 5.0 and evenly spaced conditions. Table 10 shows that performance is already saturated by 8 steps, particularly under sparse conditioning. We therefore use the 8-step initial flow as the reflow teacher, reducing teacher sampling cost without sacrificing conditional fidelity.

C.3.2 Effect of Reflow on Solver Efficiency

We evaluate the initial flow and the reflowed model under the same even-conditioning protocol at both sparse ($K = 7$) and dense ($K = 196$) densities. All results use the official test split, CFG scale 5.0, Heun integration, and ten replications.

As shown in Table 11, the initial flow is highly sensitive to solver resolution: one- and two-step sampling fail substantially, while four to eight Heun steps are required to reach the stable quality region. After reflow, a single Heun step already achieves comparable performance at both condition densities. In particular, the one-step reflowed model obtains the best FID, while its PMFD and RPD

Table 10: Solver-step ablation for the initial flow on the matched AIST++ test protocol. NFE counts guided vector-field evaluations. Results are reported as means with 95% confidence intervals.

Condition	Steps	NFE	FID ↓	PMFD ↓	RPD_{all} ↓
$K = 196$	8	16	$3.20^{\pm 0.05}$	$3.57^{\pm 0.02}$	$41.15^{\pm 0.13}$
$K = 196$	16	32	$3.03^{\pm 0.05}$	$3.56^{\pm 0.03}$	$41.42^{\pm 0.15}$
$K = 196$	50	100	$3.03^{\pm 0.05}$	$3.56^{\pm 0.03}$	$41.47^{\pm 0.15}$
$K = 7$	8	16	$3.46^{\pm 0.10}$	$3.94^{\pm 0.02}$	$52.60^{\pm 0.30}$
$K = 7$	16	32	$3.46^{\pm 0.11}$	$3.96^{\pm 0.03}$	$52.59^{\pm 0.28}$
$K = 7$	50	100	$3.48^{\pm 0.11}$	$3.96^{\pm 0.03}$	$52.70^{\pm 0.25}$

Table 11: Solver-step comparison before and after reflow under matched even conditioning. Reflow compresses the useful multi-step trajectory of the initial flow into a single Heun step. **Bold** indicates the best result.

K	Reflow	Steps	FID ↓	PMFD ↓	RPD_{all} ↓
7	×	1	$109.807^{\pm 1.821}$	$12.372^{\pm 0.090}$	$111.294^{\pm 0.278}$
	×	2	$6.688^{\pm 0.326}$	$5.438^{\pm 0.044}$	$62.526^{\pm 0.221}$
	×	4	$3.713^{\pm 0.102}$	$4.067^{\pm 0.020}$	$53.453^{\pm 0.177}$
	×	8	$3.464^{\pm 0.105}$	$\mathbf{3.945}^{\pm 0.018}$	$\mathbf{52.600}^{\pm 0.295}$
	✓	1	$\mathbf{3.231}^{\pm 0.101}$	$4.097^{\pm 0.036}$	$53.124^{\pm 0.298}$
196	×	1	$118.597^{\pm 0.497}$	$12.043^{\pm 0.039}$	$118.150^{\pm 0.096}$
	×	2	$36.170^{\pm 1.686}$	$8.225^{\pm 0.095}$	$76.499^{\pm 0.262}$
	×	4	$3.728^{\pm 0.059}$	$\mathbf{3.564}^{\pm 0.019}$	$42.300^{\pm 0.115}$
	×	8	$3.202^{\pm 0.055}$	$3.567^{\pm 0.025}$	$\mathbf{41.152}^{\pm 0.125}$
	✓	1	$\mathbf{3.115}^{\pm 0.073}$	$3.619^{\pm 0.018}$	$41.849^{\pm 0.186}$

remain close to the best multi-step non-reflow results. These results indicate that reflow primarily compresses the useful solver trajectory into a single step rather than uniformly improving every high-NFE metric.

C.4 Robustness to Visual Conditions

C.4.1 Pose Extraction and Observation Corruption

We further examine whether FlowMimic remains responsive to visual conditions under three types of perturbation: detector shift, localization noise, and missing observations. For detector shift, OpenPose is replaced at inference time by Google MediaPipe Pose (Bazarevsky et al., 2020) without retraining; for localization noise, visible joints are perturbed with Gaussian noise; and for observation loss, a contiguous 50% interval is removed. We evaluate robustness through paired motion fidelity, observed-frame joint accuracy and reprojection-based condition adherence.

For the MediaPipe setting, its 2D poses are converted to the BODY25 layout using direct joint correspondences and simple geometric construction for unmatched joints, and then processed identically to the OpenPose inputs. For the coordinate-noise setting, we add zero-mean Gaussian noise at twice the empirical OpenPose–MediaPipe localization discrepancy measured on normalized 2D joints. Results are summarized in Table 12.

Under detector shift, all fidelity metrics change only modestly when MediaPipe is used in place of OpenPose, despite the model being trained with OpenPose conditions. FlowMimic is also robust to detector-scale localization noise: even at twice the empirical OpenPose–MediaPipe noise level, all metrics remains nearly unchanged under both sparse ($K = 7$) and dense ($K = 196$) conditioning.

