MoTween: In-Betweening for Part-wise Text-to-Motion Generation

Anonymous authors
Paper under double-blind review

Abstract

Text-to-motion generation aims to synthesize semantically consistent 3D human motions from natural language descriptions. Although recent approaches have achieved remarkable progress, existing methods either overlook the inherent hierarchical structure of the human body or struggle to capture the temporal dynamics described in the text. To address these limitations, we propose MoTween, a novel two-stage framework consisting of Hierarchical Vector Quantized Variational Autoencoder (Hierarchical VQ-VAE) and In-Betweening Motion Generation. In the first stage, we introduce a Hierarchical VQ-VAE that decomposes each pose into fine-grained body parts and coarse body groups, propagating group-level context into part representations to reflect the inherent body structure. In the second stage, we generate motion representations at semantically critical keyframes and subsequently complete the full sequence by in-betweening the remaining frames, enabling temporally coherent generation. Experiments on the `HumanML3D` and `KIT-ML` datasets demonstrate that MoTween outperforms existing state-of-the-art methods on both benchmarks, while also enabling flexible part-level motion editing. *The code will be released upon publication.*

1 Introduction

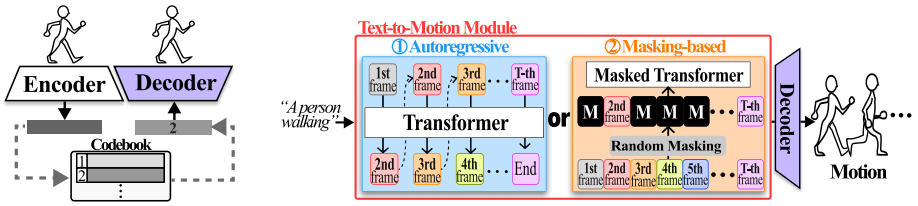
Text-to-motion generation aims to synthesize 3D human motions from textual descriptions, with applications in animation, gaming, virtual reality, and robotics, driven by large-scale motion datasets (Guo et al., 2020; 2022a; Plappert et al., 2016) and advances in generative modeling (Ho et al., 2020; Ramesh et al., 2021). Among recent approaches, VQ-VAE-based methods (Guo et al., 2024; Pinyoanuntapong et al., 2024; Yuan et al., 2024; Zhang et al., 2023a) adopt a two-stage pipeline (Fig. 1): discretizing motion into codebook tokens, then generating text-conditioned token sequences.

This framework has limitations in both stages. In Stage 1, conventional VQ-VAEs tokenize motion at the whole-body level (Fig. 1(a)), where subtle part motions are dominated by larger joint movements. Body-part-level (Cao et al., 2025; Ghosh et al., 2021; Zou et al., 2024) or joint-level (Yuan et al., 2024) quantization mitigates this, but uses flat, component-wise tokenization that overlooks inter-part dependencies and the hierarchical structure of the human body. In Stage 2 (Fig. 1(b)), autoregressive methods (Zhang et al., 2023a) suffer from error accumulation in long sequences, while masking-based methods (Guo et al., 2024; Pinyoanuntapong et al., 2024) can mask crucial transition points and reconstruct them without sufficient temporal context.

To address these limitations, we propose MoTween, a two-stage framework for text-to-3D human motion generation. In Stage 1, a Hierarchical VQ-VAE injects group-level body context into part representations through hierarchical cross-attention, reflecting the human body hierarchy. In Stage 2, a Keyframe-Guided Motion Transformer first predicts part-wise keyframe representations and then in-betweens the remaining frames, rather than generating all frames uniformly, improving temporal coherence and robustness for long sequences.

Experiments on `HumanML3D` (Guo et al., 2022a) and `KIT-ML` (Plappert et al., 2016) show that MoTween achieves state-of-the-art performance, with the best FID and R-Precision on both benchmarks. We also demonstrate that its part-wise representations enable part-level motion editing. The main contributions of this work are summarized as follows:

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.



(a) Stage1. Motion VQ-VAE (b) Stage2. Text-to-Motion

Figure 1: **Overview of the conventional VQ-VAE(Van Den Oord et al., 2017)-based two-stage approaches for text-to-motion generation.** (a) Stage 1: Motion VQ-VAE tokenizes motion at the whole-body level. (b) Stage 2: Text-to-Motion module follows either 1 Autoregressive generation or 2 Masking-based generation.

- We propose a Hierarchical VQ-VAE that reflects the hierarchical structure of the human body via hierarchical cross-attention.
- We introduce a novel motion generation approach that predicts part-wise keyframes at semantically important time points and generates the full motion sequence by in-betweening the remaining frames.
- Experiments on `HumanML3D` and `KIT-ML` demonstrate that MoTween achieves state-of-the-art performance and enables flexible part-level motion editing.

2 Related Works

2.1 Text-to-Image and Text-to-Video Generation

Text-to-motion generation has been strongly influenced by text-to-image and text-to-video generation. After early GAN-based methods (Goodfellow et al., 2014; Chang et al., 2023; Reed et al., 2016; Xu et al., 2018; Zhang et al., 2017) suffering from mode collapse and unstable training, two paradigms have dominated: autoregressive models such as DALL-E (Ramesh et al., 2021), which predict VQ-VAE tokens (Van Den Oord et al., 2017) sequentially with a text-conditioned Transformer (Vaswani et al., 2017), and diffusion models such as Stable Diffusion (Rombach et al., 2022), which perform text-conditioned latent denoising. Both have been extended to text-to-video generation (Yan et al., 2021; Brooks et al., 2024; Ho et al., 2022; Singer et al., 2023), where temporal dynamics must also be modeled, and are widely adopted in human motion generation.

2.2 Text-to-3D Human Motion Generation

Early methods aligned text and motion in a shared latent space (Ahuja & Morency, 2019; Petrovich et al., 2022; Tevet et al., 2022), but modality mismatch limited quality. Recent methods are VQ-VAE (Van Den Oord et al., 2017)-based or diffusion (Ho et al., 2020)-based. VQ-VAE-based methods (Cao et al., 2025; Guo et al., 2022b; 2024; Jiang et al., 2023; Petrovich et al., 2021; Pinyoanuntapong et al., 2024; Yuan et al., 2024; Zhang et al., 2023a; Zou et al., 2024) quantize motion into discrete tokens and generate them conditioned on text. Most tokenize the whole body holistically; body-part-level (Cao et al., 2025; Zou et al., 2024) or joint-level (Yuan et al., 2024) methods quantize each component independently without modeling inter-part dependencies. For generation, autoregressive (Zhang et al., 2023a) and masking-based (Guo et al., 2024; Pinyoanuntapong et al., 2024) methods both treat all time steps equally, ignoring key motion transitions. Diffusion-based methods (Chen et al., 2023; Dabral et al., 2023; Kong et al., 2023; Tevet et al., 2023; Zhang et al., 2024) achieve strong diversity and quality but require many sampling steps; discrete diffusion (Chi et al., 2024; Lou et al., 2023) reduces this cost in token space yet still relies on iterative sampling and inherits the limitations of quantization.

3 Proposed Methods

MoTween consists of two stages (Fig. 2). In Stage 1, the Hierarchical Vector Quantized Variational Autoencoder (Hierarchical VQ-VAE) converts continuous motion into discrete tokens while reflecting

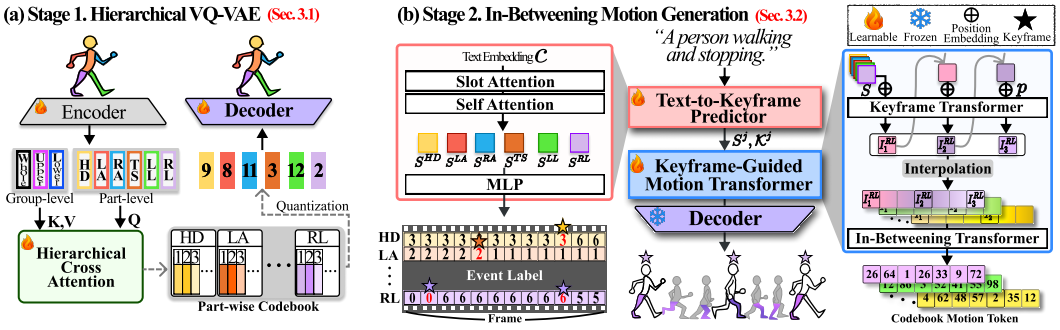


Figure 2: **Overview of the proposed MoTween.** (a) Hierarchical VQ-VAE decomposes each pose into six body parts and three body groups, (b) In-Betweening Motion Generation predicts part-wise events from text, selects keyframes at event transitions, and completes the sequence by interpolating and in-betweening the remaining frames.

the hierarchical structure of the human body, capturing both global dynamics and fine-grained local details. In Stage 2, In-Betweening Motion Generation predicts keyframe tokens at semantically important time points conditioned on the input text, and in-betweening fills the remaining tokens.

3.1 Hierarchical VQ-VAE for Part-wise Codebook Learning

A conventional VQ-VAE (Guo et al., 2022a;b) used in text-to-motion generation discretizes a motion sequence $M = \{m_t\}_{t=1}^{T}$ into a sequence of discrete motion tokens. Each pose m_t encodes joint-wise motion information at time step t (e.g., joint positions, rotations, and velocities). Specifically, an encoder maps m_t to a latent vector, which is quantized to its nearest codebook entry; the corresponding code index serves as the motion token, and a decoder reconstructs the pose from the quantized latent as shown in Fig. 1(a). However, compressing such a complex pose m_t into a single latent vector often captures major joint movements but misses fine-grained details of subtly moving body parts.

