Back to Raw Motion: Latent-Free Diffusion Transformers for Text-to-Motion and Exact Conditional Generation

Yun Chen *
KAIST
cyruby@kaist.ac.kr

Jeonghyeok Do *
KAIST
ehwjdgur0913@kaist.ac.kr

Munchurl Kim †
KAIST
mkimee@kaist.ac.kr

https://kaist-viclab.github.io/RawMo_site

Abstract

*As generative models tackle larger and more complex datasets, learned codecs enable efficient generation through compact latent representations, but may lose details and cannot guarantee exact preservation of given conditions. Generating directly in the original data space avoids errors introduced by the codec. Recent advances in image generation demonstrate that pixel diffusion can produce high-quality samples at practical computational costs. Inspired by these advances, we conduct a preliminary study of pixel diffusion, demonstrating its potential for adaptation to motion generation. Building on these findings, we present **RawMo**, a framework that brings text-to-motion generation to raw motion space. Our Motion Diffusion Transformer (MiT) models local skeletal structures and whole-body dynamics, while refining motion across frames to improve temporal consistency. Working directly in raw motion space further allows our high-noise text-motion alignment loss (HiNCE) to improve text consistency without intermediate motion decoding. RawMo supports multiple motion-conditioning tasks with a single conditional model while preserving given motion conditions exactly. Experiments on HumanML3D and KIT-ML show the best text alignment among compared methods and competitive motion quality, with ablations confirming the benefits of MiT and HiNCE.*

1. Introduction

Text-to-motion generation produces 3D human motion sequences from natural-language descriptions. The task has broad applications in character animation, games, and virtual environments, where natural and diverse movements are essential. Paired text-motion datasets [8, 27] have supported progress in learning the relationship between language and movement. Early diffusion models generate motion through iterative denoising of full motion sequences [32], but repeated processing can incur substantial sampling costs [1]. To reduce the computational burden, many approaches move generation into compact learned representations, using either continuous latent spaces or discrete motion codes [1, 9, 44]. Generation in these compact spaces has advanced efficiency while achieving improvements in motion quality and text consistency.

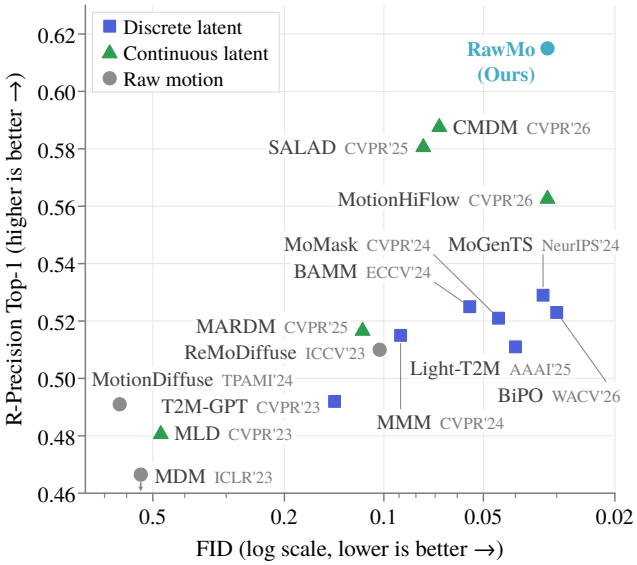

Figure 1. **Text-to-motion generation on HumanML3D.** RawMo achieves the highest R@1 with competitive FID under the 263-D evaluator of Guo *et al.* [8]. Marker shapes distinguish discrete latent, continuous latent, and raw-motion methods. The downward arrow indicates MDM's off-scale R@1 of 0.320.

As generative models tackle larger and more complex data, learned latent spaces offer compact representations that reduce computation. An encoder compresses data into latent codes, and a decoder converts generated latent codes back into data samples. However, details can be lost during

*These authors contributed equally.
†Corresponding author.

encoding and decoding, and latent-space generation alone is unable to guarantee exact preservation of given conditions. Recent work on pixel diffusion in the image domain demonstrates that direct data-space generation can achieve high quality at practical computational costs [18, 40]. We explore this direction in the motion domain, where body regions and frame order can provide the basis for organizing local motion features and modeling temporal changes. Our goal is to use these known body and time relationships to model motion directly, instead of learning a separate compressed representation. Fig. 2(a) illustrates our motivation for returning to raw motion: (i) avoiding errors introduced by a learned motion codec through direct generation and (ii) preserving given motion conditions exactly rather than matching them only approximately.

JiT [18] highlights that a denoising network can predict in one space while its loss is computed in another through output conversion. Inspired by this idea, we conduct a preliminary study comparing prediction targets and loss spaces using a plain JiT-style Transformer over raw motion patches. Consistent with JiT's image experiments, clean-motion prediction with a flow-velocity loss gives the lowest FID among the tested combinations, supporting the feasibility of extending pixel-diffusion principles to motion generation. Section 3 presents the complete study.

Building on these findings, we bring text-to-motion generation *back to raw motion* with **RawMo**. Adapting pixel diffusion to motion requires modeling body-part relationships and coordinated movement. Our Motion Diffusion Transformer (MiT) extends PixelDiT's dual-level design [40] to model skeletal structures and whole-body dynamics, with a refinement head that promotes temporal continuity. To strengthen text alignment, we pretrain motion and text encoders on paired training data to establish a shared embedding space for measuring text-motion consistency. During RawMo training, our high-noise text-motion alignment loss (HiNCE) uses these frozen pretrained encoders to supervise clean-motion predictions at high-noise timesteps, guiding the model toward the described action in the early denoising steps. Fig. 2(b) illustrates how MiT and HiNCE address motion structure and text alignment.

RawMo supports multiple motion-conditioning tasks with a single model, preserving given motion conditions *exactly* while generating the remaining motion. Comparisons on HumanML3D [8] and KIT-ML [27] assess motion quality, text consistency, and motion-conditioning performance. Fig. 1 shows HumanML3D results. RawMo achieves the strongest text alignment among compared methods and zero control error in motion-conditioned generation; ablations reveal how motion structure and supervision timing shape the balance between motion quality and text consistency.

Our main contributions are summarized as follows:

- We conduct a study demonstrating the feasibility of adapting pixel-diffusion principles to motion generation; the study identifies clean-motion prediction with a flow-velocity objective as an effective combination.
- We propose RawMo, which brings text-to-motion generation back to raw motion; MiT models body and time structure, while HiNCE provides semantic supervision at high-noise timesteps.
- Experiments on HumanML3D and KIT-ML demonstrate RawMo's superior generation performance, while a single RawMo model supports multiple motion-conditioning tasks with exact preservation of given motion conditions.

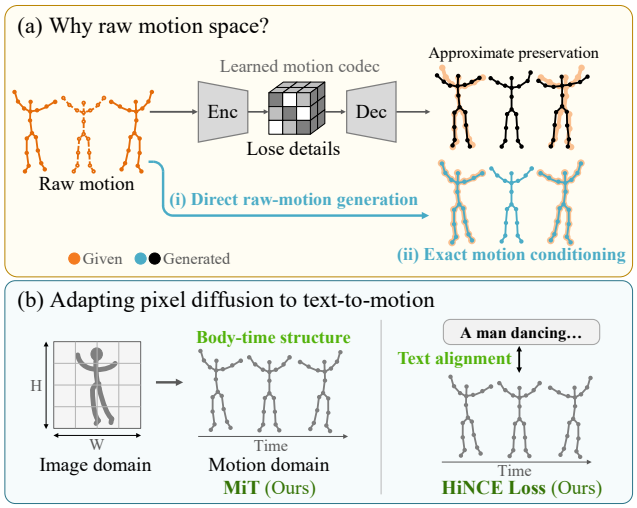

Figure 2. **Motivation for RawMo.** (a) Direct raw-motion generation avoids errors introduced by a learned motion codec and enables exact preservation of given motion conditions. (b) Adapting pixel diffusion to text-to-motion requires modeling body-time structure and aligning generated motion with textual descriptions.

2. Related Work

2.1. Pixel-space image diffusion

Pixel-space diffusion generates images directly without a learned codec. Early diffusion models performed iterative denoising in pixel space, but processing high-resolution images incurred substantial computational costs [5, 11]. Latent diffusion reduced these costs by moving denoising into compressed representations, introducing dependence on a learned encoder and decoder [29]. Recent work revisits pixel-space generation through suitable prediction targets, loss spaces, and efficient architectures [2, 18, 40]. JiT [18] separates prediction targets from loss spaces, showing that clean-image prediction with a velocity-space objective enables plain Transformers to model high-dimensional pixel patches. Architectural developments further separate global modeling from local detail synthesis [2, 40]. PixelDiT [40] combines patch-level semantic modeling with Pixel Transformer (PiT) blocks for fine-grained pixel up-

dates, while complementary designs use lightweight local refinement [2]. These developments revisit the prediction task and computational structure of direct generation, motivating raw-motion generation that accounts for body structure and temporal continuity.

2.2. Text-to-motion generation

Text-to-motion models generate human motion sequences that follow textual descriptions. Early diffusion-based methods denoise motion sequences directly in motion space [32, 46, 47], with clean-motion prediction enabling geometric supervision on the generated sequence [32]. To reduce generation costs, many approaches adopt compact learned motion representations. Continuous methods perform diffusion or flow matching in an autoencoder's latent space [1, 17, 39], while discrete methods combine vector quantization with autoregressive or masked token prediction [9, 24, 41, 44]. Skeleton-aware latent designs further capture relationships among joints, frames, and words [12]. Recent work also explores flow matching directly in motion space, with representation-aware noise scaling [38]. Our RawMo revisits direct motion generation through body-region modeling, frame-level refinement, and high-noise text-motion supervision informed by recent pixel diffusion.

2.3. Motion-conditional generation

Motion-conditional generation synthesizes motion under partial observations, such as joint trajectories, keyframes, or selected body parts. Existing approaches use inference-time inpainting or spatial guidance [15, 32], compose diffusion priors for joint-level control [30], or use learned control branches [4, 37]. Other methods refine structured motion latents through trajectory-based optimization [33], while observation-conditioned denoising learns completion from randomly sampled frame and joint observations [3]. Unified frameworks further combine generation and editing through a common formulation of source motion, conditions, and target motion [10]. Prior work incorporates partial motion observations during denoising [32] and uses sampling-time projection to enforce exact linear spatial constraints [35]. Our RawMo preserves given motion conditions exactly using a single conditional model with a global skip.

3. Preliminary: Back to Raw Motion

To assess the feasibility of transferring pixel-space diffusion principles to motion, we conduct a preliminary study in raw motion space. JiT [18] motivates our study by showing that a plain Transformer can generate high-quality images directly from raw patches without a learned codec.

For a motion sequence $\mathbf{x} \in \mathbb{R}^{T\times D}$ with T frames and D features per frame, we form patches of $s = 4$ consecutive frames, yielding $\mathbf{x}_{\text{patch}} \in \mathbb{R}^{(T/s)\times(sD)}$. Our plain baseline applies the same grouping to the noisy sequence, linearly embeds each patch into a Transformer token, and uses a linear output head to reconstruct $s \times D$ motion features per token. The study includes both the plain baseline and RawMo (Sec. 4) to examine whether JiT's findings for images hold across motion architectures.

Table 1. **Prediction targets and loss spaces on HumanML3D.** We report FID↓ averaged over three seeds. Rows specify prediction targets and columns specify loss spaces. Both architectures achieve their lowest FID with clean-motion prediction and a velocity-space loss.