A more challenging case is a contiguous observation outage. When all pose observations are removed from half of the sequence, fidelity degrades primarily within the missing interval, while pose accuracy at the remaining observed timestamps is largely preserved. For example, at $K = 7$,

Table 12: Robustness to pose-extraction variations and corrupted observations. MediaPipe (Bazarevsky et al., 2020) replaces OpenPose at inference time without retraining. The $2\times$ noise setting adds Gaussian perturbations at twice the empirical OpenPose–MediaPipe localization discrepancy, while the 50% outage removes all pose observations within a contiguous half of the sequence. RPD is always evaluated against the original uncorrupted observations. MPJPE_o is MPJPE on the remaining observed condition frames and is reported in mm.

K	Condition	PMFD ↓	MPJPE_o ↓	$\mathrm{RPD}_{\mathrm{unobs.}}$ ↓	$\mathrm{RPD}_{\mathrm{all}}$ ↓
7	OpenPose	$4.10^{\pm 0.04}$	$364.94^{\pm 13.66}$	$53.23^{\pm 0.30}$	$53.12^{\pm 0.30}$
7	MediaPipe	$4.21^{\pm 0.03}$	$356.21^{\pm 2.66}$	$54.09^{\pm 0.17}$	$53.98^{\pm 0.17}$
7	$2\times$ coordinate noise	$4.12^{\pm 0.03}$	$369.02^{\pm 11.14}$	$53.84^{\pm 0.22}$	$53.74^{\pm 0.22}$
7	50% temporal outage	$4.63^{\pm 0.04}$	$355.65^{\pm 14.24}$	$63.28^{\pm 0.53}$	$63.00^{\pm 0.52}$
196	OpenPose	$3.62^{\pm 0.02}$	$322.30^{\pm 3.55}$	–	$41.85^{\pm 0.19}$
196	MediaPipe	$3.82^{\pm 0.02}$	$325.10^{\pm 2.95}$	–	$43.80^{\pm 0.13}$
196	$2\times$ coordinate noise	$3.61^{\pm 0.02}$	$322.57^{\pm 2.21}$	–	$41.97^{\pm 0.18}$
196	50% temporal outage	$3.93^{\pm 0.03}$	$318.18^{\pm 5.11}$	$62.32^{\pm 0.54}$	$53.16^{\pm 0.30}$

$\mathrm{MPJPE_o}$ remains comparable to the uncorrupted setting, whereas $\mathrm{RPD_{unobserved}}$ increases from 53.2 to 63.3. This behavior suggests that missing visual evidence mainly increases uncertainty over the unconstrained portion of the trajectory rather than disrupting the model's ability to follow the available conditions. Thus, these experiments demonstrate robustness to practical detector variation, localization noise, and substantial temporal observation loss.

Table 13: Sensitivity to input camera view. All four views are included during training and evaluated independently under the same protocol. RPD denotes the standard reprojection error of generated motion against the 2D input conditions. Real RPD replaces the generated motion with the corresponding target 3D motion and therefore reflects the view-dependent reference error induced by pose extraction and the sequence-shared projection model. **Bold** indicates the best result.

K	View	FID ↓	PMFD ↓	$\mathrm{RPD}_{\mathrm{all}}$ ↓	Real RPD
7	Frontal	$\mathbf{3.231}^{\pm 0.101}$	$\mathbf{4.097}^{\pm 0.036}$	$\mathbf{53.124}^{\pm 0.298}$	21.741
	Low-angle frontal	$3.420^{\pm 0.109}$	$4.137^{\pm 0.036}$	$61.957^{\pm 0.416}$	25.944
	Left-frontal 45°	$3.844^{\pm 0.148}$	$4.235^{\pm 0.041}$	$60.956^{\pm 0.267}$	40.281
	Right-frontal 45°	$3.539^{\pm 0.115}$	$4.157^{\pm 0.037}$	$62.184^{\pm 0.145}$	40.656
196	Frontal	$\mathbf{3.115}^{\pm 0.073}$	$3.619^{\pm 0.018}$	$\mathbf{41.849}^{\pm 0.186}$	21.520
	Low-angle frontal	$3.156^{\pm 0.109}$	$3.614^{\pm 0.017}$	$49.434^{\pm 0.208}$	25.688
	Left-frontal 45°	$3.667^{\pm 0.116}$	$\mathbf{3.584}^{\pm 0.025}$	$51.262^{\pm 0.188}$	39.900
	Right-frontal 45°	$3.647^{\pm 0.066}$	$3.614^{\pm 0.022}$	$53.872^{\pm 0.087}$	40.214

C.4.2 Sensitivity to Camera View

Although canonical evaluation uses the frontal camera, FlowMimic is trained with frontal, low-angle frontal, left-frontal, and right-frontal views. Table 13 evaluates these viewpoints independently. PMFD remains stable across views, while FID shows a moderate degradation for oblique views.

RPD also increases under non-frontal views. However, part of this increase comes from the reprojection evaluation itself. To quantify this effect, we add Real RPD, which replaces the generated motion with the target 3D motion and compares its reprojection against the 2D pose extracted from the input demonstration video. Since the target motion corresponds to the demonstrated sequence, this value serves as a view-dependent reference error caused by pose-extraction differences and the sequence-shared projection approximation. For the oblique cameras, this floor rises from approximately 22–26 to about 40, indicating that the increase in generated RPD cannot be attributed entirely to generation error. Even without correcting for this view-dependent floor, non-frontal RPD remains close to CondMDI's 56.05 in the main comparison, falling below it under dense conditioning and only moderately exceeding it under sparse conditioning. It also remains substantially below all other baselines, indicating that FlowMimic retains condition adherence across the tested viewpoints.