3.1.1 Hierarchical Body Modeling.

To address this problem, we propose a Hierarchical VQ-VAE that leverages the inherent hierarchy of the human body. As illustrated in Fig. 2(a), we represent each pose m_t at two levels: a fine part level and a coarse group level. At the fine part-level, we partition the pose m_t into six body parts:

$$m_t = \{m_t^j\}_{j \in \mathcal{P}}, \quad \text{where} \quad \mathcal{P} = \{\text{HD, LA, RA, TS, LL, RL}\}, \tag{1}$$

where TS denotes the torso (global position), and HD, LA, RA, LL, RL denote head, left/right arm, and left/right leg, respectively. At the coarse group-level, we define three body groups–whole, upper, and lower body–as subsets of $\mathcal{P}$:

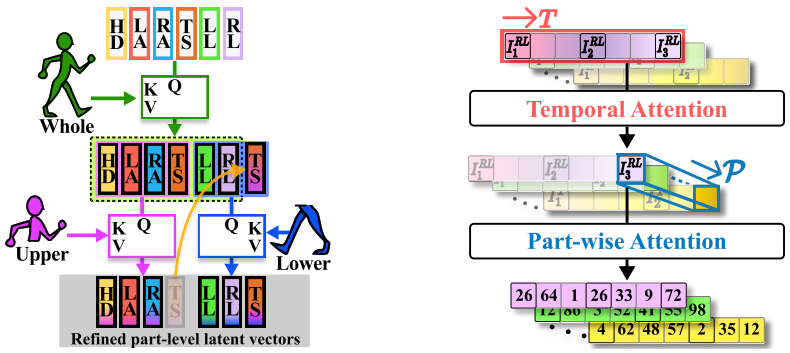
$$\mathcal{P}_{\text{Whole}} = \mathcal{P}, \quad \mathcal{P}_{\text{Upper}} = \{\text{HD, LA, RA, TS}\}, \quad \mathcal{P}_{\text{Lower}} = \{\text{TS, LL, RL}\}. \tag{2}$$

We encode the six body parts and three body groups using nine independent component-specific encoders, producing part-level and group-level latent vectors. Despite the varying input dimensions, all encoders are designed to output a latent vector in the same d-dimensional space, yielding $\{z_t^j\}_{j \in \mathcal{P}}$ for the six fine-grained parts and z_t^{Whole}, z_t^{Upper}, z_t^{Lower} for the three group-level components.

3.1.2 Hierarchical Cross-Attention.

While these latent vectors capture part-specific motion cues, they are learned independently and thus do not model inter-part dependencies or global body coherence. To address this, we apply a hierarchical cross-attention scheme that progressively refines the part-level latent vectors, as shown in Fig. 3(a). In each cross-attention module, the fine part-level latent vectors serve as the query, while the coarse group-level latent vectors serve as the key and value to guide their contextual refinement with global body information.

In our Hierarchical VQ-VAE, all six part latent vectors first attend to the whole-body group latent to absorb global body context. They are then refined by cross-attending to the corresponding sub-group



(a) Hierarchical Cross-Attention (b) In-Betweening Transformer

Figure 3: **Detailed architecture of the proposed modules.** (a) Hierarchical cross-attention refines part-level latents using whole-, upper-, and lower-body group latents. (b) The In-Betweening Transformer applies self-attention along the temporal and then part axes.

latent: upper-body parts (head, torso, left arm, right arm) attend to $\mathcal{P}_{\text{Upper}}$, while lower-body parts (torso, left leg, right leg) attend to $\mathcal{P}_{\text{Lower}}$. Since the torso belongs to both sub-groups (as well as $\mathcal{P}_{\text{Whole}}$), it attends to both $\mathcal{P}_{\text{Upper}}$ and $\mathcal{P}_{\text{Lower}}$, successively accumulating whole-body, upper-body, and lower-body context. This hierarchical propagation enriches each part-level latent vector with hierarchical group context, enabling more body-aware quantization.

3.1.3 Part-wise Codebook Learning.

These refined part-level latent vectors are quantized via part-specific codebooks by mapping each latent to its nearest codebook entry, and the resulting code indices are used as discrete motion tokens. A decoder then reconstructs the original motion sequence from the quantized latent vectors. We train the hierarchical VQ-VAE end-to-end by minimizing the reconstruction error between the input and reconstructed motions, enabling accurate discretization of continuous motion into compact token sequences. After training, the decoder and codebooks are frozen and reused in Stage 2 (Sec. 3.2) to support text-to-motion generation.

3.2 In-Betweening Motion Generation with Keyframes

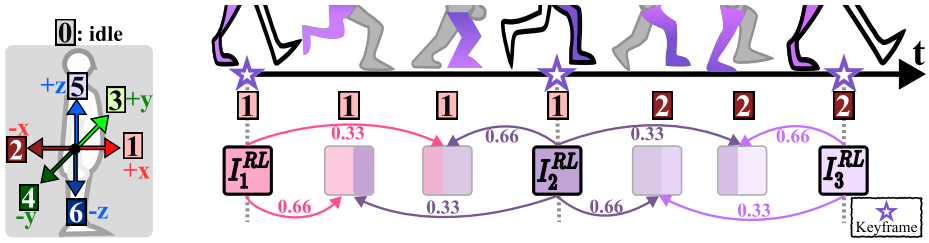
Building upon the hierarchical codebook learned in Stage 1, Stage 2 generates human motion from the input text (Fig. 2(b)). Unlike prior methods (Jiang et al., 2023; Zhang et al., 2023a; Zhong et al., 2023) that generate all motion frames uniformly, MoTween predicts semantically important part-wise keyframe motions, then interpolates between them to form the complete sequence. To this end, MoTween consists of a Text-to-Keyframe Predictor and a Keyframe-Guided Motion Transformer.

3.2.1 Text-to-Keyframe Predictor.

We first define motion events based on primary movement patterns along the x-, y-, and z-axes. For each body part, an event corresponds to a temporal segment during which the dominant motion direction or kinematic trend remains consistent. Part-wise keyframes are defined as the time steps where this motion event changes, marking semantically meaningful transitions in motion dynamics.

• **Part-wise Motion Event Labeling:** To construct ground-truth labels for text-to-keyframe predictor training, we assign each body part $j \in \mathcal{P}$ an event class at every frame t based on its dominant velocity direction between consecutive frames. Specifically, each part is classified into one of seven discrete event classes $\text{cls}_t^j \in \{0, \ldots, 6\}$: Idle, or positive/negative motion along each spatial axis as shown in Fig. 4(a).

Although 3D motion can involve simultaneous movements along multiple axes, our goal is not to model full multi-axis directions but to detect event changes that define part-wise keyframes. While this labeling ignores diagonal motion, the dominant axis captures the principal movement direction and provides sufficiently reliable supervision for keyframe prediction. Additional details for the motion event labeling are provided in the appendix.



(a) (b)

Figure 4: **Motion event labeling and keyframe interpolation.** (a) Part-wise motion events fall into seven classes: idle or motion along $\pm x$, $\pm y$, and $\pm z$. (b) Keyframes (★) mark transitions between consecutive events; intermediate frames are linearly interpolated between adjacent keyframes.

• **Architecture:** The Text-to-Keyframe Predictor takes a text description as input, encodes it into part-level representations, and predicts the event classes at each frame. As shown on the left of Fig. 2(b), the CLIP (Radford et al., 2021) text embedding C is decomposed via Slot Attention (Locatello et al., 2020) into part-aligned slots, enabling separate modeling of motion semantics for different parts. However, these slots do not explicitly capture inter-part dependencies. We therefore apply self-attention (Vaswani et al., 2017) across slots to enable contextual information exchange among parts. The resulting features $\mathcal{S} = \{S^j\}_{j\in\mathcal{P}}$ encode both individual motion semantics and their mutual relationships derived from the input text.

The resulting feature set $\mathcal{S}$ is fed into a lightweight multi-layer perceptron (MLP) to predict a motion event class cls for each body part at each of the t time steps. The predictor is trained using the event labels constructed as described above. Predicting a dense sequence of part-wise event classes provides stable supervision and avoids directly regressing discrete keyframe indices. Part-wise keyframes are then obtained as the time steps t where the predicted event class changes, naturally supporting a variable number of keyframes. We define a set of part-wise keyframes as $\mathcal{K}^j = \{t \in \{1,\dots,T-1\} \mid \text{cls}_t^j \neq \text{cls}_{t+1}^j\}$.

3.2.2 Keyframe-Guided Motion Transformer.

Given the predicted part-wise keyframes, we leverage them as sparse temporal anchors for motion generation. To this end, we introduce the Keyframe-Guided Motion Transformer, which consists of two components: a keyframe transformer and an in-betweening transformer. The keyframe transformer first generates motion-oriented latent representations at the predicted keyframe positions from the corresponding part-wise text embeddings. These representations serve as structural anchors that encode the core motion dynamics for each part. The in-betweening transformer then models the temporal dependencies across intermediate frames to produce a coherent full-length motion sequence.