Prediction	Plain JiT-style			RawMo (Ours)		
	$\mathbf{x}$-loss	$\boldsymbol{\epsilon}$-loss	$\mathbf{v}$-loss	$\mathbf{x}$-loss	$\boldsymbol{\epsilon}$-loss	$\mathbf{v}$-loss
$\mathbf{x}$	.0450	.0819	.0350	.0328	.0909	.0317
$\boldsymbol{\epsilon}$	61.68	62.95	61.77	.0477	.2244	.0406
$\mathbf{v}$	9.08	8.69	8.41	.0492	.0812	.0355

A denoising network does not need to use the same space for prediction and loss computation. Following JiT [18], we denote the network by f_θ and distinguish the *prediction target* from the *loss space*. The prediction target specifies the direct network output, while the loss space specifies where the converted output is compared with the corresponding ground truth to compute the training loss.

We construct the noisy motion sequence $\mathbf{x}_\tau$ by linearly interpolating between clean motion $\mathbf{x}$ and Gaussian noise $\boldsymbol{\epsilon}$ as follows:

$$\mathbf{x}_\tau = \tau\mathbf{x} + (1-\tau)\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}), \tag{1}$$

where $\tau = 0$ corresponds to pure Gaussian noise and $\tau = 1$ to clean motion. The flow velocity along this path is given by $\mathbf{v} = \mathbf{x} - \boldsymbol{\epsilon}$. The three prediction targets are linked by the interpolation path: given $\mathbf{x}_\tau$ and $0 < \tau < 1$, any one of $\mathbf{x}$, $\boldsymbol{\epsilon}$, and $\mathbf{v}$ determines the other two. For example, a network conditioned on caption $\mathbf{y}$ can output a clean-motion prediction $\hat{\mathbf{x}} = f_\theta(\mathbf{x}_\tau, \tau; \mathbf{y})$. The corresponding noise prediction $\hat{\boldsymbol{\epsilon}}$ and velocity prediction $\hat{\mathbf{v}}$ are obtained as:

$$\hat{\boldsymbol{\epsilon}} = \frac{\mathbf{x}_\tau - \tau\hat{\mathbf{x}}}{1-\tau}, \quad \hat{\mathbf{v}} = \frac{\hat{\mathbf{x}} - \mathbf{x}_\tau}{1-\tau}. \tag{2}$$

The converted predictions allow a network that directly predicts clean motion to receive supervision in clean-motion, noise, or velocity space. The prediction target and loss space can therefore be chosen separately.

Algebraic conversion does not imply identical training behavior. Although the three prediction targets are convertible, their training formulations differ in network parameterization and timestep-dependent loss weighting. Table 1 evaluates all nine combinations using squared ℓ_2 losses. Rows indicate the direct network output, and columns indicate the loss space.

The results reproduce JiT's findings for images: both the plain baseline and RawMo achieve their lowest FID

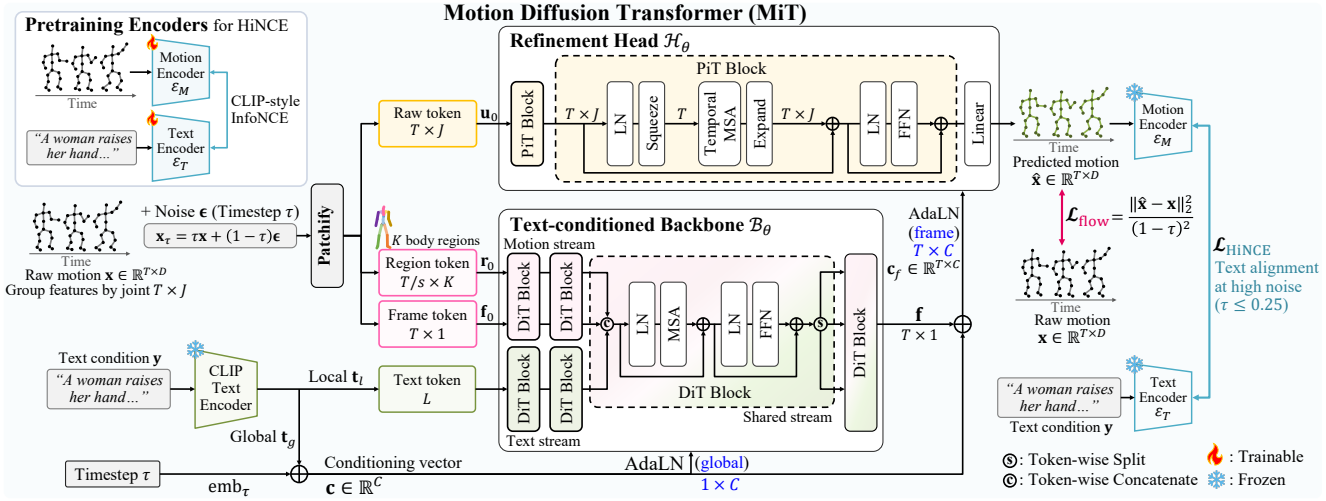

Figure 3. **RawMo framework.** The Motion Diffusion Transformer (MiT) predicts clean motion through a text-conditioned backbone and a refinement head. The backbone integrates body-region and frame representations with text, while the head refines raw tokens through frame-wise conditioning and temporal attention. Training combines a velocity-space objective with HiNCE, which provides text-alignment supervision at high noise. The motion and text encoders $\mathcal{E}_M$ and $\mathcal{E}_T$ are pretrained for HiNCE and remain frozen during MiT training.

with clean-motion prediction and a velocity-space loss. We therefore adopt this combination for RawMo, with the objective defined as:

$$\mathcal{L}_{\text{flow}} = \mathbb{E}\big[\|\hat{\mathbf{v}} - \mathbf{v}\|_2^2\big] = \mathbb{E}\left[\frac{\|\hat{\mathbf{x}} - \mathbf{x}\|_2^2}{(1-\tau)^2}\right]. \tag{3}$$

The agreement across the two architectures supports the feasibility of bringing text-to-motion generation *back to raw motion* through these pixel-diffusion principles and provides an empirical basis for RawMo.

4. RawMo

Building on the findings in Sec. 3, RawMo generates text-conditioned motion directly in raw motion space using clean-motion prediction and a velocity-space objective. Fig. 3 provides an overview of RawMo. The Motion Diffusion Transformer (MiT) combines body-region and frame representations in a text-conditioned backbone, followed by joint-level reconstruction through a refinement head (Sec. 4.1). HiNCE provides text-alignment supervision for clean-motion predictions at high-noise timesteps (Sec. 4.2). A global skip preserves given motion conditions exactly in each clean-motion prediction (Sec. 4.3).

4.1. Motion Diffusion Transformer (MiT)

MiT implements the denoising network f_θ through a text-conditioned backbone $\mathcal{B}_\theta$ and a refinement head $\mathcal{H}_\theta$, where θ denotes the trainable parameters. For a motion sequence $\mathbf{x} \in \mathbb{R}^{T \times D}$ with T frames and D motion features per frame ($D = 263$ on HumanML3D; 251 on KIT-ML) [8], we construct the noisy input $\mathbf{x}_\tau$ at timestep τ using Eq. 1. The network receives a caption $\mathbf{y}$ as the text condition.

MiT constructs three token types from $\mathbf{x}_\tau$: body-region tokens, frame tokens, and raw tokens. Body-region and frame tokens enter the text-conditioned backbone $\mathcal{B}_\theta$, while raw tokens take a separate path to the refinement head $\mathcal{H}_\theta$.

Text-conditioned backbone. Each body-region token represents one of $K = 12$ body regions over $s = 4$ consecutive frames, providing local skeletal context within a short temporal window. Each frame token represents all D features of a single frame, retaining a sequence of whole-body states for modeling temporal changes. The tokenization operators $\mathcal{P}_{\text{region}}$ and $\mathcal{P}_{\text{frame}}$ group the corresponding motion features and linearly project them into initial region tokens $\mathbf{r}_0$ and frame tokens $\mathbf{f}_0$, respectively, as:

$$\begin{aligned} \mathbf{r}_0 &= \mathcal{P}_{\text{region}}(\mathbf{x}_\tau) \in \mathbb{R}^{(T/s)K \times C}, \\ \mathbf{f}_0 &= \mathcal{P}_{\text{frame}}(\mathbf{x}_\tau) \in \mathbb{R}^{T \times C}, \end{aligned} \tag{4}$$

where C denotes the backbone token dimension. Region and frame tokens form the input motion stream $[\mathbf{r}_0; \mathbf{f}_0]$, where $[\cdot;\cdot]$ denotes token concatenation.

A frozen CLIP text encoder [28] encodes the caption $\mathbf{y}$ into a global sentence-level embedding $\mathbf{t}_g \in \mathbb{R}^{C_{\text{text}}}$ and local token-level embeddings $\mathbf{t}_l \in \mathbb{R}^{L \times C_{\text{text}}}$, where C_{text} is the dimension and L is the token count. The global embedding summarizes the entire caption, while the local embeddings provide detailed text context for joint attention with motion. We combine the sentence-level embedding and timestep into the global conditioning vector $\mathbf{c}$ as:

$$\mathbf{c} = \psi_\tau(\text{emb}_\tau) + \psi_g(\mathbf{t}_g) \in \mathbb{R}^C, \tag{5}$$

where emb_τ is the sinusoidal embedding of timestep τ, and ψ_τ and ψ_g are separate multilayer perceptrons that map

their inputs to C-dimensional features. The global conditioning vector modulates each backbone block through adaptive normalization, providing timestep and text context.

Following FLUX [16], the backbone processes motion and text through dual-stream blocks followed by single-stream blocks. Dual-stream blocks maintain separate parameters for the motion stream $[\mathbf{r}_0; \mathbf{f}_0]$ and the local text stream $\mathbf{t}_l$, while attention over both streams allows information exchange between motion and text. Single-stream blocks then process the concatenated motion and text streams with shared parameters. After the final block, we retain only the updated frame features $\mathbf{f}$, defining the backbone output as:

$$\mathbf{f} = \mathcal{B}_\theta([\mathbf{r}_0; \mathbf{f}_0; \psi_l(\mathbf{t}_l)]; \mathbf{c}) \in \mathbb{R}^{T \times C}, \tag{6}$$

where ψ_l projects local text embeddings to dimension C. The output frame features $\mathbf{f}$ incorporate body-region and local text information, providing context for the refinement head $\mathcal{H}_\theta$. The backbone uses rotary positional embeddings (RoPE) [31] to encode the temporal position of each motion token. Region tokens use the center of their s-frame interval, while frame tokens use their frame indices.

Refinement head. Reconstructing each temporal patch as a separate group of frames can introduce periodic artifacts at patch boundaries. Our refinement head $\mathcal{H}_\theta$ addresses this issue by coordinating joint-level reconstruction across frames through temporal attention. We adapt the Pixel Transformer (PiT) blocks of PixelDiT [40] to motion, grouping joint features by frame. Along the raw-token path, a joint-wise projection $\mathcal{P}_{\text{joint}}$ embeds the noisy motion features of each joint at each frame into raw tokens $\mathbf{u}_0$ as:

$$\mathbf{u}_0 = \mathcal{P}_{\text{joint}}(\mathbf{x}_\tau) \in \mathbb{R}^{T \times J \times C_h}, \tag{7}$$

where J denotes the number of joints and C_h denotes the head token dimension. The backbone frame features $\mathbf{f}$ are combined with the global conditioning vector $\mathbf{c}$ to obtain frame-wise conditioning $\mathbf{c}_f$ as:

$$\mathbf{c}_f = \mathbf{c} \oplus \mathbf{f} \in \mathbb{R}^{T \times C}, \tag{8}$$

where $\oplus$ denotes element-wise addition after repeating $\mathbf{c}$ across the T frames. Each frame's conditioning vector modulates its joint features through adaptive normalization, supplying $\mathcal{H}_\theta$ with timestep, text, and motion context.

Within each head block, a squeeze operation compacts the J joint features of each frame into one vector. Temporal self-attention is then applied across the resulting T frame vectors, which are subsequently expanded back into joint features. A final linear projection maps the refined features back to the original motion channels, producing the clean-motion prediction $\hat{\mathbf{x}}$ as:

$$\hat{\mathbf{x}} = f_\theta(\mathbf{x}_\tau, \tau; \mathbf{y}) = \mathcal{H}_\theta(\mathbf{u}_0; \mathbf{c}_f) \in \mathbb{R}^{T \times D}. \tag{9}$$

Grouping and conditioning operate per frame, while temporal attention coordinates updates across time.

4.2. Text Alignment at High Noise

We use our high-noise text-motion alignment loss (HiNCE) to improve the text consistency of clean-motion predictions. Before MiT training, we jointly pretrain a motion encoder $\mathcal{E}_M$ and a text encoder $\mathcal{E}_T$ on paired training data. Through a CLIP-style InfoNCE objective [22, 28], the two encoders learn a shared embedding space for measuring how well a motion expresses its caption. Fig. 3 (top left) illustrates this pretraining stage.

During MiT training, $\mathcal{E}_M$ and $\mathcal{E}_T$ remain frozen and serve as a fixed semantic reference for HiNCE. Working in raw motion space allows HiNCE to supervise clean-motion predictions directly, passing semantic feedback to MiT without an intermediate motion decoder.

Let $d(\hat{\mathbf{x}}, \mathbf{y})$ denote a text-motion distance score between a predicted motion $\hat{\mathbf{x}}$ and caption $\mathbf{y}$, defined as the negative similarity score computed from $\mathcal{E}_M(\hat{\mathbf{x}})$ and $\mathcal{E}_T(\mathbf{y})$. To compute HiNCE over a batch of N motion-caption pairs, we sample a timestep τ_i for the i-th pair. The corresponding clean-motion prediction $\hat{\mathbf{x}}_i$ is paired with caption $\mathbf{y}_i$. We apply alignment supervision at high-noise timesteps to encourage the described action during the early denoising stages. Since smaller τ corresponds to higher noise, we select the subset $\mathcal{I} = \{i : \tau_i \leq 0.25\}$. For a nonempty $\mathcal{I}$, HiNCE is defined as:

$$\mathcal{L}_{\text{HiNCE}} = -\frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \log \frac{\exp[-d(\hat{\mathbf{x}}_i, \mathbf{y}_i)]}{\sum_{j=1}^{N} \exp[-d(\hat{\mathbf{x}}_i, \mathbf{y}_j)]}. \tag{10}$$

Only high-noise motion predictions contribute to $\mathcal{L}_{\text{HiNCE}}$, while each predicted motion is compared against all N captions, indexed by j. We set $\mathcal{L}_{\text{HiNCE}} = 0$ when $\mathcal{I}$ is empty.

Combining HiNCE with the velocity-space objective $\mathcal{L}_{\text{flow}}$ from Sec. 3 gives the overall text-to-motion training objective as:

$$\mathcal{L} = \mathcal{L}_{\text{flow}} + \lambda \mathcal{L}_{\text{HiNCE}}, \tag{11}$$

where $\lambda = 0.03$ is the empirically chosen weight of the alignment loss. $\mathcal{E}_M$ and $\mathcal{E}_T$ are used only during training to compute $\mathcal{L}_{\text{HiNCE}}$.

4.3. Exact Motion Conditioning

By generating directly in raw motion space, RawMo can preserve given motion conditions exactly in each clean-motion prediction. Given the noisy motion $\mathbf{x}_\tau$, we train the conditional model with the observed motion $\mathbf{x}_c \in \mathbb{R}^{T \times D}$ and a binary mask $\mathbf{m} \in \{0, 1\}^{T \times D}$ as additional input channels. Unobserved entries in $\mathbf{x}_c$ are zero-filled, and $\mathbf{m}$ is 1 at observed entries and 0 elsewhere.

A global skip keeps the given motion unchanged and completes the remaining parts using the clean-motion prediction $\hat{\mathbf{x}}$ [32]. The conditioned clean-motion prediction $\tilde{\mathbf{x}}$ is defined as:

$$\tilde{\mathbf{x}} = \mathbf{m} \odot \mathbf{x}_c + (1 - \mathbf{m}) \odot \hat{\mathbf{x}}, \tag{12}$$

Table 2. **Text-to-motion generation on HumanML3D.** Left: the 263-D evaluator [8]; right: the 67-D evaluator [21]. RawMo achieves the highest R-Precision and lowest MMDist under both evaluators, with the lowest FID under the 67-D evaluator. Across tables, results are means; ± denotes standard deviation. **Bold**/underline indicate the best/second-best results; dashes denote unreported results.

Text-to-motion generation		Guo *et al*. evaluator (263-D)					MARDM evaluator (67-D)				
Method	**Venue**	R@1↑	R@2↑	R@3↑	FID↓	MMDist↓	R@1↑	R@2↑	R@3↑	FID↓	MMDist↓
Discrete latent space											
T2M-GPT [44]	CVPR'23	.492	.679	.775	.141	3.121	.470	.659	.758	.335	3.505
MoMask [9]	CVPR'24	.521	.713	.807	.045	2.958	.490	.687	.786	.116	3.353
MMM [25]	CVPR'24	.515	.708	.804	.089	2.926	.487	.683	.782	.132	3.359
BAMM [24]	ECCV'24	.525	.720	.814	.055	2.919	–	–	–	–	–
MoGenTS [41]	NeurIPS'24	.529	.719	.812	.033	2.867	–	–	–	–	–
Light-T2M [42]	AAAI'25	.511	.699	.795	.040	3.002	–	–	–	–	–
BiPO [13]	WACV'26	.523	.714	.809	**.030**	2.880	–	–	–	–	–
Continuous latent space											
MLD [1]	CVPR'23	.481	.673	.772	.473	3.196	.461	.651	.750	.431	3.445
SALAD [12]	CVPR'25	.581	.769	.857	.076	2.649	.552	.748	.839	.124	2.990
MARDM [21]	CVPR'25	.517	.708	.805	.116	2.968	.500	.695	.795	.114	3.270
ACMDM [20]	arXiv'25	–	–	–	–	–	.522	.713	.807	.058	3.205
MotionHiFlow [17]	CVPR'26	.563	.754	.843	.032	2.691	–	–	–	–	–
CMDM [39]	CVPR'26	.588	.778	.860	.068	2.620	.563	.759	.849	.078	2.920
Raw motion space											
MDM [32]	ICLR'23	.320	.498	.611	.544	5.566	.440	.636	.742	.518	3.640
ReMoDiffuse [46]	ICCV'23	.510	.698	.795	.103	2.974	.468	.653	.754	.883	3.414
MotionDiffuse [47]	TPAMI'24	.491	.681	.782	.630	3.113	.450	.641	.753	.778	3.490
RawMo (Ours)	–	**.615**±.003	**.799**±.003	**.877**±.001	.032±.006	**2.481**±.011	**.584**±.002	**.774**±.002	**.859**±.002	**.028**±.006	**2.835**±.008

Table 3. **Text-to-motion generation on KIT-ML.** Results use the 251-D evaluator [8]. RawMo achieves the best scores across all five metrics, covering both motion quality and text alignment.

Method	R@1↑	R@2↑	R@3↑	FID↓	MMDist↓
Discrete latent space					
T2M-GPT [44]	.416	.627	.745	.514	3.007
MoMask [9]	.433	.656	.781	.204	2.779
BAMM [24]	.438	.661	.788	.183	2.723
MoGenTS [41]	.445	.671	.797	.143	2.711
Light-T2M [42]	.444	.670	.794	.161	2.746
Continuous latent space					
MLD [1]	.390	.609	.734	.404	3.204
SALAD [12]	.477	.711	.828	.296	2.585
MoMADiff [48]	.458	.678	.797	.122	2.633
MotionHiFlow [17]	.482	.704	.825	.135	2.552
Raw motion space					
MDM [32]	.164	.291	.396	.497	9.191
ReMoDiffuse [46]	.427	.641	.765	.155	2.814
MotionDiffuse [47]	.417	.621	.739	1.954	2.958
RawMo (Ours)	**.488**±.011	**.716**±.010	**.831**±.012	**.118**±.013	**2.442**±.039

where $\odot$ denotes element-wise multiplication. The first term carries the given motion directly to the output, while the second supplies predictions only for the unobserved parts. The global skip therefore preserves the given motion conditions exactly in each clean-motion prediction.

5. Experiments

5.1. Datasets and Evaluation Metrics

Datasets. We evaluate RawMo on HumanML3D [8] and KIT Motion-Language (KIT-ML) [27]. HumanML3D contains 14,616 motion sequences and 44,970 descriptions (22 joints, 20 fps), while KIT-ML contains 3,911 sequences and 6,278 descriptions (21 joints, 12.5 fps). We follow the standard training, validation, and test splits [8].

Text-to-motion evaluation. We report R-Precision (R@1/2/3) for text-motion retrieval, Fréchet Inception Distance (FID) for comparing generated and real motion distributions, and Multimodal Distance (MMDist) for the average embedding distance between paired motions and captions. On HumanML3D, we use both the standard 263-D evaluator of Guo *et al*. [8] and the 67-D evaluator introduced by MARDM [21]. The latter retains root motion and local joint positions. We use both evaluators to verify our performance across different motion representations. KIT-ML uses only the 251-D evaluator of Guo *et al*.

Motion-conditioned evaluation. We follow OmniControl [37] for pelvis control, reporting FID, R@3, Diversity (Div.), and foot skating ratio (Skate). Diversity is better when closer to the real-data reference, while Skate measures foot sliding during ground contact. Trajectory error (Traj.) is the fraction of trajectories with any controlled-frame error above 50 cm, while location error (Loc.) is the fraction of controlled frames with errors above this threshold. Average error (Err.) is the mean 3-D positional error (m) at controlled frames. We also report FID, R@3, and Div. under the 67-D evaluator.

5.2. Implementation Details

We train our models from scratch on the training split for 30k steps on HumanML3D and 20k on KIT-ML, using a batch size of 128 on one NVIDIA GeForce RTX 4090 GPU. AdamW uses a learning rate of 2×10^{-4}, $(\beta_1, \beta_2) = (0.9, 0.95)$, and weight decay 0.01. The warmup–stable–

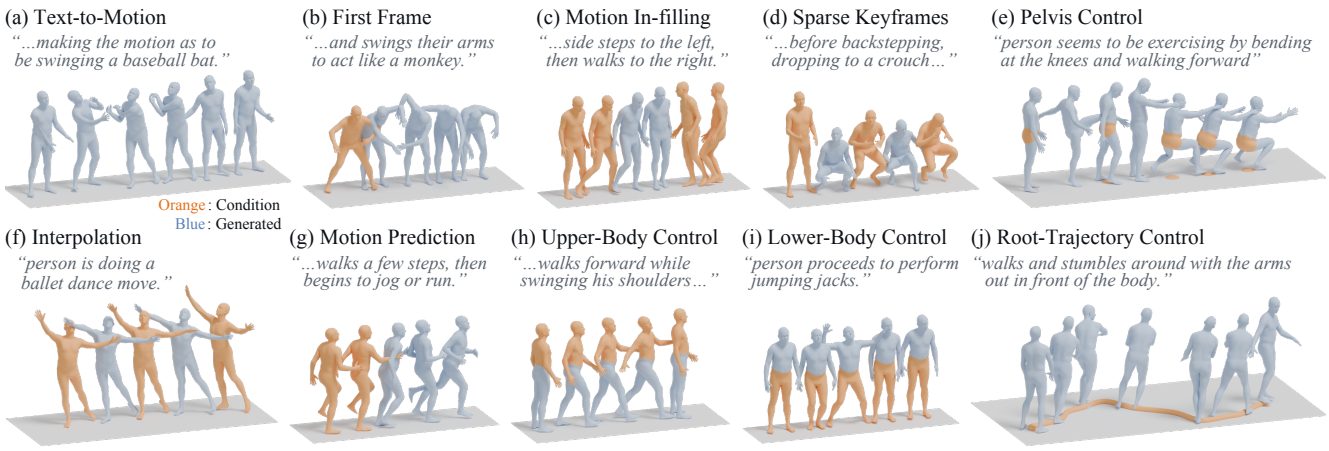

Figure 4. **Qualitative results of RawMo.** (a) Text-to-motion generation; (b–j) motion-conditioned generation using a single conditional model. Orange indicates given motion conditions, which are preserved exactly; blue indicates generated motion. The *Appendix* provides visual comparisons for text-to-motion and quantitative and qualitative results for diverse motion-conditioning tasks.