• **Keyframe Transformer:** The keyframe transformer autoregressively generates part-wise keyframe representations by converting the part-wise text embedding S^j into motion-oriented latent representations at the predicted keyframe positions (see Fig. 2(b)). Given the keyframe index set $\mathcal{K}^j$, it produces one representation per keyframe in temporal order. We initialize the first input as $I_0^j = S^j$. For the i-th keyframe at time $\mathcal{K}^j(i)$, the transformer updates the representation by conditioning on the previous keyframe and adding an absolute temporal position embedding $\mathbf{p}$:

$$I_0^j = S^j, \quad I_{\mathcal{K}^j(i)}^j = \text{KFTransformer}\left(I_{\mathcal{K}^j(i-1)}^j + \mathbf{p}_{\mathcal{K}^j(i)}\right), \tag{3}$$

where $i \in \{1,\dots,|\mathcal{K}^j|\}$ denotes the keyframe order. In contrast to conventional approaches (Guo et al., 2024; Pinyoanuntapong et al., 2024; Yuan et al., 2024; Zhang et al., 2023a) that generate representations for every frame, our keyframe transformer generates a sparse set of K keyframe representations. This design allows the model to focus on semantically meaningful moments.

• **In-Betweening Transformer:** The In-Betweening Transformer completes the full motion sequence by generating intermediate representations between the predicted keyframes. For each part j, we construct a feature sequence of length T by placing the generated keyframe representations $I_{\mathcal{K}^j(i)}^j$ at their corresponding temporal positions. For time steps between two consecutive keyframes,

Table 1: **Evaluation on the `HumanML3D` (Guo et al., 2022a) and `KIT-ML` (Plappert et al., 2016).** $\rightarrow$ indicates closer to ground-truth motion is better. The best and runner-up values are **bold** and underlined.

Dataset	Methods	Top1 ↑	Top2 ↑	Top3 ↑	FID ↓	MM-Dist ↓	Diversity →
HumanML 3D	Ground Truth	$0.511^{\pm.003}$	$0.703^{\pm.003}$	$0.797^{\pm.002}$	$0.002^{\pm.000}$	$2.974^{\pm.008}$	$9.503^{\pm.065}$
	TEMOS (Petrovich et al., 2022)	$0.424^{\pm.002}$	$0.612^{\pm.002}$	$0.722^{\pm.002}$	$3.734^{\pm.028}$	$3.703^{\pm.008}$	$8.973^{\pm.071}$
	TM2T (Guo et al., 2022b)	$0.424^{\pm.003}$	$0.618^{\pm.003}$	$0.729^{\pm.002}$	$1.501^{\pm.017}$	$3.467^{\pm.011}$	$8.589^{\pm.076}$
	ReMoDiffuse (Zhang et al., 2023b)	$0.510^{\pm.005}$	$0.698^{\pm.006}$	$0.795^{\pm.004}$	$0.103^{\pm.004}$	$2.974^{\pm.016}$	$9.018^{\pm.075}$
	MotionGPT (Jiang et al., 2023)	-	-	-	0.567	3.775	9.006
	T2M-GPT (Zhang et al., 2023a)	$0.492^{\pm.003}$	$0.679^{\pm.002}$	$0.775^{\pm.002}$	$0.141^{\pm.005}$	$3.121^{\pm.009}$	$9.761^{\pm.081}$
	DiverseMotion (Lou et al., 2023)	$0.515^{\pm.003}$	$0.706^{\pm.002}$	$0.802^{\pm.002}$	$0.072^{\pm.004}$	$2.941^{\pm.007}$	$9.683^{\pm.102}$
	MMM (Pinyoanuntapong et al., 2024)	$0.504^{\pm.003}$	$0.696^{\pm.003}$	$0.794^{\pm.002}$	$0.080^{\pm.003}$	$2.998^{\pm.007}$	$9.411^{\pm.058}$
	MoMask (Guo et al., 2024)	$0.521^{\pm.002}$	$0.713^{\pm.002}$	$0.807^{\pm.002}$	$0.045^{\pm.002}$	$2.958^{\pm.008}$	-
	MoGenTS (Yuan et al., 2024)	$\underline{0.529}^{\pm.003}$	$\underline{0.719}^{\pm.002}$	$\underline{0.812}^{\pm.002}$	$\underline{0.033}^{\pm.001}$	$\underline{2.867}^{\pm.006}$	$\underline{9.570}^{\pm.077}$
	MoMask++ (Guo et al., 2025)	$0.528^{\pm.003}$	$0.718^{\pm.003}$	$0.811^{\pm.002}$	$0.072^{\pm.003}$	$2.912^{\pm.009}$	-
	MoTiGA (Chen et al., 2026)	$0.523^{\pm.002}$	-	$0.808^{\pm.003}$	$0.041^{\pm.002}$	$2.991^{\pm.009}$	-
	Ours	$\mathbf{0.537}^{\pm.003}$	$\mathbf{0.730}^{\pm.003}$	$\mathbf{0.818}^{\pm.002}$	$\mathbf{0.031}^{\pm.002}$	$\mathbf{2.861}^{\pm.010}$	$\mathbf{9.542}^{\pm.082}$
KIT-ML	Ground Truth	$0.424^{\pm.005}$	$0.649^{\pm.006}$	$0.779^{\pm.006}$	$0.031^{\pm.004}$	$2.788^{\pm.012}$	$11.080^{\pm.097}$
	TEMOS (Petrovich et al., 2022)	$0.353^{\pm.002}$	$0.561^{\pm.002}$	$0.687^{\pm.002}$	$3.717^{\pm.028}$	$3.417^{\pm.008}$	$10.84^{\pm.100}$
	TM2T (Guo et al., 2022b)	$0.280^{\pm.005}$	$0.463^{\pm.006}$	$0.587^{\pm.005}$	$3.599^{\pm.153}$	$4.591^{\pm.026}$	$9.473^{\pm.117}$
	ReMoDiffuse (Zhang et al., 2023b)	$0.427^{\pm.014}$	$0.641^{\pm.004}$	$0.765^{\pm.055}$	$0.155^{\pm.006}$	$2.814^{\pm.012}$	$10.80^{\pm.105}$
	MotionGPT (Jiang et al., 2023)	-	-	-	0.597	3.394	10.54
	T2M-GPT (Zhang et al., 2023a)	$0.416^{\pm.006}$	$0.627^{\pm.006}$	$0.745^{\pm.006}$	$0.514^{\pm.029}$	$3.007^{\pm.023}$	$10.86^{\pm.094}$
	DiverseMotion (Lou et al., 2023)	$0.416^{\pm.005}$	$0.637^{\pm.008}$	$0.760^{\pm.011}$	$0.468^{\pm.098}$	$2.892^{\pm.041}$	$10.873^{\pm.101}$
	MMM (Pinyoanuntapong et al., 2024)	$0.381^{\pm.005}$	$0.590^{\pm.006}$	$0.718^{\pm.005}$	$0.429^{\pm.019}$	$3.146^{\pm.019}$	$10.633^{\pm.097}$
	MoMask (Guo et al., 2024)	$0.433^{\pm.007}$	$0.656^{\pm.005}$	$0.781^{\pm.005}$	$0.204^{\pm.011}$	$2.779^{\pm.022}$	-
	MoGenTS (Yuan et al., 2024)	$\underline{0.445}^{\pm.006}$	$\underline{0.671}^{\pm.006}$	$\underline{0.797}^{\pm.005}$	$\underline{0.143}^{\pm.004}$	$\underline{2.711}^{\pm.024}$	$\underline{10.918}^{\pm.090}$
	MoMask++ (Guo et al., 2025)	$0.440^{\pm.006}$	$0.662^{\pm.005}$	$0.783^{\pm.004}$	$0.268^{\pm.009}$	$2.756^{\pm.024}$	-
	MoTiGA (Chen et al., 2026)	$0.443^{\pm.005}$	-	$0.790^{\pm.005}$	$0.180^{\pm.005}$	$2.792^{\pm.012}$	-
	Ours	$\mathbf{0.448}^{\pm.005}$	$\mathbf{0.676}^{\pm.005}$	$\mathbf{0.799}^{\pm.004}$	$\mathbf{0.137}^{\pm.003}$	$\mathbf{2.707}^{\pm.027}$	$\mathbf{10.921}^{\pm.105}$

the representations are initialized by linear interpolation based on their relative temporal distances as described in Fig. 4(b). The resulting initial full-length sequence, denoted as $Z^j \in \mathbb{R}^{T\times d}$, serves as the input to the In-Betweening Transformer.

The In-Betweening Transformer processes Z by alternately applying self-attention along the temporal axis (T) and across parts ($j \in \mathcal{P}$), thereby learning both temporal continuity and structural consistency as shown in Fig. 3(b). First, for each body part j, we apply self-attention along the temporal axis to refine the interpolated sequence:

$$\hat{Z}^j = \mathrm{SelfAttn}_T(Z^j) \quad \text{along axis } T, \quad Z^j \in \mathbb{R}^{T\times d}. \tag{4}$$

Then, at each time step t, we apply self-attention across body parts to model inter-part dependencies:

$$\bar{Z}_t = \mathrm{SelfAttn}_{\mathcal{P}}(\hat{Z}_t) \quad \text{along axis } \mathcal{P}, \quad \hat{Z}_t = \{\hat{Z}_t^j\}_{j\in\mathcal{P}} \in \mathbb{R}^{|\mathcal{P}|\times d}. \tag{5}$$

The final latent representation $\bar{Z}$ is fed into part-specific MLP heads to produce a probability distribution over codebook indices (i.e., motion tokens) for each part. The index with the highest probability is selected, and the corresponding codebook embedding is retrieved. The retrieved embeddings are then fed into the pre-trained Hierarchical VQ-VAE decoder (Sec. 3.1) to reconstruct the 3D human motion $\hat{M} = \{\hat{m}_t\}_{t=1}^T$.