Table 4. **Motion-conditioned generation on HumanML3D.** Pelvis control follows OmniControl [37], averaging over 1, 2, or 5 observed frames and 25% or 100% frame coverage. Left: the 263-D evaluator; right: the 67-D evaluator following ProjFlow [35]. Real-data Diversity is 9.503 (263-D) and 10.455 (67-D). Evaluator-independent skating and control metrics are reported only on the left. A single conditional RawMo model achieves zero Traj., Loc., and Err., together with the best FID and R@3 under both evaluators.

Pelvis control (OmniControl)		**Guo *et al*. evaluator (263-D)**							**MARDM evaluator (67-D)**		
Method	**Venue**	FID↓	R@3↑	Div.	Skate↓	Traj.↓	Loc.↓	Err.↓	FID↓	R@3↑	Div.
MDM [32]	ICLR'23	.698	.602	9.197	.1019	.4022	.3076	.5959	1.792	.673	9.131
PriorMDM [30]	ICLR'24	.475	.583	9.156	.0897	.3457	.2132	.4417	.393	.707	9.847
GMD [15]	ICCV'23	.576	.665	9.206	.1009	.0931	.0321	.1439	.238	.763	10.011
OmniControl [37]	ICLR'24	.218	.687	9.422	**.0547**	.0387	.0096	.0338	.081	.789	10.323
InterControl [34]	NeurIPS'24	.159	.671	9.482	.0729	.0132	.0004	.0496	–	–	–
MotionLCM [4]	ECCV'24	.531	.752	9.253	–	.1887	.0769	.1897	–	–	–
TLControl [33]	ECCV'24	.271	.779	9.569	–	**0**	**0**	.0108	–	–	–
MaskControl [26]	ICCV'25	.061	.809	9.496	**.0547**	**0**	**0**	.0098	.066	.799	10.474
ACMDM [20]	arXiv'25	–	–	–	.0591	.0075	.0010	.0100	.067	.805	10.481
ProjFlow [35]	CVPR'26	–	–	–	.0629	**0**	**0**	**0**	.107	.784	10.644
RawMo (Ours)	–	**.040**	**.849**	9.685	.0674	**0**	**0**	**0**	**.055**	**.835**	10.767

decay schedule uses 2k linear warmup steps and linear decay to zero over the final 40% of training. We use bfloat16 and gradient-norm clipping at 1.0. A frozen CLIP ViT-B/32 encodes captions, with 0.1 caption dropout for classifier-free guidance. HiNCE uses $\lambda = 0.03$ for $\tau \leq 0.25$. We use a 12-block Transformer backbone (width 384, 6 heads) and a 2-block refinement head. The conditional model follows the HumanML3D settings without HiNCE, taking observations as extra input. Sampling uses Heun's [14] method with 16 intervals (31 network evaluations).

5.3. Text-to-Motion Generation

Baselines. We compare RawMo with methods that generate discrete motion codes [9, 13, 24, 25, 41, 42, 44], model continuous motion latents [1, 12, 17, 20, 21, 39, 48], or generate directly in raw motion space [32, 46, 47].

HumanML3D. Table 2 shows that RawMo achieves the highest R-Precision at all three retrieval levels and the lowest MMDist under both evaluators. With the 263-D evaluator of Guo *et al*. [8], RawMo improves R@1 from 0.588 to 0.615 and reduces MMDist from 2.620 to 2.481 compared with CMDM [39]. Although RawMo records a slightly higher FID than BiPO [13] under the standard evaluator (0.032 versus 0.030), RawMo reaches the highest R@1 of 0.584 and the lowest FID of 0.028 under the 67-D evaluator, compared with an FID of 0.058 for ACMDM [20]. The text-alignment advantage holds across both evaluators.

KIT-ML. Table 3 shows RawMo leads all five metrics. Compared with MotionHiFlow [17], RawMo improves R@1 from 0.482 to 0.488 and reduces MMDist from 2.552 to 2.442. RawMo achieves the lowest mean FID (0.118 vs. 0.122 for MoMADiff [48]). Results show competitive motion quality and improved text alignment on both datasets.

Table 5. **Ablation studies on HumanML3D.** We examine the refinement head, token types, HiNCE settings, and network size. R@1, FID, and MMDist use the 263-D evaluator; FID_{67} denotes FID under the 67-D evaluator.

Variant	Params	R@1↑	FID↓	MMDist↓	FID_{67}↓
Refinement head					
w/o $\mathcal{H}_\theta$	41.7M	.610±.003	.039±.004	2.513±.007	.060±.019
Token type					
w/o $\mathbf{r}_0$	45.4M	.612±.003	.034±.004	2.492±.011	.036±.006
w/o $\mathbf{f}_0$	42.2M	.612±.002	.044±.006	2.496±.012	.046±.007
Loss function					
w/o HiNCE	45.7M	.583±.001	.035±.004	2.605±.009	.029±.003
$\tau \leq 0.1$	45.7M	.601±.002	.039±.006	2.531±.004	.039±.005
$\tau \leq 1.0$	45.7M	.620±.001	.031±.003	2.467±.008	.049±.005
λ=0.015	45.7M	.607±.003	.036±.004	2.515±.007	.031±.004
λ=0.06	45.7M	.613±.002	.031±.005	2.482±.014	.029±.006
Network size					
Width 256	22.3M	.606±.002	.028±.001	2.514±.011	.029±.006
Width 512	78.0M	.615±.003	.039±.006	2.472±.010	.031±.002
Depth 16	53.4M	.613±.002	.035±.002	2.485±.013	.031±.005
Depth 24	68.8M	.614±.002	.036±.006	2.482±.053	.030±.002
RawMo (Full)	45.7M	.615±.003	.032±.006	2.481±.011	.028±.006

5.4. Motion-Conditioned Generation

Baselines. Baselines cover inference-time control using motion priors [15, 30, 32], learned control mechanisms [4, 37], and optimization or projection for constraint enforcement [33, 35], alongside recent motion-control frameworks [20, 26, 34].

Pelvis control. Pelvis control requires the generated full-body motion to follow pelvis positions specified at selected frames. We preserve height with the global skip and adjust root velocities during sampling to match horizontal targets (see *Appendix*). Table 4 reports results on HumanML3D following the protocol of OmniControl [37], using a single conditional RawMo model across all observation densities. RawMo achieves **0 Traj., 0 Loc., and 0 Err.**, demonstrating exact preservation of the specified pelvis positions. Compared with MaskControl [26] under the 263-D evaluator of Guo *et al*. [8], RawMo reduces FID from 0.061 to 0.040 and improves R@3 from 0.809 to 0.849. Under the 67-D evaluator [21], RawMo also achieves the lowest FID (0.055) and highest R@3 (0.835). These results demonstrate exact spatial control alongside improved motion quality and text alignment under both evaluators. Fig. 4 shows qualitative results of our RawMo on text-to-motion generation and diverse motion-conditioning tasks.

5.5. Ablations and Attention Analysis

Table 5 examines motion modeling, HiNCE settings, and network sizes on HumanML3D. Fig. 5 shows the relative contributions of frame, region, and text tokens to attention outputs in the trained backbone. The contribution measure is defined in the *Appendix*.

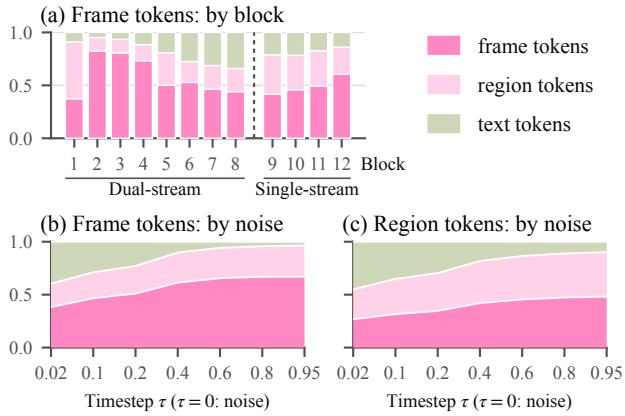

Figure 5. **Sources of attention outputs.** Contributions from frame, region, and text tokens, averaged over five seeds. (a) By backbone block, averaged over timesteps; (b, c) by timestep τ, averaged over blocks. Smaller τ means higher noise.

Motion modeling. Removing the refinement head worsens FID_{67} from 0.028 to 0.060 and MMDist from 2.481 to 2.513. These degradations support our refinement head's role in coordinating reconstruction across frames. Removing either body-region or frame tokens also worsens FID under both evaluators. Fig. 5(a) shows the sources of frame-token attention outputs by block. Early dual-stream blocks mainly use frame and region information, with text contributions increasing in later dual-stream blocks. In the single-stream stage, frame contributions rise again, and the final frame features provide context for the refinement head. These features already carry region and text information from earlier blocks, so lower direct text contributions do not necessarily indicate weaker text influence.

HiNCE. Without HiNCE, R@1 drops from 0.615 to 0.583 and MMDist worsens from 2.481 to 2.605. Extending supervision from our high-noise range ($\tau \leq 0.25$) to all timesteps ($\tau \leq 1.0$) improves R@1 to 0.620 and MMDist to 2.467, but worsens FID from 0.028 to 0.049 under the 67-D evaluator, showing a trade-off between text alignment and motion quality. Fig. 5(b,c) shows that text accounts for a larger share of the attention outputs of frame and region tokens at high noise, where HiNCE provides semantic supervision. As the noise level decreases, the relative contributions of frame and region tokens increase.

6. Conclusion

We presented **RawMo**, bringing text-to-motion generation *back to raw motion*. MiT captures skeletal structure and temporal dynamics, while HiNCE improves text alignment through clean-motion supervision at high noise. Experiments on HumanML3D and KIT-ML demonstrate the potential of direct raw-motion generation for high-quality, text-consistent motion with exact condition preservation.

Table 6. **Overview of the *Appendix*.**

Section	Contents
Sec. A	Text-to-motion generation on large-scale datasets
Sec. B	Baseline comparisons across nine conditioning tasks
Sec. C	Additional ablations on KIT-ML
Sec. D	Prediction targets and loss spaces
Sec. E	Computational cost on HumanML3D
Sec. F	Additional visual comparisons
Sec. G	Implementation details
Sec. H	Limitations of zero-shot control and sampling latency

This supplement extends the main paper with text-to-motion results on MotionMillion and RoMo, baseline comparisons across nine HumanML3D motion-conditioning tasks, and ablation studies on KIT-ML. Additional experiments compare all nine combinations of prediction target and loss space on HumanML3D and KIT-ML and report computational costs on HumanML3D. We also provide visual comparisons and implementation details, including attention contribution measurement, and discuss limitations and future work in zero-shot motion control and sampling speed. Table 6 summarizes the contents.