3.3 Loss Functions

3.3.1 Training Objective of the Hierarchical VQ-VAE.

The Hierarchical VQ-VAE (Stage 1) is trained to reconstruct the input motion from its quantized representation using the standard VQ-VAE objective (Van Den Oord et al., 2017). The loss for body part j at time step t is defined as:

$$\mathcal{L}_{\mathrm{S1}} = \sum_{t=1}^{T}\sum_{j\in\mathcal{P}} \left(\|m_t^j - \hat{m}_t^j\|_1 + \|\,\mathrm{sg}(z_t^j) - e_t^j\|_2^2 + \beta\|z_t^j - \mathrm{sg}(e_t^j)\|_2^2 \right), \tag{6}$$

where z_t^j denotes the encoder output, e_t^j is the nearest codebook entry to z_t^j, $\hat{m}_t^j$ is the reconstructed pose, and $\mathrm{sg}(\cdot)$ denotes the stop-gradient operation. The three terms correspond to the reconstruction,

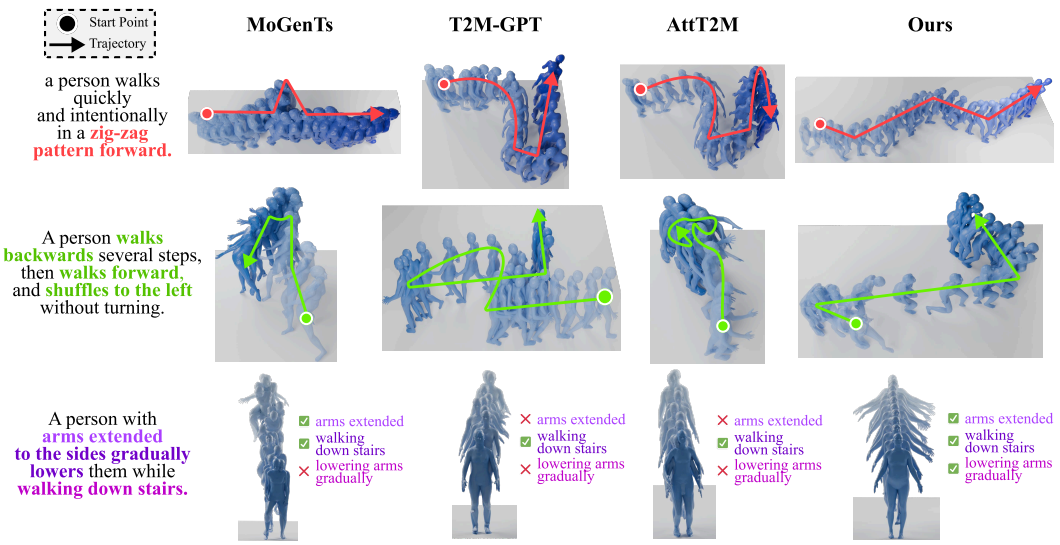


Figure 5: **Qualitative comparison with state-of-the-art methods.** We compare motion generation results of MoTween with prior methods under diverse, complex text prompts. Colors become darker as time progresses. MoTween produces motions that better match the prompt semantics.

codebook, and commitment losses, where the commitment loss is weighted by β. After training, the decoder and codebook are frozen and used in In-Betweening Motion Generation.

3.3.2 Training Objective of the In-Betweening Motion Generation.

The training objective of In-Betweening Motion Generation consists of two components: motion event prediction and motion token prediction. Both objectives minimize the negative log-likelihood of the ground-truth (GT) sequences, where $p_\theta(\cdot \mid C)$ denotes the model distribution conditioned on the text embedding C. The event loss supervises the GT event sequence E_{gt}, while the motion loss supervises the GT codebook indices Q obtained from the pre-trained Hierarchical VQ-VAE as

$$\mathcal{L}_{\text{event}} = -\mathbb{E}_{(E_{gt},C)\sim\mathcal{D}}\left[\log p_\theta(E_{gt}|C)\right], \ \mathcal{L}_{\text{motion}} = -\mathbb{E}_{(Q,C)\sim\mathcal{D}}\left[\log p_\theta(Q|C)\right], \tag{7}$$

where $\mathcal{D}$ denotes training dataset. The total loss for In-Betweening Motion Generation (Stage 2) is defined as: $\mathcal{L}_{\text{S2}} = \lambda\mathcal{L}_{\text{event}} + (1-\lambda)\mathcal{L}_{\text{motion}}$, where λ is a hyperparameter that controls the relative importance between the event prediction and motion prediction objectives.

4 Experimental Results

4.1 Datasets and Settings

Datasets. We evaluate on `HumanML3D` (Guo et al., 2022a), which contains 14,616 motions from AMASS (Mahmood et al., 2019) and HumanAct12 (Guo et al., 2020) with 44,970 text descriptions, and `KIT-ML` (Plappert et al., 2016), which contains 3,911 motions from KIT (Mandery et al., 2015) and CMU (Carnegie Mellon University) with 6,278 descriptions. Following prior work, motions are represented with global features (torso velocity, torso height, foot contact) and torso-centered local features (joint positions, velocities, rotations), yielding 263 and 251 dimensions, respectively. Both datasets use an 80%/5%/15% train/validation/test split.

Evaluation Metrics. We follow the protocol of (Guo et al., 2022a) with its pretrained motion and text feature extractors, repeating each experiment 20 times and reporting the mean with a 95% confidence interval (Guo et al., 2022a; 2024). R-Precision (Top-1,2,3) measures text-motion alignment in a 32-candidate retrieval setting; FID (Heusel et al., 2017) measures the distributional gap between generated and real motion features; MM-Dist is the average Euclidean distance between generated motion and text features; Diversity is the average pairwise distance among 300 random motion pairs, with values closer to ground truth being better.

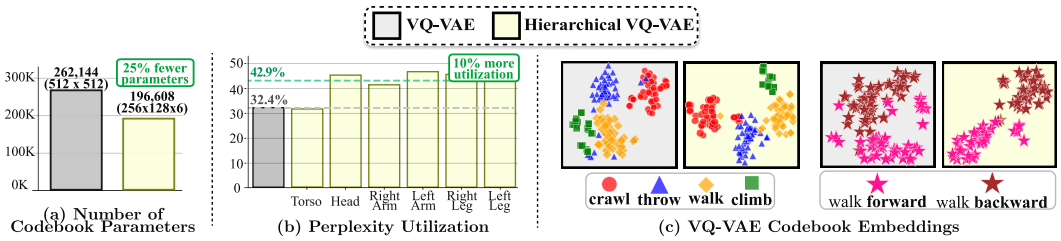


Figure 6: **Conventional vs. Hierarchical VQ-VAE.** (a) Codebook parameters. (b) Codebook perplexity (higher means better utilization). (c) t-SNE of the latent space for different action types (left) and different directions within the same walking action (right).

Implementation Details. The Hierarchical VQ-VAE (Sec. 3.1) uses independent 1D convolutional encoders that map part- and group-level pose sequences into a shared 128-dimensional latent space. Stage 2 (Sec. 3.2) uses the frozen CLIP ViT-B/32 text encoder (Radford et al., 2021) for 512-dimensional text embeddings, with λ=0.4. Both stages are trained with AdamW (learning rate 0.0001) on three NVIDIA A100 GPUs.

4.2 Comparison with State-of-the-Art Methods

We compare MoTween with existing state-of-the-art methods on `HumanML3D` (Guo et al., 2022a) and `KIT-ML` (Plappert et al., 2016), including diffusion-based methods (Lou et al., 2023; Zhang et al., 2023b), autoregressive methods (Guo et al., 2022b; Jiang et al., 2023; Zhang et al., 2023a), and masked prediction methods (Guo et al., 2024; Pinyoanuntapong et al., 2024; Yuan et al., 2024). All baseline results are reported from the original papers.

4.2.1 Text-to-Motion Evaluation.

Table 1 summarizes the quantitative results on `HumanML3D` (Guo et al., 2022a) and `KIT-ML` (Plappert et al., 2016). MoTween outperforms prior state-of-the-art methods on both datasets, achieving the best results across all evaluation metrics, including R-Precision, FID, and MM-Dist. These improvements stem from the keyframe-guided in-betweening strategy, which focuses on semantically important keyframes rather than treating all frames equally, leading to better temporal coherence.

Figure 5 presents qualitative comparisons on motion generation under complex text queries. These examples require synthesizing complex actions with spatiotemporal context, rather than a single simple motion. Compared to other methods, MoTween better captures the semantic context of the text prompts and generates more natural motions that faithfully reflect the described actions. Additional quantitative and qualitative analyses, including inference time, are in the appendix.

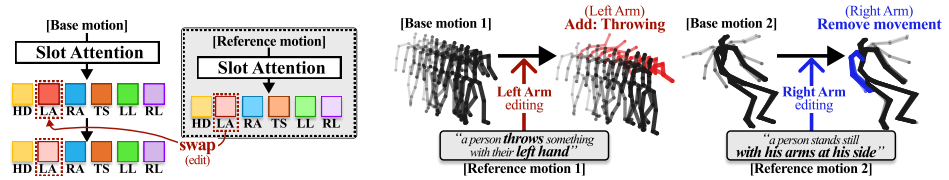
4.2.2 Effect of Hierarchical VQ-VAE.

To validate the effectiveness of the proposed Hierarchical VQ-VAE, we compare it with baselines: whole-body VQ-VAE (Guo et al., 2022b). Figure 6(a) shows that the proposed Hierarchical VQ-VAE uses 25% fewer parameters than the whole-body VQ-VAE (Guo et al., 2022b) by reducing the per-part latent dimensionality from 512 to 128, even though it employs six part-specific codebooks. Furthermore, Fig. 6(b) reports higher codebook perplexity, indicating that MoTween references a broader set of codebook entries and thus achieves more effective codebook utilization.

The t-SNE visualization in Fig. 6(c) indicates that the Hierarchical VQ-VAE learns a more semantically structured latent space. It produces clearly separable clusters across motion types, whereas the conventional VQ-VAE (Guo et al., 2022b) exhibits substantial overlap. Additional comparisons with a part-based VQ-VAE are provided in the appendix.

4.2.3 Slot-based Part-level Motion Editing.

Many models support motion editing (Kim et al., 2023; Zhang et al., 2023c), but diffusion-based methods (Tevet et al., 2023; Zhang et al., 2024) often re-synthesize non-conditioned regions, causing



(a) Motion editing process (b) Results of edited motions

Figure 7: **Part-level motion editing via slot swapping.** Our method edits a base motion by swapping its part slots with those from a reference motion that provides the desired motion characteristics. Darker poses indicate later timesteps, and colored joints denote the edited body parts.

Table 2: **Ablation studies on `HumanML3D` (Guo et al., 2022a) dataset.** The best and runner-up values are **bold** and underlined.

Methods	Top1 ↑	Top2 ↑	Top3 ↑	FID ↓	MM-Dist ↓	Diversity →
Ground Truth	0.511	0.703	0.797	0.002	2.974	9.503
w/o Self-Attention (in Event Predictor)	<u>0.526</u>	0.720	<u>0.814</u>	<u>0.036</u>	2.869	<u>9.562</u>
w/o Interpolation	0.523	0.713	0.798	0.042	2.886	9.564
w/o Part-wise Attention	0.520	0.717	0.803	0.046	<u>2.867</u>	9.415
w/o Temporal Attention	0.522	<u>0.721</u>	0.792	0.040	2.870	9.412
w/o In-Betweening Transformer	0.517	0.709	0.799	0.045	2.896	9.621
Ours with all methods	**0.537**	**0.730**	**0.818**	**0.031**	**2.861**	**9.542**

boundary inconsistencies. Masking-based methods (Guo et al., 2024; Pinyoanuntapong et al., 2024) mainly support temporal editing, and region-specific codebooks are limited to coarse body regions.

In contrast, our method obtains part-disentangled slot representations via Slot Attention (Locatello et al., 2020) (Fig. 2(b)). Since each body part is represented by an independent slot, part-level editing can be performed by swapping a target part slot with one from another motion while preserving the remaining context (Fig. 7(a)). A collection of reusable part slots further enables diverse edits through simple slot selection and swapping (Fig. 7(b)).

4.3 Ablation Studies

We ablate each component of MoTween on `HumanML3D` (Tab. 2); performance consistently improves as each module is added. Replacing keyframe interpolation with random masking prediction (w/o Interpolation) degrades performance, showing that keyframes provide useful structural anchors and better initialization. Using only linear interpolation without the In-Betweening Transformer also lowers performance, indicating that the transformer effectively refines in-between frames. Removing self-attention across body-part slots in the Event Predictor causes a further drop, showing that modeling inter-part correlations benefits coordinated event prediction, especially for compositional descriptions involving multiple body parts.

5 Conclusions and Future Works

In this paper, we proposed MoTween, a two-stage framework for text-to-3D human motion generation. MoTween discretizes motion with a Hierarchical VQ-VAE that reflects the human body hierarchy, and generates motion through keyframe-guided in-betweening focused on semantically important transitions. This design captures both part-level compositional structure and temporal dynamics.

Experiments on `HumanML3D` and `KIT-ML` show that MoTween achieves state-of-the-art quantitative and qualitative performance. Its part-disentangled slot representations also enable flexible part-level motion editing. However, the current rule-based event representation uses only seven velocity classes, which may not capture fine-grained or diagonal movements. Future work will explore finer-grained or multi-label event representations for richer motion dynamics.

AI USE STATEMENT

In this work, we used generative AI tools (large language models) solely for writing assistance. Specifically, we used them to improve the readability of the manuscript, condense text to meet the page limit, suggest a structure for the appendix, and adjust LaTeX formatting of tables and figures. We have not used generative AI tools to generate synthetic data, develop the theoretical framework, formulate or prove mathematical claims, design experiments, implement methods, clean or reformat datasets, analyze data, or interpret results. Translation and qualitative/thematic analysis are not applicable to this work. All AI-assisted edits were reviewed by the authors to ensure that the technical content, claims, and numerical results remained unchanged from the original author-written text. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We have made efforts to ensure that our results are reproducible. The Hierarchical VQ-VAE, including the body-part decomposition, hierarchical cross-attention, and part-wise codebook learning, is described in Sec. 3.1 and Eqs. 1–2, and the In-Betweening Motion Generation pipeline in Sec. 3.2 with Eqs. 3–5; the training objectives for both stages are given in Sec. 3.3 (Eqs. 6–7). Datasets, motion representations, data splits, evaluation protocol, and implementation details (latent dimension, text encoder, λ, optimizer, and hardware) are specified in Sec. 4.1. The ground-truth event labeling procedure is given as Algorithm 1 in Appendix B together with the label construction parameters τ, α, and β (Appendix B.2), and the exact joint partition, group definitions, and module-sharing details are provided in Appendix C. All experiments use the publicly available `HumanML3D` and `KIT-ML` datasets with the standard evaluation code of Guo et al. (2022a). Source code, pretrained checkpoints, the generated event labels, and configuration files will be released upon publication.

REFERENCES

Chaitanya Ahuja and Louis-Philippe Morency. Language2pose: Natural language grounded pose forecasting. In *Proceedings of the International Conference on 3D Vision (3DV)*, 2019.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024.

Bin Cao, Sipeng Zheng, Ye Wang, Lujie Xia, Qianshan Wei, Qin Jin, Jing Liu, and Zongqing Lu. Motionctrl: A real-time controllable vision-language-motion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Carnegie Mellon University. CMU mocap dataset.

Huiwen Chang, Han Zhang, Jarred Barber, Aaron Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Patrick Murphy, William T. Freeman, Michael Rubinstein, Yuanzhen Li, and Dilip Krishnan. Muse: Text-to-image generation via masked generative transformers. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2023.

Xiaodong Chen, Qian Bao, Xudong Liu, Jianping Fang, Jintao Fang, Yongdong Zhang, Tao Mei, and Wu Liu. Multi-level causal llm-based text-to-motion generation with human alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9342–9351, 2026.

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)*, 2023.

Seunggeun Chi, Hyung-gun Chi, Hengbo Ma, Nakul Agarwal, Faizan Siddiqui, Karthik Ramani, and Kwonjoon Lee. M2d2m: Multi-motion generation from text with discrete diffusion models. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024.

Rishabh Dabral, Muhammad Hamza Mughal, Vladislav Golyanik, and Christian Theobalt. Mofusion: A framework for denoising-diffusion-based motion synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Anindita Ghosh, Noshaba Cheema, Cennet Oguz, Christian Theobalt, and Philipp Slusallek. Synthesis of compositional animations from textual descriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2014.

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

Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022a.

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

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

Chuan Guo, Inwoo Hwang, Jian Wang, and Bing Zhou. Snapmogen: Human motion generation from expressive texts, 2025. URL `https://arxiv.org/abs/2507.09122`.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

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)*, pp. 7158–7168. IEEE, 2025.

Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. Motiongpt: Human motion as a foreign language. *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

Jihoon Kim, Jiseob Kim, and Sungjoon Choi. Flame: Free-form language-based motion synthesis & editing. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2023.

Hanyang Kong, Kehong Gong, Dongze Lian, Michael Bi Mi, and Xinchao Wang. Priority-centric human motion generation in discrete latent space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

Yunhong Lou, Linchao Zhu, Yaxiong Wang, Xiaohan Wang, and Yi Yang. Diversemotion: Towards diverse human motion generation via discrete diffusion. *arXiv preprint arXiv:2309.01372*, 2023.

Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

Christian Mandery, Ömer Terlemez, Martin Do, Nikolaus Vahrenkamp, and Tamim Asfour. The kit whole-body human motion database. In *International Conference on Advanced Robotics (ICAR)*, 2015.

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

Mathis Petrovich, Michael J Black, and Gül Varol. Temos: Generating diverse human motions from textual descriptions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

Ekkasit Pinyoanuntapong, Pu Wang, Minwoo Lee, and Chen Chen. Mmm: Generative masked motion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Matthias Plappert, Christian Mandery, and Tamim Asfour. The KIT motion-language dataset. *Big Data*, 4(4):236–252, 2016.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021.

Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2016.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

Guy Tevet, Brian Gordon, Amir Hertz, Amit H Bermano, and Daniel Cohen-Or. Motionclip: Exposing human motion generation to clip space. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-or, and Amit Haim Bermano. Human motion diffusion model. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157*, 2021.

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. *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

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*, volume 39, pp. 9797–9805, 2025.

Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017.

Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Shaoli Huang, Yong Zhang, Hongwei Zhao, Hongtao Lu, and Xi Shen. T2m-gpt: Generating human motion from textual descriptions with discrete representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023a.

Jianrong Zhang, Hehe Fan, and Yi Yang. Energymogen: Compositional human motion generation with energy-based diffusion model in latent space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

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)*, 2023b.

Mingyuan Zhang, Huirong Li, Zhongang Cai, Jiawei Ren, Lei Yang, and Ziwei Liu. Finemogen: Fine-grained spatio-temporal motion generation and editing. *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2023c.

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 (TPAMI)*, 46(6):4115–4128, 2024.