A. Text-to-Motion Generation on Large-Scale Datasets

We extend the HumanML3D and KIT-ML experiments to two large-scale motion datasets, MotionMillion [7] and RoMo [45], following their official splits. On both datasets, RawMo directly generates the provided 272-D motion features at 30 fps. Each baseline is retrained separately on each dataset using its official implementation and training schedule for each stage. Checkpoints and, where applicable, guidance scales are selected based on the lowest validation FID.

Table 7 reports results on both datasets. On MotionMillion, RawMo ranks first on all five metrics, reducing FID from 19.79 (MotionStreamer [36]) to 9.38 and MM-Dist from 23.58 (CMDM [39]) to 21.18. On RoMo, RawMo achieves the lowest FID of 8.57, followed by MotionStreamer at 12.92, and ranks second on R@1 (0.531) and MM-Dist (30.02).

B. Motion-Conditioned Generation on HumanML3D

We extend the main paper's pelvis-control comparison to the other eight motion-conditioning tasks in Fig. 4. OmniControl [37], MaskControl [26], and ACMDM [20] are fine-tuned from their released control checkpoints on the same mixture of nine tasks using their official training code. These baselines receive the specified joint positions in global coordinates.

Table 8 shows that RawMo achieves the highest R@3 and zero error at the conditioned joint positions on all nine tasks. RawMo also achieves the lowest FID on eight tasks and ranks second on sparse keyframes, with FID 0.033 compared with 0.024 for ACMDM. Averaged across tasks, RawMo obtains FID 0.033 and R@3 0.837; the second-best values are 0.044 for ACMDM and 0.813 for MaskControl, respectively.

C. Ablations on KIT-ML

Table 9 extends the main paper's HumanML3D ablations to KIT-ML [27]. Removing the refinement head $\mathcal{H}_\theta$ produces the largest degradation in mean FID, from 0.118 to 0.177. Without frame tokens $\mathbf{f}_0$, FID worsens to 0.150, compared with 0.124 without body-region tokens $\mathbf{r}_0$.

The full KIT-ML model applies HiNCE at $\tau \leq 0.25$ with weight $\lambda = 0.02$. Removing HiNCE lowers R@1 from 0.488 to 0.477. Applying HiNCE at all timesteps improves R@1 to 0.495 but worsens FID to 0.132. Increasing backbone depth to 16 gives the lowest mean FID (0.115), with 57.5M parameters compared with 49.9M for the full model. Overall, both datasets show similar trends when removing motion-modeling components or changing the HiNCE timestep range.

D. Prediction Targets and Loss Spaces

Table 1 in the main paper compares prediction targets and loss spaces for a plain JiT-style Transformer [18] and RawMo on HumanML3D over three seeds. Table 10 in this supplement extends this comparison to both HumanML3D and KIT-ML, with the architecture fixed to RawMo. The prediction target and loss space are independently chosen from clean motion $\mathbf{x}$, velocity $\mathbf{v}$, and noise $\boldsymbol{\epsilon}$, giving nine combinations. For each combination, we convert the network output into the selected loss space and compute squared error against the corresponding ground truth. RawMo adopts clean-motion prediction with a velocity-space loss, for which the per-sample error is $\|\hat{\mathbf{x}} - \mathbf{x}\|_2^2/(1-\tau)^2$.

Clean-motion prediction with a velocity-space loss gives the highest HumanML3D R@1 of 0.615 and the lowest FID_{67} of 0.028. Under the 263-D evaluator on HumanML3D, clean-motion prediction gives the same reported FID of 0.032 with either a velocity-space loss or a clean-motion loss. On KIT-ML, clean-motion prediction with a velocity-space loss gives the lowest FID of 0.118. Velocity prediction with a noise-space loss gives the highest KIT-ML R@1 of 0.496, but FID worsens to 0.196.

E. Computational Cost

Table 11 compares the computational cost of the methods evaluated on HumanML3D in the main paper. Generator

Table 7. **Text-to-motion generation on MotionMillion and RoMo.** We extend evaluation to two large-scale motion datasets. For each dataset, we evaluate all methods on a fixed subset of 8,192 clips selected from its own test set. Results are averaged over three random seeds. Both datasets are evaluated with the MotionMillion evaluator, making the RoMo evaluation cross-corpus. RawMo achieves the best results on all five metrics on MotionMillion and the lowest FID on RoMo.

Method	Venue	MotionMillion					RoMo				
		R@1↑	R@2↑	R@3↑	FID↓	MM-Dist↓	R@1↑	R@2↑	R@3↑	FID↓	MM-Dist↓
Discrete latent space											
T2M-GPT [44]	CVPR'23	.372	.497	.569	222.70	34.51	.223	.350	.440	140.57	34.82
MoMask [9]	CVPR'24	.737	.881	.929	51.98	26.72	.504	.675	.762	47.00	30.38
MMM [25]	CVPR'24	.591	.718	.778	112.41	29.98	.305	.443	.534	82.88	33.89
BAMM [24]	ECCV'24	.769	.905	.948	30.50	25.55	.508	.672	.759	34.33	30.31
MoGenTS [41]	NeurIPS'24	.760	.896	.941	50.23	25.90	.520	.683	.765	45.10	30.09
Light-T2M [42]	AAAI'25	.636	.793	.862	32.56	28.57	.462	.627	.719	30.90	30.76
MotionMillion [7]	ICCV'25	.468	.636	.721	40.98	32.33	.281	.418	.517	25.56	33.75
Continuous latent space											
MLD [1]	CVPR'23	.517	.692	.775	34.07	31.48	.306	.456	.551	26.37	33.39
SALAD [12]	CVPR'25	.614	.772	.843	58.68	29.26	.523	.693	.780	32.56	30.21
MARDM [21]	CVPR'25	.578	.759	.837	33.00	29.85	.414	.580	.677	14.98	31.44
MotionStreamer [36]	ICCV'25	.823	.933	.964	19.79	23.92	.492	.655	.745	12.92	30.54
ACMDM [20]	arXiv'25	.571	.746	.825	30.53	30.11	.363	.522	.618	13.47	32.44
MotionHiFlow [17]	CVPR'26	.773	.901	.944	28.03	25.23	.492	.656	.743	38.08	30.58
CMDM [39]	CVPR'26	.828	.939	.970	24.14	23.58	**.569**	**.735**	**.813**	18.42	**29.22**
Raw motion space											
MDM [32]	ICLR'23	.341	.502	.599	153.56	35.67	.236	.367	.463	293.53	35.71
ReMoDiffuse [46]	ICCV'23	.443	.609	.696	130.45	33.69	.402	.559	.656	26.09	32.16
MotionDiffuse [47]	TPAMI'24	.185	.294	.367	122.75	38.51	.238	.362	.449	46.46	34.90
RawMo	–	**.900**$_{\pm.001}$	**.970**$_{\pm.001}$	**.986**$_{\pm.001}$	**9.38**$_{\pm0.21}$	**21.18**$_{\pm0.02}$	.531$_{\pm.002}$	.691$_{\pm.002}$	.774$_{\pm.002}$	**8.57**$_{\pm0.41}$	30.02$_{\pm0.04}$

Table 8. **Motion-conditioned generation on HumanML3D.** The nine tasks correspond to Fig. 4(b)–(j) of the main paper. Results are averaged over five training seeds per method, with one test evaluation per seed. FID and R@3 use the 263-D evaluator of Guo *et al.* [8]. Err. measures the mean 3D positional error (cm) at the specified joints and frames.

Task	OmniControl [37]			MaskControl [26]			ACMDM [20]			**RawMo (Ours)**		
	FID↓	R@3↑	Err.↓	FID↓	R@3↑	Err.↓	FID↓	R@3↑	Err.↓	FID↓	R@3↑	Err.↓
(b) First Frame	.175	.784	9.4	.065	.805	1.0	.086	.808	4.8	**.056**	**.855**	**0**
(c) Motion In-filling	.075	.796	7.8	.047	.821	2.6	.039	.802	3.9	**.028**	**.831**	**0**
(d) Sparse Keyframes	.184	.787	7.1	.054	.820	2.3	**.024**	.814	4.8	.033	**.838**	**0**
(e) Pelvis Control	.144	.791	4.2	.077	.805	1.0	.055	.810	5.6	**.040**	**.849**	**0**
(f) Interpolation	.124	.804	8.1	.025	.820	2.9	.026	.806	4.4	**.020**	**.821**	**0**
(g) Motion Prediction	.106	.798	1.4	.068	.816	1.9	.040	.807	3.1	**.029**	**.845**	**0**
(h) Upper-Body Control	.083	.798	6.5	.030	.806	2.7	.030	.798	3.5	**.029**	**.822**	**0**
(i) Lower-Body Control	.103	.798	5.2	.043	.810	2.6	.050	.805	3.4	**.021**	**.819**	**0**
(j) Root-Trajectory Control	.129	.792	5.7	.052	.810	1.8	.046	.807	3.2	**.045**	**.849**	**0**
Average	.125	.794	6.2	.051	.813	2.1	.044	.806	4.1	**.033**	**.837**	**0**

parameter counts exclude the frozen text encoder. Generation time includes text encoding, sampling, and motion decoding for methods that use a decoder. NFE denotes the number of function evaluations during sampling, with a plus sign separating successive sampling stages. Passes counts individual generator forward passes.

MARDM [21] and ACMDM [20] use adaptive solvers, so their evaluation counts can vary. We measure MARDM runtime using weights we trained. ACMDM runtime is estimated assuming an NFE of 134, which is the median evaluation count of its smaller released model. BiPO [13] runtime is omitted because pretrained weights were unavailable for our measurements.

RawMo uses a 45.7M-parameter generator and 16 Heun sampling intervals [14], requiring 31 generator evaluations and 62 forward passes with classifier-free guidance. Gen-

Table 9. **Ablation studies on KIT-ML.** We examine the refinement head, token types, HiNCE settings, and network size using the 251-D evaluator of Guo *et al.* [8]. For each of five training seeds, we select the guidance scale on the validation set and average ten test evaluations. Results are reported as mean ± standard deviation across the five training seeds.