Chongyang Zhong, Lei Hu, Zihao Zhang, and Shihong Xia. Attt2m: Text-driven human motion generation with multi-perspective attention mechanism. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Qiran Zou, Shangyuan Yuan, Shian Du, Yu Wang, Chang Liu, Yi Xu, Jie Chen, and Xiangyang Ji. Parco: Part-coordinating text-to-motion synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024.

A Overview

The appendix is organized as follows:

- Section B: Event labeling algorithm and label construction parameters
- Section C: Body-part decomposition and architectural details
- Section D: Direct evaluation of the Text-to-Keyframe Predictor and error propagation analysis
- Section E: Isolated evaluation of the Hierarchical VQ-VAE
- Section F: Inter-part coherence analysis
- Section G: Additional qualitative analysis
- Section H: Comparison with recent methods and length-stratified evaluation
- Section I: Additional part-level motion editing results
- Section J: Inference time comparison
- Section K: Qualitative visualization of part-wise event labels
- Section L: Limitations

B Event Labeling Algorithm and Label Construction

B.1 Label Construction

`HumanML3D` and `KIT-ML` provide no manual event or keyframe annotations. We therefore construct part-wise event pseudo-labels from ground-truth motion to supervise the Text-to-Keyframe Predictor (Sec. 3.2), and define keyframes at event transitions. The full procedure is summarized in Algorithm 1.

For the torso, we reconstruct the global trajectory from cumulative root velocities, while all other parts use torso-relative joint positions to capture local motion independent of global displacement. Event labeling uses the root for the torso, the head for the head part, the hands for the arms, and the feet for the legs. For each part and frame, we apply Gaussian smoothing to the velocities to suppress jitter, and assign a directional label along the dominant axis when its magnitude exceeds a threshold τ; otherwise, the frame is labeled as Idle, yielding seven classes in total. To improve label quality, segments shorter than α frames are removed, and Idle gaps of length $\leq \beta$ between identical event segments are filled to ensure temporal continuity. Qualitative examples of the resulting label sequences are shown in Sec. K.

B.2 Label Parameters and Sensitivity

Table 3 lists the three parameters. They are not learned; they are used only for constructing training labels and are fixed for all samples. Thresholds vary only by action category and body part (action-aware policy): root thresholds are 0.02 and 0.03–0.05, and limb thresholds are 0.006 and 0.055, with α/β set to 2–6 and 2–4 frames, respectively. All settings, code, and labels will be released.

To assess sensitivity to label construction, we compare the action-aware policy with a single global setting shared across all action categories (Tab. 4). The global policy lowers keyframe matching F1 from 82.8% to 75.2% and worsens FID from 0.031 to 0.072, confirming that event construction quality affects downstream generation. This is not a complete sweep over τ, α, and β; a systematic one-at-a-time analysis is left for future work.

Table 3: **Event label construction parameters.**

Param.	Role	Setting
τ	Movement threshold	By part and action
α	Min. event duration	2–6 frames
β	Max. Idle gap	2–4 frames

Table 4: **Label construction policy.** KF F1 uses a ± 5-frame tolerance.

Policy	KF F1 ↑	FID ↓	Top-1 ↑
Global	75.2	0.072	0.520
Action-aware	**82.8**	**0.031**	**0.537**

Algorithm 1 Part-wise Motion Event Labeling

Require: Motion sequence $M = \{m_t\}_{t=1}^{T}$, body parts $\mathcal{P}$, threshold τ, min-segment length α, max-gap length β
Ensure: Frame-wise event labels $E^j \in \{0, \ldots, 6\}^T$ for each $j \in \mathcal{P}$

for each body part $j \in \mathcal{P}$ **do**
  if j is torso **then**
    Reconstruct absolute positions by accumulating root velocities
  else
    Extract torso-relative joint positions $\{p_t^j\}_{t=1}^{T}$
  end if
  Compute frame-wise velocities v_t^j and apply Gaussian smoothing
  for each frame $t = 1, \ldots, T$ **do**
    Find dominant axis $a^* = \arg\max_{a \in \{x,y,z\}} |v_{t,a}^j|$
    if $|v_{t,a^*}^j| > \tau$ **then**
      $E_t^j \leftarrow$ directional label $\in \{1, \ldots, 6\}$ based on $(a^*,\ \text{sign}(v_{t,a^*}^j))$
    else
      $E_t^j \leftarrow 0$ (Idle)
    end if
  end for
  Remove segments shorter than α frames (set to Idle)
  Fill Idle gaps of length $\leq \beta$ between same-label segments
end for
return $\{E^j\}_{j \in \mathcal{P}}$

C Body-Part Decomposition and Architectural Details

Joint partition. The six parts follow complete kinematic chains: head (HD) covers neck-to-head; left/right arms (LA/RA) cover shoulder-to-hand; torso (TS) covers root/pelvis-to-spine; and left/right legs (LL/RL) cover hip-to-foot. The three groups are whole body (all six parts), upper body (HD, LA, RA, TS), and lower body (TS, LL, RL). TS belongs to all three groups because it connects the upper and lower kinematic chains and therefore receives whole-, upper-, and lower-body context through Hierarchical Cross-Attention.

Stage 1. The Hierarchical VQ-VAE employs nine component-specific 1D convolutional encoders for the six parts and three groups. Although their input dimensions differ, all encoders map inputs into a shared 128-dimensional latent space. After Hierarchical Cross-Attention injects group-level context into the part representations, the six refined part-level latents are quantized using six part-specific codebooks and jointly reconstructed into full-body motion by a single decoder.

Stage 2. Stage 2 is not six independent pipelines. Slot Attention decomposes the shared CLIP text embedding into six part-aligned slots with fixed identities (HD, LA, RA, TS, LL, RL) throughout training and inference; each slot is routed to its corresponding part-specific event head and supervised by that part's pseudo event sequence, without sample-wise reassignment or Hungarian matching. This fixed routing breaks permutation symmetry and maintains distinct left/right identities, while cross-slot self-attention exchanges context without changing slot order. The Keyframe Transformer generates sparse part-specific anchors from the corresponding slots, and a single shared In-Betweening Transformer jointly processes all six sequences using temporal and then part-wise attention. Part-specific heads predict tokens for the corresponding codebooks, and the frozen Stage-1 decoder reconstructs the final motion. Inter-part coordination is thus modeled through Hierarchical Cross-Attention in Stage 1 and part-wise attention in Stage 2.

D Direct Evaluation of the Text-to-Keyframe Predictor

D.1 Event and Keyframe Accuracy

The Text-to-Keyframe Predictor predicts a frame-wise event class for each body part; a keyframe is derived at the time step where two consecutive predicted event classes differ. On the `HumanML3D` test set, the predictor achieves 95.8% frame-wise event accuracy. Since keyframes are defined by event transitions, we additionally evaluate whether these transition points are correctly localized (Tab. 5). A predicted keyframe is considered correct when it lies within ± 5 frames of a ground-truth keyframe, and the keyframe-count error is the average absolute difference between the predicted and ground-truth numbers of keyframes per part and motion. Part-wise keyframe accuracies (Tab. 6) are computed from the transitions of each slot's predicted event sequence; together with the fixed slot-to-part routing, they indicate that each slot captures the transition patterns of its assigned body part, and separate LA/RA supervision maintains consistent left-right identities.

Table 5: **Keyframe localization accuracy.**

Metric	Result
Frame-wise event accuracy ↑	95.8%
Matched keyframes (F1, $\pm$5f) ↑	82.8%
Avg. keyframe-count error ↓	3.48

Table 6: **Part-wise keyframe accuracy (%).**

Part	HD	LA	RA	TS	LL	RL
KF Acc. ↑	81.8	76.8	65.0	66.2	69.7	69.5

D.2 Robustness to Event Prediction Errors and Oracle Keyframes

To quantify how upstream event prediction errors propagate to generation, we evaluate three predictor checkpoints with different frame-wise event accuracies while keeping the Keyframe-Guided Motion Transformer and all other settings fixed (Tab. 7). Generation quality consistently degrades as event accuracy decreases, since inaccurate events yield less reliable keyframe locations. However, the degradation is progressive rather than abrupt: even at 85.0% event accuracy, the model retains Top-1/2/3 R-Precision of 0.518/0.717/0.808, showing that moderate upstream errors do not cause catastrophic failure.

To isolate the effect of errors in predicted keyframe positions, we further replace the predicted event transitions with transitions derived from ground-truth event sequences, retaining the same Keyframe Transformer and In-Betweening Transformer (Tab. 8). Ground-truth transitions provide only modest improvements (FID 0.031 $\rightarrow$ 0.030), indicating that transition-location errors account for a limited portion of the remaining generation error.

Table 7: **Effect of event prediction accuracy.** All other components are fixed.

Event acc. ↑	FID ↓	Top-1 ↑	Top-2 ↑	Top-3 ↑
85.0%	0.088	0.518	0.717	0.808
90.0%	0.056	0.526	0.721	0.810
95.8%	**0.031**	**0.537**	**0.730**	**0.818**

Table 8: **Oracle keyframe-position analysis.**

KF-position source	FID ↓	Top-1 ↑	Top-2 ↑	Top-3 ↑
Predicted	0.031	0.537	0.730	0.818
Ground-truth	**0.030**	**0.542**	**0.741**	**0.827**

D.3 Keyframe-First vs. Full-Sequence Autoregressive Generation

The Keyframe Transformer is autoregressive, but it predicts only a sparse set of part-wise keyframe representations, while the remaining frames are generated jointly by the In-Betweening Transformer; MoTween thus reduces, rather than eliminates, the autoregressive horizon. To directly measure this benefit, we construct a controlled variant that replaces the Keyframe Transformer with a full-sequence autoregressive Transformer generating representations for all frames, while retaining the same Hierarchical VQ-VAE, text conditioning, and training protocol (Tab. 9). Full-sequence autoregression substantially degrades both FID and R-Precision, demonstrating the benefit of predicting sparse keyframes first and completing the remaining frames through in-betweening.