Variant	Params	R@1↑	R@3↑	FID↓	MMDist↓
Refinement head					
w/o $\mathcal{H}_\theta$	41.6M	$.472_{\pm.006}$	$.822_{\pm.003}$	$.177_{\pm.021}$	$2.542_{\pm.039}$
Token type					
w/o $\mathbf{r}_0$	49.5M	$.484_{\pm.006}$	$.828_{\pm.005}$	$.124_{\pm.005}$	$2.464_{\pm.021}$
w/o $\mathbf{f}_0$	43.3M	$.483_{\pm.002}$	$.826_{\pm.005}$	$.150_{\pm.025}$	$2.506_{\pm.026}$
Loss function					
w/o HiNCE	49.9M	$.477_{\pm.006}$	$.820_{\pm.008}$	$.133_{\pm.012}$	$2.481_{\pm.025}$
$\tau \leq 0.10$	49.9M	$.481_{\pm.008}$	$.823_{\pm.005}$	$.142_{\pm.008}$	$2.471_{\pm.017}$
$\tau \leq 1$	49.9M	$.495_{\pm.006}$	$.834_{\pm.008}$	$.132_{\pm.036}$	$2.446_{\pm.040}$
$\lambda = 0.01$	49.9M	$.478_{\pm.006}$	$.821_{\pm.002}$	$.126_{\pm.006}$	$2.493_{\pm.012}$
$\lambda = 0.04$	49.9M	$.485_{\pm.009}$	$.823_{\pm.003}$	$.121_{\pm.008}$	$2.476_{\pm.009}$
Network size					
Width 256	25.8M	$.486_{\pm.011}$	$.825_{\pm.003}$	$.131_{\pm.015}$	$2.479_{\pm.018}$
Width 512	82.8M	$.482_{\pm.003}$	$.824_{\pm.003}$	$.119_{\pm.010}$	$2.480_{\pm.023}$
Depth 16	57.5M	$.489_{\pm.006}$	$.832_{\pm.002}$	$.115_{\pm.004}$	$2.451_{\pm.017}$
Depth 24	72.9M	$.486_{\pm.002}$	$.824_{\pm.005}$	$.116_{\pm.008}$	$2.460_{\pm.011}$
RawMo (Full)	49.9M	$.488_{\pm.011}$	$.831_{\pm.012}$	$.118_{\pm.013}$	$2.442_{\pm.039}$

Table 10. **Prediction targets and loss spaces for RawMo.** Results are averaged over five training seeds, with the guidance scale selected on the validation set for each seed. On HumanML3D, R@1 and FID use the 263-D evaluator of Guo *et al.* [8], while FID_{67} uses the 67-D evaluator [21]. KIT-ML results use the 251-D evaluator of Guo *et al.*

Predict	Loss	HumanML3D			KIT-ML	
		R@1↑	FID↓	FID_{67}↓	R@1↑	FID↓
$\mathbf{x}$	$\mathbf{x}$	0.609	**0.032**	0.040	0.485	0.123
$\mathbf{x}$	$\mathbf{v}$	**0.615**	**0.032**	**0.028**	0.488	**0.118**
$\mathbf{x}$	$\boldsymbol{\epsilon}$	0.579	0.089	0.109	0.484	0.175
$\mathbf{v}$	$\mathbf{x}$	0.604	0.050	0.105	0.481	0.216
$\mathbf{v}$	$\mathbf{v}$	0.603	0.038	0.039	0.484	0.168
$\mathbf{v}$	$\boldsymbol{\epsilon}$	0.590	0.079	0.102	**0.496**	0.196
$\boldsymbol{\epsilon}$	$\mathbf{x}$	0.608	0.051	0.059	0.470	0.437
$\boldsymbol{\epsilon}$	$\mathbf{v}$	0.611	0.041	0.044	0.478	0.168
$\boldsymbol{\epsilon}$	$\boldsymbol{\epsilon}$	0.577	0.209	0.347	0.490	0.167

eration takes 0.594 s for one clip and 28.1 ms per clip at batch size 128. Compared with RawMo, ReMoDiffuse [46] is faster for single-clip generation at 0.269 s, but slower at batch size 128 with 42.5 ms per clip.

F. Additional Visual Comparisons

Text-to-motion generation. Fig. 6 extends Fig. 4(a) of the main paper with comparisons against MoMask, SALAD, and MotionHiFlow on HumanML3D and KIT-ML. In the first HumanML3D example, the character generated by RawMo holds both hands together in front of the torso, matching the text prompt. In the KIT-ML example of a jump to the left, RawMo generates an airborne pose.

Motion-conditioned generation. Figs. 7 and 8 extend the motion-conditioned visualizations in Fig. 4(b)–(j) of the main paper with baseline comparisons. These comparisons also complement the quantitative evaluation in Table 8 of this supplement. A single conditional RawMo model is compared with the fine-tuned OmniControl, MaskControl, and ACMDM baselines on two HumanML3D test motions, each under all nine motion-conditioning tasks. The baselines show nonzero positional errors at the conditioned joint positions, while RawMo preserves the given motion conditions exactly in every shown case.

Pelvis control. Figs. 9 and 10 provide visual comparisons corresponding to the quantitative pelvis-control results in Table 4 of the main paper. We compare sparse pelvis conditions at five randomly selected frames with a fully specified pelvis trajectory given at every frame. RawMo records zero error at the conditioned pelvis positions for both motions under both settings. In the ballet example, RawMo generates arm poses that differ from the reference while keeping the given pelvis positions unchanged.

G. Implementation Details

Attention contribution measurement. Fig. 5 in the main paper reports the relative magnitudes of attention updates contributed by frame, region, and text tokens. For each frame or region query, let $\mathcal{S}_g$ contain the key indices from source $g \in \{\text{frame}, \text{region}, \text{text}\}$. Within each attention head, we compute $\sum_{j \in \mathcal{S}_g} a_j \mathbf{z}_j$, where a_j is the attention weight from the query to key j and $\mathbf{z}_j$ is the corresponding value vector. After concatenating heads, we apply the attention gate, output projection without its bias, root mean square layer normalization (RMSNorm) [43] scaling, and residual gate. We denote the resulting attention update from source g in block ℓ by $\mathbf{u}_g^\ell$. All sources share the RMSNorm scaling computed from the full projected output, including the projection bias. The separate bias contribution is excluded from the source shares.

Let $Q \in \{\text{frame}, \text{region}\}$ denote the query type. We use E_Q to first average update norms over queries of type Q within each motion, then average the resulting motion-level means across motions. The contribution share of source g in block ℓ is

$$s_{Q,g}^\ell = \frac{\mathrm{E}_Q[\|\mathbf{u}_g^\ell\|_2]}{\sum_{g'} \mathrm{E}_Q[\|\mathbf{u}_{g'}^\ell\|_2]}.$$

For Fig. 5(a), we average the mean update norms over timesteps before computing the shares. For Fig. 5(b,c),

Table 11. **Computational cost on HumanML3D.** NFE denotes the number of function evaluations during sampling, counting each classifier-free guided prediction once. Passes counts individual generator forward passes. Both counts use 196-frame clips. Measured times are medians on a single NVIDIA RTX 4090 using each test clip's length, up to 196 frames. ACMDM [20] times are estimates.

Method	Autoencoder		Generator params	Text encoder	NFE	Passes	Time	
	Type	Params					1 clip (s)	Batch 128 (ms/clip)
Discrete latent space								
T2M-GPT [44]	VQ-VAE	19.4M	228.4M	CLIP-B	50	50	0.218	186.4
MoMask [9]	RVQ-VAE	19.4M	25.4M	CLIP-B	$10 + 5$	30	0.086	2.7
MMM [25]	VQ-VAE	18.0M	236.3M	CLIP-B	10	10	0.080	6.3
BAMM [24]	RVQ-VAE	19.4M	25.4M	CLIP-B	$98 + 5$	206	0.308	8.4
MoGenTS [41]	RVQ-VAE	113.8M	99.4M	CLIP-B	$20 + 10$	60	0.324	20.1
Light-T2M [42]	–	–	4.5M	CLIP-B	10	20	0.112	1.5
BiPO [13]	6 part-specific VQ-VAEs	6.3M	166.0M	CLIP-B	$51 + 25$	76	–	–
Continuous latent space								
MLD [1]	VAE	18.0M	8.3M	CLIP-L	50	100	0.145	3.8
SALAD [12]	VAE	0.17M	9.9M	CLIP-B	50	100	0.298	23.0
MARDM [21]	AE	18.8M	290.8M	CLIP-B	$18 +$ adaptive	$36 + 3.6$k	3.976	189.0
ACMDM [20]	AE	17.1M	596.2M	CLIP-B	adaptive	$2 \times$ NFE	2.218	1334.0
MotionHiFlow [17]	VAE	2.4M	30.8M	CLIP-B	14	28	0.200	8.1
CMDM [39]	VAE	76.3M	38.7M	DistilBERT	82	164	0.928	11.7
Raw motion space								
MDM [32]	–	–	17.9M	CLIP-B	1000	2000	4.473	399.6
ReMoDiffuse [46]	–	–	47.0M	CLIP-B	50	200	0.269	42.5
MotionDiffuse [47]	–	–	87.2M	CLIP-B	1000	1000	8.078	504.5
RawMo (Ours)	–	–	45.7M	CLIP-B	31	62	0.594	28.1

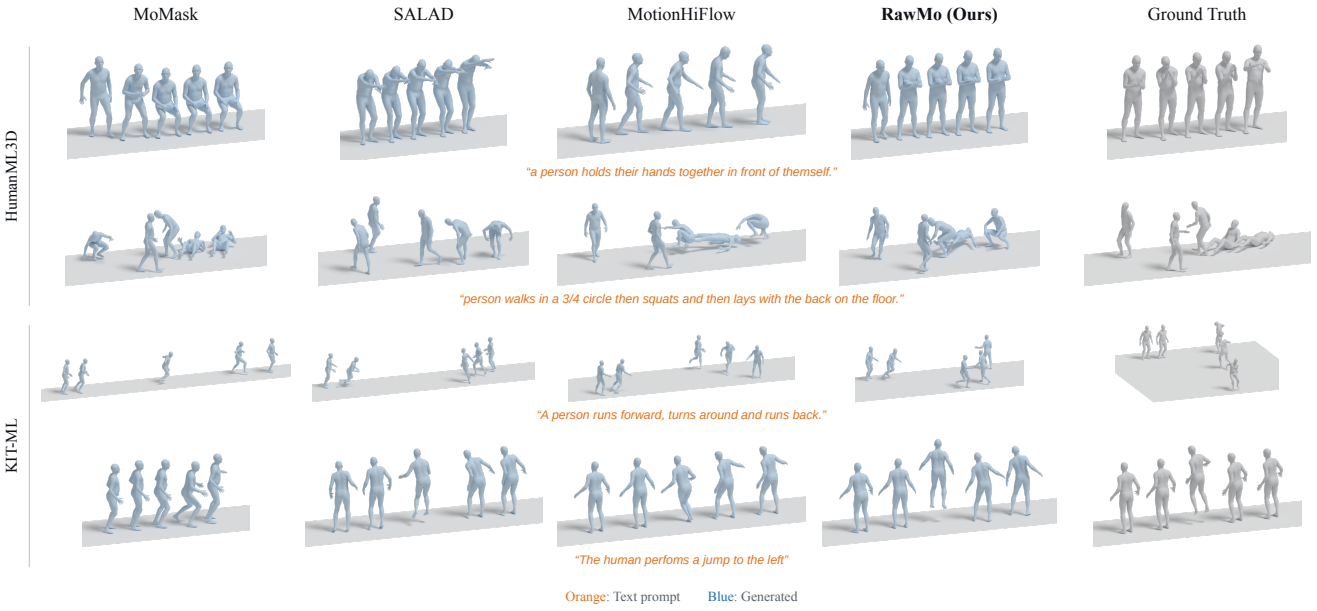

Figure 6. **Text-to-motion generation on HumanML3D and KIT-ML.** We extend the text-to-motion visualization in Fig. 4(a) of the main paper with comparisons against MoMask, SALAD, and MotionHiFlow. The top two rows show HumanML3D examples and the bottom two show KIT-ML examples. Methods in each row receive the text prompt shown in orange. Across these visual comparisons, generated motion is blue and the corresponding ground-truth reference is gray in the rightmost column.

we compute the shares separately in each block at each timestep, then average them over the 12 backbone blocks.

We average the results over five random seeds used to train RawMo for text-to-motion generation on HumanML3D. The analysis uses exponential moving average (EMA) weights and only the text-conditioned branch. All



Figure 7. **Motion-conditioned generation: a ballet example.** We extend Fig. 4(b)–(j) of the main paper by comparing RawMo with OmniControl, MaskControl, and ACMDM on nine motion-conditioning tasks. All rows use one HumanML3D sequence and its text prompt, with different motion conditions. In the visual comparisons, orange marks the given conditions and blue shows generated motion. Err. reports the mean 3D positional error in centimeters at the conditioned joint positions for the shown example. A single conditional RawMo model preserves the conditions exactly across all nine tasks.

evaluations use the same 1,024 test motions and the first text prompt for each motion. For each motion and timestep $\tau \in \{0.02, 0.1, 0.2, 0.4, 0.6, 0.8, 0.95\}$, the noise sample is fixed across seeds.

Motion feature representation. On HumanML3D and KIT-ML, RawMo directly generates the motion features introduced by Guo *et al.* [8]. For a skeleton with J joints, each frame contains the root angular velocity about the vertical axis, two planar root-velocity components, root height, the positions and 6-D rotations of the $J-1$ non-root joints, the velocities of all J joints, and four foot-contact indicators. The non-root joint positions are expressed in the root's heading frame relative to its projection onto the ground plane. This representation has $D = 12J - 1$ channels per frame: 263 for HumanML3D with $J = 22$ and 251 for KIT-ML with $J = 21$. We standardize each channel using training-split statistics. Root heading and planar position are recovered by integrating the root velocities.

Network architecture. Table 12 summarizes the architecture, training, and sampling settings. The backbone $\mathcal{B}_\theta$ uses dual-stream and single-stream blocks with an MLP ratio of 4, temporal RoPE [31], sigmoid gating on attention outputs, and RMSNorm before and after each branch. The global conditioning vector modulates each block through adaptive normalization. Each body-region token pools three to five joints over four frames and receives a learned region embedding. The 12 regions cover the pelvis, spine, chest, head, and bilateral hips, legs, shoulders, and arms. Adjacent regions share a junction joint. Each frame token is formed by projecting all D features through a rank-128 bot-

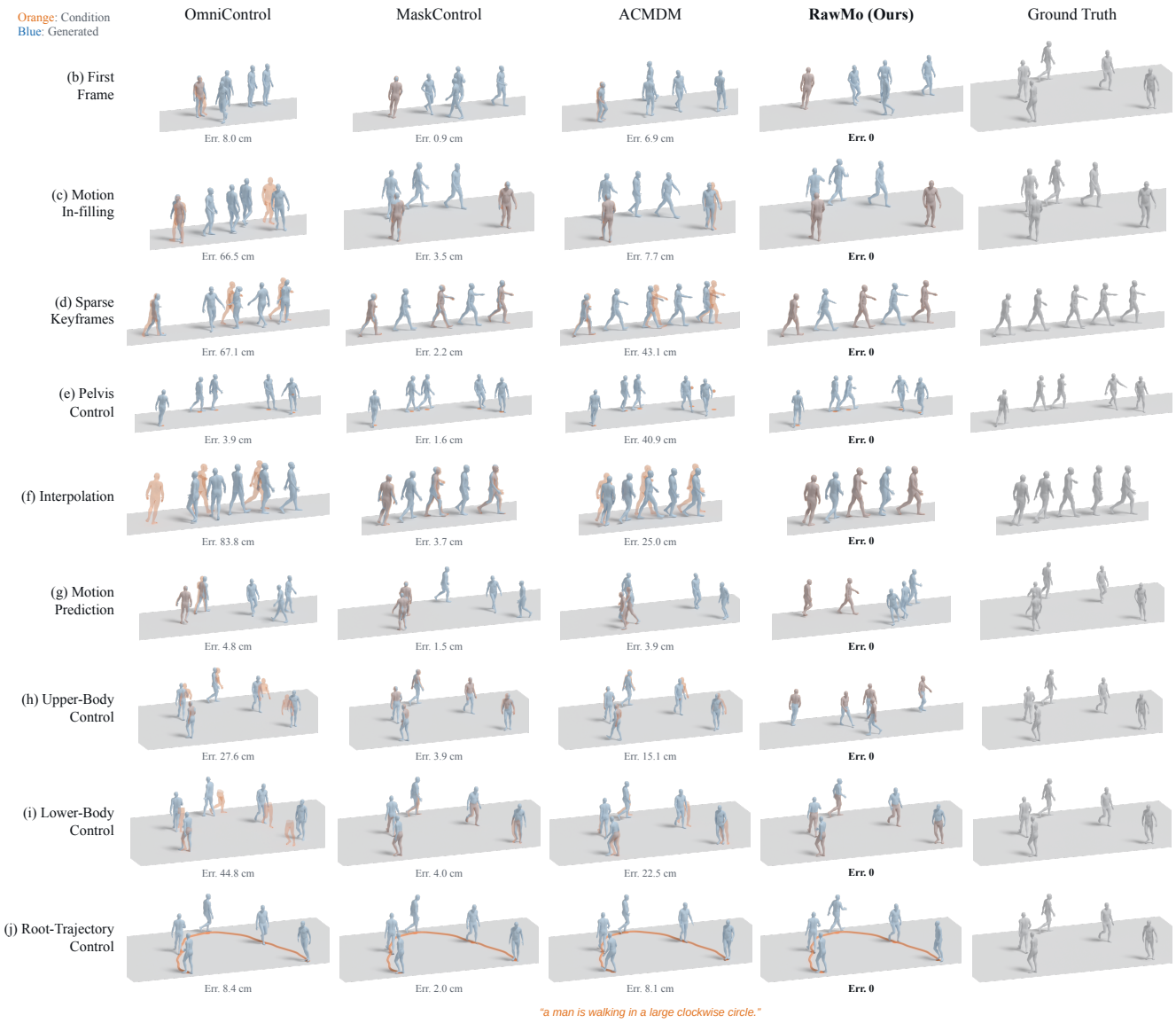

Figure 8. **Motion-conditioned generation: a clockwise walking example.** We compare RawMo with the baseline methods on a HumanML3D walking sequence under different motion conditions. The bottom row provides the horizontal root trajectory as the motion condition. RawMo records zero error at the conditioned joint positions across all nine tasks. The task order, color conventions, and error metric follow Fig. 7.

tleneck. In each refinement block, the $J \times 32$ joint features of each frame are compressed into a 384-D vector. Four-head temporal attention with RoPE mixes these vectors across frames. The block then expands the resulting vectors back into joint features and applies a joint-wise MLP. A frozen CLIP ViT-B/32 [28] encodes the text prompt. All text-to-motion results use dedicated T2M models, while all motion-conditioned results use a single conditional RawMo model. The conditional model is trained on HumanML3D without HiNCE.

Generator training. To sample training timesteps, we first draw an initial noise fraction ρ from a logit-normal distribution. We denote the resolution-shift factor by κ and determine the training timestep τ using the resolution shift [6]:

$$1 - \tau = \frac{\kappa\rho}{1 + (\kappa - 1)\rho}.$$

Table 12 lists the logit-normal parameters and shift factor. During training, we replace the denominator $1 - \tau$ in Eq. 2 with $\max(1 - \tau, 0.05)$ when converting the clean-motion prediction to flow velocity.

We use AdamW [19] with a learning rate of 2×10^{-4} and weight decay 0.01. The learning rate is warmed up over

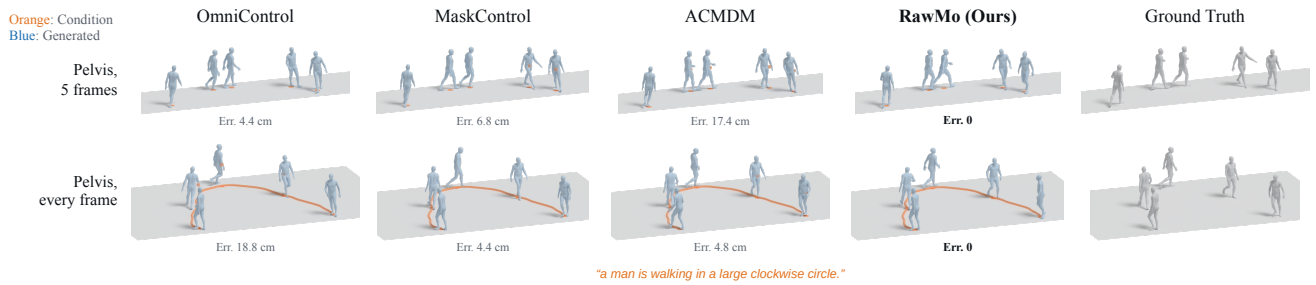

Figure 9. **Pelvis control: a clockwise walking example.** The pelvis-control visual comparisons on HumanML3D complement the quantitative results in Table 4 of the main paper. Given the pelvis positions, each method generates full-body motion using the text prompt shown in the figure. The top row provides pelvis positions at five randomly selected frames, while the bottom row provides them at every frame. Orange marks the given conditions and blue shows generated motion. RawMo preserves the conditioned pelvis positions exactly in both settings, with zero error at the controlled frames.

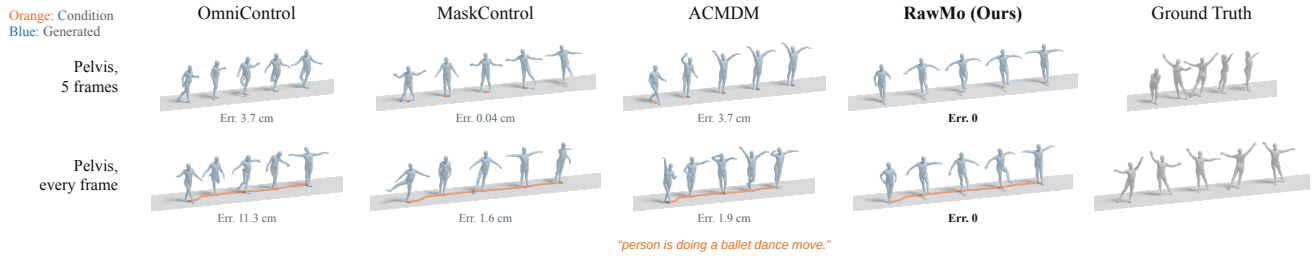

Figure 10. **Pelvis control: a ballet example.** The observation settings and visualization conventions follow Fig. 9. All methods receive the given pelvis positions and the text prompt shown in the figure. The baselines show nonzero positional errors under both sparse and dense observations, while RawMo preserves the conditioned pelvis positions exactly.

2,000 steps, held constant, and finally decayed linearly to zero. Training uses bfloat16 precision and gradient-norm clipping at 1.0. Text prompts are dropped with probability 0.1 for classifier-free guidance. Sampling uses an exponential moving average of the generator weights with decay 0.9999.

HiNCE implementation. HiNCE uses a motion encoder and a text encoder. The motion encoder projects standardized motion features to 256 dimensions, reduces temporal resolution by a factor of four using two strided 1-D convolutions, and applies a four-layer, four-head Transformer. The text encoder applies a two-layer Transformer to frozen CLIP token features. Both encoders use mean pooling and output unit-normalized 512-D embeddings.

We jointly train the encoders with symmetric InfoNCE [22] and a learned temperature initialized to 0.07. Training uses a batch size of 256 and a learning rate of 10^{-4}, with 1,000 warm-up steps followed by cosine decay. The training budget is 12k steps on HumanML3D and 6k on KIT-ML. Checkpoints are selected by validation R@3, choosing step 3k for HumanML3D and step 1k for KIT-ML. Following TMR [23], we mask near-duplicate negatives when the cosine similarity between the CLIP end-of-text features of the negative and positive captions exceeds 0.9. During text-to-motion generator training, both encoders remain frozen. The HiNCE loss is applied at high-noise timesteps $\tau \leq 0.25$, excluding queries with dropped text prompts and masking the same near-duplicate negatives. To account for the motion encoder's temporal stride, we average the losses for the four input offsets $0, 1, 2, 3$.

Motion-conditioning inputs. The conditional model receives the given motion features $\mathbf{x}_c$, their feature mask $\mathbf{m}$, and six root-conditioning channels. The six root-conditioning channels comprise two position channels, two heading channels, and two availability masks. The position channels encode the root's horizontal position in global coordinates as $(x, z)/(2\,\mathrm{m})$. The heading channels encode the root heading angle θ as $(\cos\theta, \sin\theta)$. One availability mask indicates whether the root position is given, and the other indicates whether the root heading is given. Zero-initialized linear projections inject these conditions into the joint features used for region pooling, the frame tokens, and the raw joint tokens.