Table 9: **Keyframe-first vs. full-sequence autoregressive generation.** Stage-1 representation and training settings are identical.

Stage-2 generation strategy	FID ↓	Top-1 ↑	Top-2 ↑	Top-3 ↑
Full-sequence autoregressive	0.069	0.521	0.644	0.798
Keyframe-first (MoTween)	**0.031**	**0.537**	**0.730**	**0.818**

E Isolated Evaluation of the Hierarchical VQ-VAE

We evaluate the Hierarchical VQ-VAE independently against a whole-body tokenizer, a joint-level tokenizer, and a flat part-wise tokenizer (ParCo-style (Zou et al., 2024)) on `HumanML3D` (Tab. 10). The Hierarchical VQ-VAE achieves the best reconstruction FID, MPJPE, and codebook utilization, as well as the highest R-Precision. The whole-body tokenizer provides a coarse representation in which subtle part motions are dominated by larger movements, while the joint-level tokenizer preserves local detail but excessively separates strongly related joints, making coordinated token generation harder. The proposed 6-part/3-group design preserves part-specific detail while incorporating whole-, upper-, and lower-body context before quantization. We do not claim this decomposition is universally optimal; it performs best among the evaluated configurations. Fig. 6 of the main paper further compares parameter count, utilization, and latent structure against the whole-body VQ-VAE, and Fig. 8(b) compares against ParCo.

Table 10: **Stage-1 tokenizer comparison on `HumanML3D`.** Util. denotes codebook utilization.

Tokenizer	FID ↓	MPJPE ↓	Util. (%) ↑	Top-1 ↑	Top-2 ↑	Top-3 ↑
Whole-body VQ-VAE	0.0704	15.90	32.4	0.501	0.696	0.782
Joint-level VQ-VAE	0.0640	–	–	0.520	0.712	0.804
Flat part-wise VQ-VAE	0.0238	13.09	41.1	0.536	0.720	0.814
Hierarchical VQ-VAE	**0.0180**	**12.09**	**42.9**	**0.541**	**0.737**	**0.826**

F Inter-Part Coherence Analysis

To directly evaluate inter-part coordination, we report the Inter-part Velocity Correlation MAE, which measures the difference between generated and ground-truth velocity correlations over the 15 body-part pairs (Tab. 11). Removing Hierarchical Cross-Attention (HCA) or part-wise attention increases the error and worsens FID, showing that both hierarchical context in Stage 1 and part-wise attention in Stage 2 improve inter-part motion coherence. Together with Tab. 2 of the main paper, where removing part-wise attention, temporal attention, or the entire In-Betweening Transformer yields FID 0.046, 0.040, and 0.045 versus 0.031 for the full model, these results isolate the contribution of each coordination mechanism.

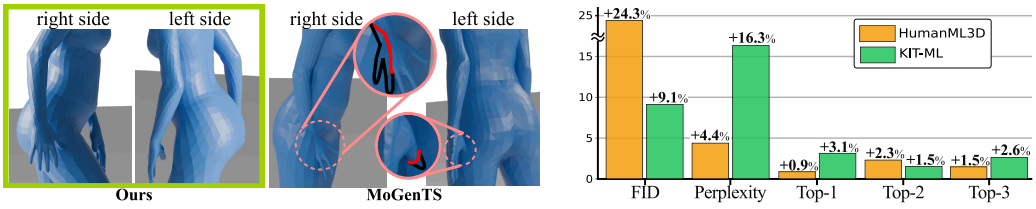
Table 11: **Inter-part velocity correlation MAE on `HumanML3D`.** Lower is better.

Method	Velocity Corr. MAE ↓	FID ↓
w/o HCA	0.2539 ± 0.2907	0.064
w/o Part-wise Attn.	0.2478 ± 0.1890	0.046
MoTween	$\mathbf{0.1818 \pm 0.1306}$	**0.031**

G Additional Qualitative Analysis

A closer comparison in Fig. 8(a) shows that MoTween produces physically plausible poses without the body-part penetration artifacts observed in MoGenTS (Yuan et al., 2024), owing to the proposed hierarchical body-part encoding. In addition, as shown in Fig. 8(b), the Hierarchical VQ-VAE consistently outperforms the part-based VQ-VAE ParCo (Zou et al., 2024) in Stage 1, indicating that

explicitly modeling group-level context via hierarchical cross-attention improves coordinated motion representations beyond part-level modeling alone.



(a) Close-up comparison (b) Performance gap with ParCo

Figure 8: (a) Close-up comparison with MoGenTS (Yuan et al., 2024). Red lines highlight body-part penetration artifacts. (b) Performance gap between the proposed Hierarchical VQ-VAE and the part-based VQ-VAE method ParCo (Zou et al., 2024), where positive values indicate improvements over ParCo.

As shown in Fig. 9, linear interpolation alone produces irregular and largely independent trajectories of the arm embeddings in the latent space (Fig. 9(a)). In contrast, the In-Betweening Transformer produces bilaterally symmetric and naturally synchronized trajectories (Fig. 9(b)), suggesting that it models coordinated inter-part dynamics (e.g., alternating arm swings during walking).

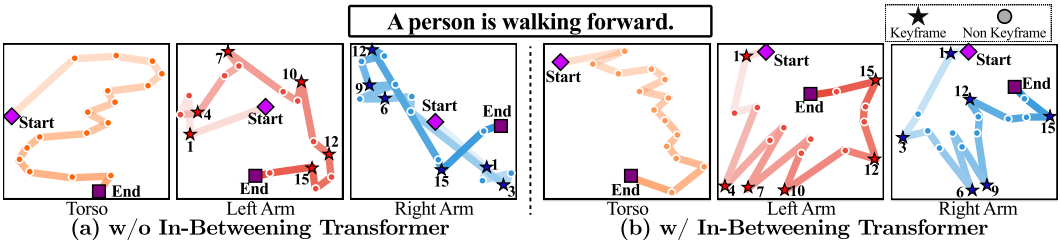


Figure 9: **Trajectory visualization of the torso, left arm, and right arm in the embedding space.** (a) without the In-Betweening Transformer and (b) with it. Color intensity increases over time, and numbered star markers denote the keyframes.

H Comparison with Recent Methods and Length-Stratified Evaluation

H.1 Recent Text-to-Motion Methods

Table 12 extends the HumanML3D comparison to recent methods: Light-T2M (Zeng et al., 2025), EnergyMoGen (Zhang et al., 2025), SALAD (Hong et al., 2025), and MoMask++ (Guo et al., 2025). We also evaluated the released SALAD implementation under the same evaluation pipeline as MoTween; since the reproduced result differs from the published one, we report both for transparency. Except for SALAD (reproduced), values are taken from the respective papers under the standard protocol. The published SALAD result achieves stronger R-Precision, whereas MoTween achieves the lowest FID among the compared methods; under our reproduction, MoTween outperforms SALAD on all four metrics.

Table 12: **Comparison with recent methods on `HumanML3D`.** Best in bold, second-best underlined.

Method	FID ↓	Top-1 ↑	Top-2 ↑	Top-3 ↑
MoGenTS (Yuan et al., 2024)	<u>0.033</u>	0.529	0.719	0.812
Light-T2M (Zeng et al., 2025)	0.040	0.511	0.699	0.795
EnergyMoGen (Zhang et al., 2025)	0.176	0.526	0.718	0.815
SALAD (reported) (Hong et al., 2025)	0.076	**0.581**	**0.769**	**0.857**
SALAD (reproduced)	0.088	0.525	0.725	0.809
MoMask++ (Guo et al., 2025)	0.072	0.528	0.718	0.811
MoTween (Ours)	**0.031**	<u>0.537</u>	<u>0.730</u>	<u>0.818</u>

H.2 Relation to Motion In-Betweening Methods

Dedicated motion in-betweening methods receive user-provided poses or keyframes as input, whereas MoTween receives only a text description and must predict both the part-wise keyframe positions and their representations; their end-to-end results are therefore not directly comparable without a matched-input protocol. Among text-to-motion methods under the same protocol, MMM (Pinyoanuntapong et al., 2024) is the closest masked-completion baseline. As reported in Tab. 1 of the main paper, MMM obtains FID 0.080 and Top-1/2/3 of 0.504/0.696/0.794 on `HumanML3D`, whereas MoTween achieves 0.031 and 0.537/0.730/0.818.

H.3 Length-Stratified Evaluation

To examine performance across motion durations, we divide the `HumanML3D` test set into short, medium, and long groups and evaluate all compared methods using their released implementations under the same pipeline and samples (Tab. 13). MoTween achieves the lowest FID and highest Top-1 in every group, including the long-motion subset, showing that performance does not degrade disproportionately on longer sequences.