Conditional training tasks. The HumanML3D conditional model is trained on a mixture of text-only and motion-conditioned generation tasks. The temporal training tasks correspond to Fig. 4(b,d,f,g) of the main paper: (b) the first frame; (d) periodic keyframes with a stride of 2–32

Table 12. **RawMo architecture, training, and sampling settings.** T2M denotes text-to-motion generation, and Cond. denotes motion-conditioned generation. The Heun row reports the number of sampling intervals, with the corresponding number of network evaluations in parentheses. CFG denotes the classifier-free guidance scale.

	HumanML3D T2M	**KIT-ML T2M**	**HumanML3D Cond.**
Parameters	45.7M	49.9M	50.5M
Backbone blocks (dual + single)	$8+4$	$8+4$	$8+4$
Width C / heads	$384/6$	$384/6$	$384/6$
Refinement-head blocks	2	4	2
Extra inputs	–	–	$\mathbf{x}_c$, $\mathbf{m}$, root condition
HiNCE weight λ / window	0.03 / $\tau \leq 0.25$	0.02 / $\tau \leq 0.25$	not used
Training steps (batch size 128)	30k	20k	30k
AdamW (β_1, β_2)	$(0.9, 0.95)$	$(0.9, 0.99)$	$(0.9, 0.95)$
Linear decay over final	40%	25%	40%
Noise-level sampling (μ, σ) / shift	$(0.8, 1.2)/5$	$(0.8, 0.8)/3$	$(0.8, 1.2)/5$
Heun intervals (network evaluations)	16 (31)	50 (99)	16 (31)
CFG	3.5	3.5	3

frames or random keyframes with a count selected from $\{1, 2, 3, 5, 8, 12, 20, 30\}$; (f) both endpoint frames; and (g) a prefix covering the first 10–50% of the clip. Spatial conditioning covers (h) upper-body control, (i) lower-body control, (j) root-path control, and (e) pelvis control. For pelvis control, we specify positions at 1, 2, or 5 frames, or at 10%, 25%, 50%, or 100% of the frames. Motion in-filling in Fig. 4(c) is evaluated zero-shot.

Preserving pelvis positions. Pelvis control specifies the pelvis position at selected frames. At each network evaluation, the global skip copies the given pelvis height into the predicted clean motion. Horizontal pelvis positions are recovered by converting the planar root velocities to global coordinates and integrating them over motion frames. We denote the corrected horizontal displacement from frame t to $t+1$ in global coordinates by $\Delta\widetilde{\mathbf{p}}_t$. The i-th controlled frame in temporal order has frame index k_i and given horizontal position $\mathbf{p}_{k_i}$. For consecutive controlled frames $k_i < k_{i+1}$, the displacements must satisfy

$$\sum_{t=k_i}^{k_{i+1}-1} \Delta\widetilde{\mathbf{p}}_t = \mathbf{p}_{k_{i+1}} - \mathbf{p}_{k_i}.$$

We adjust the planar root-velocity entries between these frames that are not fixed by the feature mask $\mathbf{m}$, so that the accumulated displacement satisfies this constraint. The corrected clean prediction is then converted into a flow velocity using Eq. 2 of the main paper. The sampler uses this velocity to update the noisy motion at the next sampling step.

Sampling. We apply classifier-free guidance to the flow velocity and integrate from noise at $\tau = 0$ to clean motion at $\tau = 1$. The sampling grid is obtained by applying the same resolution shift used for training to uniformly spaced noise fractions from 1 to 0.001. Across N sampling intervals, Heun's method [14] followed by a final Euler step requires $2N-1$ network evaluations. Text-to-motion guidance scales are selected on the validation split.

H. Limitations

Zero-shot motion control. RawMo performs motion control using a conditional model trained separately from the dedicated text-to-motion model. Enabling the text-to-motion model to follow motion conditions without additional training remains an open problem.

Sampling latency. RawMo generates motion through iterative denoising. As reported in Table 11, generating a single HumanML3D clip on an NVIDIA RTX 4090 takes a median of 0.594 s. Future work could focus on reducing the number of sampling steps while preserving motion quality, with the goal of lowering generation latency and improving responsiveness in interactive applications.

References

[1] 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)*, pages 18000–18010. IEEE, 2023. 1, 3, 6, 7, 10, 12

[2] Zhennan Chen, Junwei Zhu, Xu Chen, Jiangning Zhang, Xiaobin Hu, Hanzhen Zhao, Chengjie Wang, Jian Yang, and Ying Tai. Dip: Taming diffusion models in pixel space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 36136–36146, 2026. 2, 3

[3] 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*, pages 1–9, 2024. 3

[4] Wenxun Dai, Ling-Hao Chen, Jingbo Wang, Jinpeng Liu, Bo Dai, and Yansong Tang. Motionlcm: Real-time controllable motion generation via latent consistency model. In *European Conference on Computer Vision*, pages 390–408. Springer, 2024. 3, 7, 8

[5] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 2

[6] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024. 14

[7] 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 *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13336–13348. IEEE, 2025. 9, 10

[8] 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)*, pages 5142–5151. IEEE, 2022. 1, 2, 4, 6, 7, 8, 10, 11, 13

[9] 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)*, pages 1900–1910. IEEE, 2024. 1, 3, 6, 7, 10, 12

[10] Ziyan Guo, Zeyu Hu, De Wen Soh, and Na Zhao. Motionlab: Unified human motion generation and editing via the motion-condition-motion paradigm. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13869–13879, 2025. 3

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2

[12] Seokhyeon Hong, Chaelin Kim, Serin Yoon, Junghyun Nam, Sihun Cha, and Junyong Noh. Salad: Skeleton-aware latent diffusion for text-driven motion generation and editing. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7158–7168. IEEE, 2025. 3, 6, 7, 10, 12

[13] Seong-Eun Hong, Soobin Lim, Juyeong Hwang, Minwook Chang, and Hyeongyeop Kang. Bipo: Bidirectional partial occlusion network for text-to-motion synthesis. In *2026 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 32–42. IEEE, 2026. 6, 7, 10, 12

[14] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022. 7, 10, 16

[15] 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)*, pages 2151–2162. IEEE, 2023. 3, 7, 8

[16] Black Forest Labs. Flux. `https://github.com/black-forest-labs/flux`, 2024. 5

[17] 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)*, 2026. 3, 6, 7, 10, 12

[18] Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 36115–36125, 2026. 2, 3, 9

[19] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 14

[20] Zichong Meng, Zeyu Han, Xiaogang Peng, Yiming Xie, and Huaizu Jiang. Absolute coordinates make motion generation easy. *arXiv preprint arXiv:2505.19377*, 2025. 6, 7, 8, 9, 10, 12

[21] Zichong Meng, Yiming Xie, Xiaogang Peng, Zeyu Han, and Huaizu Jiang. Rethinking diffusion for text-driven human motion generation: Redundant representations, evaluation, and masked autoregression. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 27859–27871. IEEE, 2025. 6, 7, 8, 10, 11, 12

[22] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 5, 15

[23] Mathis Petrovich, Michael J Black, and Gül Varol. Tmr: Text-to-motion retrieval using contrastive 3d human motion synthesis. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9454–9463. IEEE, 2023. 15

[24] Ekkasit Pinyoanuntapong, Muhammad Usama Saleem, Pu Wang, Minwoo Lee, Srijan Das, and Chen Chen. Bamm: Bidirectional autoregressive motion model. In *European Conference on Computer Vision*, pages 172–190. Springer, 2024. 3, 6, 7, 10, 12

[25] 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)*, pages 1546–1555. IEEE, 2024. 6, 7, 10, 12

[26] Ekkasit Pinyoanuntapong, Muhammad Saleem, Korrawe Karunratanakul, Pu Wang, Hongfei Xue, Chen Chen, Chuan Guo, Junli Cao, Jian Ren, and Sergey Tulyakov. Maskcontrol: Spatio-temporal control for masked motion synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9955–9965, 2025. 7, 8, 9, 10

[27] Matthias Plappert, Christian Mandery, and Tamim Asfour. The kit motion-language dataset. *Big data*, 4(4):236–252, 2016. 1, 2, 6, 9

[28] A Radford. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021. 4, 5, 14

[29] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 10674–10685. ieee, 2022. 2

[30] Yonatan Shafir, Guy Tevet, Roy Kapon, and Amit Bermano. Human motion diffusion as a generative prior. In *International Conference on Learning Representations*, pages 8717–8733, 2024. 3, 7, 8

[31] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. 5, 13

[32] 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. 1, 3, 5, 6, 7, 8, 10, 12

[33] Weilin Wan, Zhiyang Dou, Taku Komura, Wenping Wang, Dinesh Jayaraman, and Lingjie Liu. Tlcontrol: Trajectory and language control for human motion synthesis. In *European Conference on Computer Vision*, pages 37–54. Springer, 2024. 3, 7, 8

[34] Zhenzhi Wang, Jingbo Wang, Yixuan Li, Dahua Lin, and Bo Dai. Intercontrol: Zero-shot human interaction generation by controlling every joint. *Advances in Neural Information Processing Systems*, 37:105397–105424, 2024. 7, 8

[35] Akihisa Watanabe, Qing Yu, Edgar Simo-Serra, and Kent Fujiwara. Projflow: Projection sampling with flow matching for zero-shot exact spatial motion control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. 3, 7, 8

[36] 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 *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10086–10096. IEEE, 2025. 9, 10

[37] Yiming Xie, Varun Jampani, Lei Zhong, Deqing Sun, and Huaizu Jiang. Omnicontrol: Control any joint at any time for human motion generation. In *International Conference on Learning Representations*, pages 28176–28194, 2024. 3, 6, 7, 8, 9, 10

[38] Qing Yu and Kent Fujiwara. Motionspaceflow: Representation-aware flow matching in direct motion space. *arXiv preprint arXiv:2609.34190*, 2026. 3

[39] Qing Yu, Akihisa Watanabe, and Kent Fujiwara. Causal motion diffusion models for autoregressive motion generation. In *CVPR*, 2026. 3, 6, 7, 9, 10, 12

[40] Yongsheng Yu, Wei Xiong, Weili Nie, Yichen Sheng, Shiqiu Liu, and Jiebo Luo. Pixeldit: Pixel diffusion transformers for image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14273–14282, 2026. 2, 5

[41] Weihao Yuan, Yisheng He, Weichao Shen, Yuan Dong, Xiaodong Gu, Zilong Dong, Liefeng Bo, and Qixing Huang. Mogents: Motion generation based on spatial-temporal joint modeling. *Advances in Neural Information Processing Systems*, 37:130739–130763, 2024. 3, 6, 7, 10, 12

[42] Ling-An Zeng, Guohong Huang, Gaojie Wu, and Wei-Shi Zheng. Light-t2m: A lightweight and fast model for text-to-motion generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 9797–9805, 2025. 6, 7, 10, 12

[43] Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in neural information processing systems*, 32, 2019. 11

[44] 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*, pages 14730–14740, 2023. 1, 3, 6, 7, 10, 12

[45] Jiahao Zhang, Joseph Liu, Young-Yoon Lee, Seonghyeon Moon, Victor Zordan, Guy Tevet, C Karen Liu, Stephen Gould, Oren Jacob, Haomiao Jiang, et al. Romo: A large-scale, richly organized dataset and semantic taxonomy for human motion generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16408–16419, 2026. 9

[46] 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)*, pages 364–373. IEEE, 2023. 3, 6, 7, 10, 11, 12

[47] 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. 3, 6, 7, 10, 12

[48] Zongye Zhang, Bohan Kong, Qingjie Liu, and Yunhong Wang. Towards robust and controllable text-to-motion via masked autoregressive diffusion. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 9326–9335, 2025. 6, 7