Table 13: **Length-stratified evaluation on `HumanML3D`.**

Method	FID ↓			Top-1 ↑		
	Short	Medium	Long	Short	Medium	Long
MMM (Pinyoanuntapong et al., 2024)	0.1651	0.1237	0.1312	0.4508	0.4987	0.4799
MoMask (Guo et al., 2024)	0.1527	0.1134	0.1261	0.4574	0.5028	0.4822
MoGenTS (Yuan et al., 2024)	0.1433	0.0912	0.1166	0.4663	0.5156	0.4916
MoTween	**0.1237**	**0.0741**	**0.0884**	**0.4788**	**0.5402**	**0.5251**

I Additional Part-Level Motion Editing Results

Part-level editing is not the primary objective of MoTween, but an additional capability enabled by its part-disentangled slots: one or multiple body-part slots can be replaced while the remaining slots are preserved, without retraining or modifying the generation pipeline. Figure 10 presents additional results demonstrating two extended capabilities of the slot-swapping mechanism. First, simultaneous multi-part editing is supported by jointly swapping multiple target slots from a single reference motion. Second, edits can be composed from multiple reference motions, where motion characteristics for different parts are independently sourced from distinct references and applied to the base motion at once. These capabilities arise naturally from the part-disentangled slot representations, where each slot can be independently selected and swapped while preserving the motion context of the remaining body parts. As we do not provide a dedicated quantitative editing benchmark, these results are qualitative.

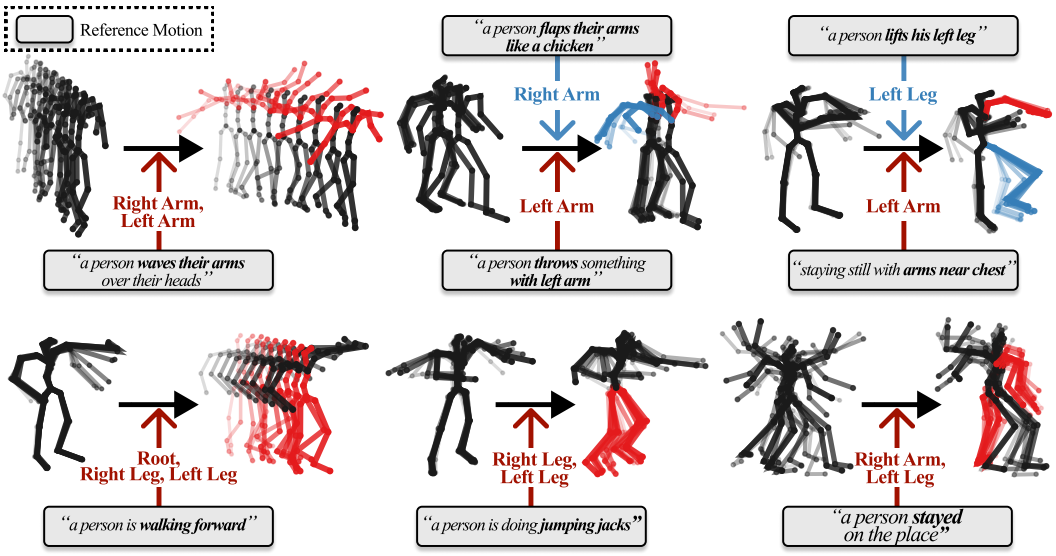


Figure 10: **Part-level motion editing via slot swapping.** Darker poses indicate later timesteps, and colored joints denote the edited body parts.

J Inference Time Comparison

Table 14 compares the average inference time per motion. MoTween achieves 0.240s, roughly 2–3× faster than T2M-GPT, MoGenTS, and AttT2M, and about 90× faster than the diffusion-based MotionDiffuse. It remains competitive with efficient recent baselines such as Light-T2M and MoMask while maintaining strong generation performance and part-level controllability. This efficiency stems from generating a small number of keyframes first and interpolating between them, rather than generating all frames sequentially.

Table 14: **Inference time comparison.** Average time per motion in seconds; relative time is with respect to MoTween.

Method	Time (s) ↓	Relative
MotionDiffuse (Zhang et al., 2024)	22.255	92.73×
MoGenTS (Yuan et al., 2024)	0.800	3.33×
T2M-GPT (Zhang et al., 2023a)	0.773	3.22×
AttT2M (Zhong et al., 2023)	0.490	2.04×
Light-T2M (Zeng et al., 2025)	**0.151**	**0.63**×
MoMask (Guo et al., 2024)	0.181	0.75×
MoTween (Ours)	0.240	1.00×

K Qualitative Visualization of Part-Wise Event Labels

Figures 11 and 12 visualize part-wise event label sequences produced by Algorithm 1 on `HumanML3D`. Each row shows the frame-wise event class assigned to a body part over time, and keyframes are located at the transitions between consecutive events. The examples illustrate that the labels capture the dominant direction of each part's motion and that keyframes fall at semantically meaningful transition points, such as the onset and termination of arm or leg movements.

L Limitations

While MoTween achieves state-of-the-art performance, it has several limitations. First, the motion event representation relies on rule-based velocity classification with seven discrete classes. These

labels serve only as auxiliary supervision for detecting dominant-direction transitions and do not constrain the final generated motion to axis-aligned trajectories; the qualitative results in Fig. 5 of the main paper include a zig-zag trajectory and a gradual arm-lowering motion while descending stairs. Nevertheless, dominant-axis labeling may overlook subtle transitions involving simultaneous changes along multiple axes, and keyframe placement may be suboptimal for motions with gradual transitions that lack a clear dominant axis. Finer-grained or multi-label event representations are a promising direction. Second, since generation is conditioned on predicted keyframes, inaccurate event or keyframe predictions can degrade generated motion, particularly for complex compositional descriptions; Sec. D shows that this degradation is progressive rather than catastrophic. Third, the label construction parameters were selected empirically without a complete sensitivity sweep, and our validation is limited to `HumanML3D` and `KIT-ML`. Extending MoTween to larger datasets such as Motion-X or to humanoid-oriented benchmarks would require adapting the body-part decomposition and retraining both stages, which we leave for future work.

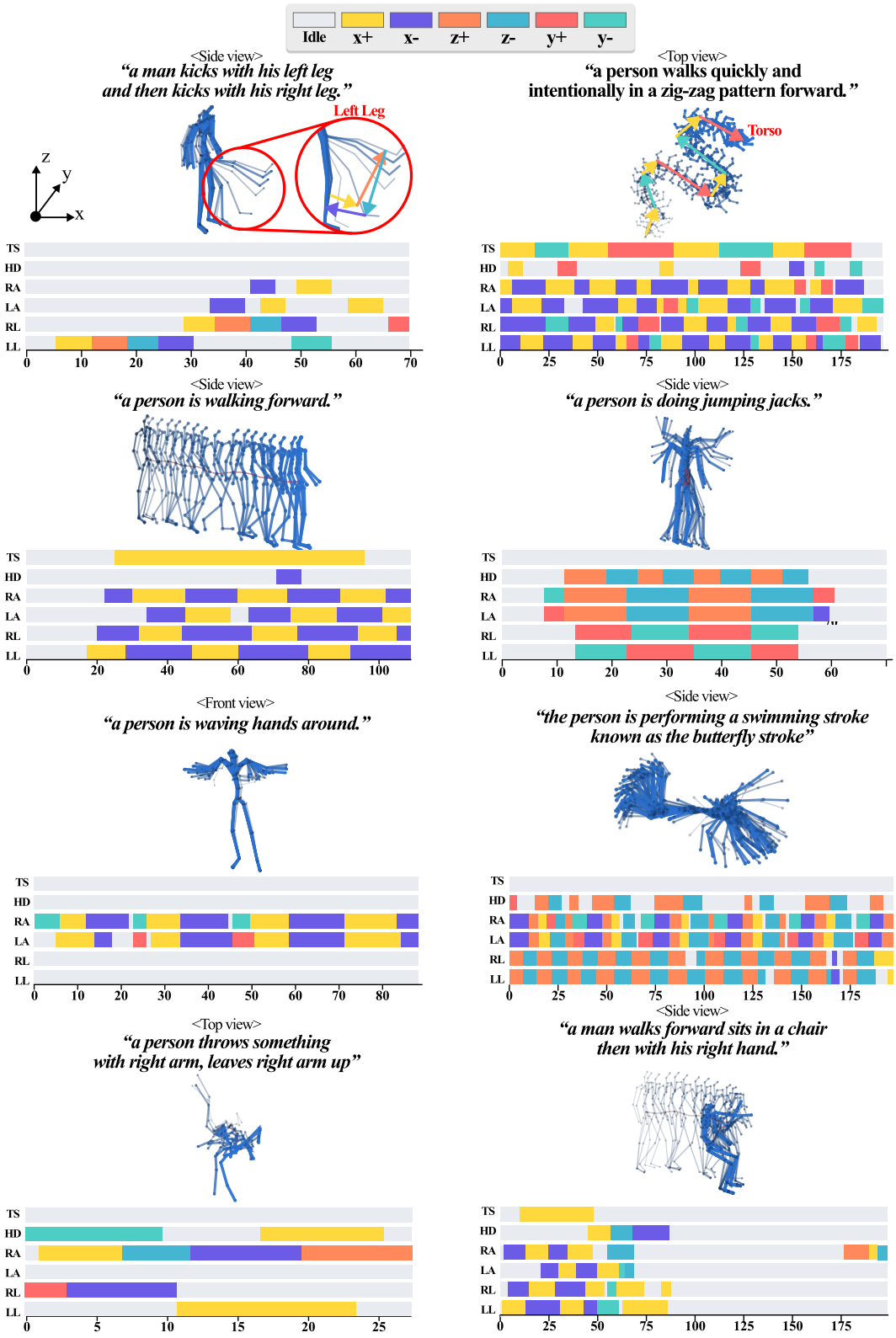


Figure 11: **Visualization of part-wise motion event labels.** Darker poses indicate later timesteps.

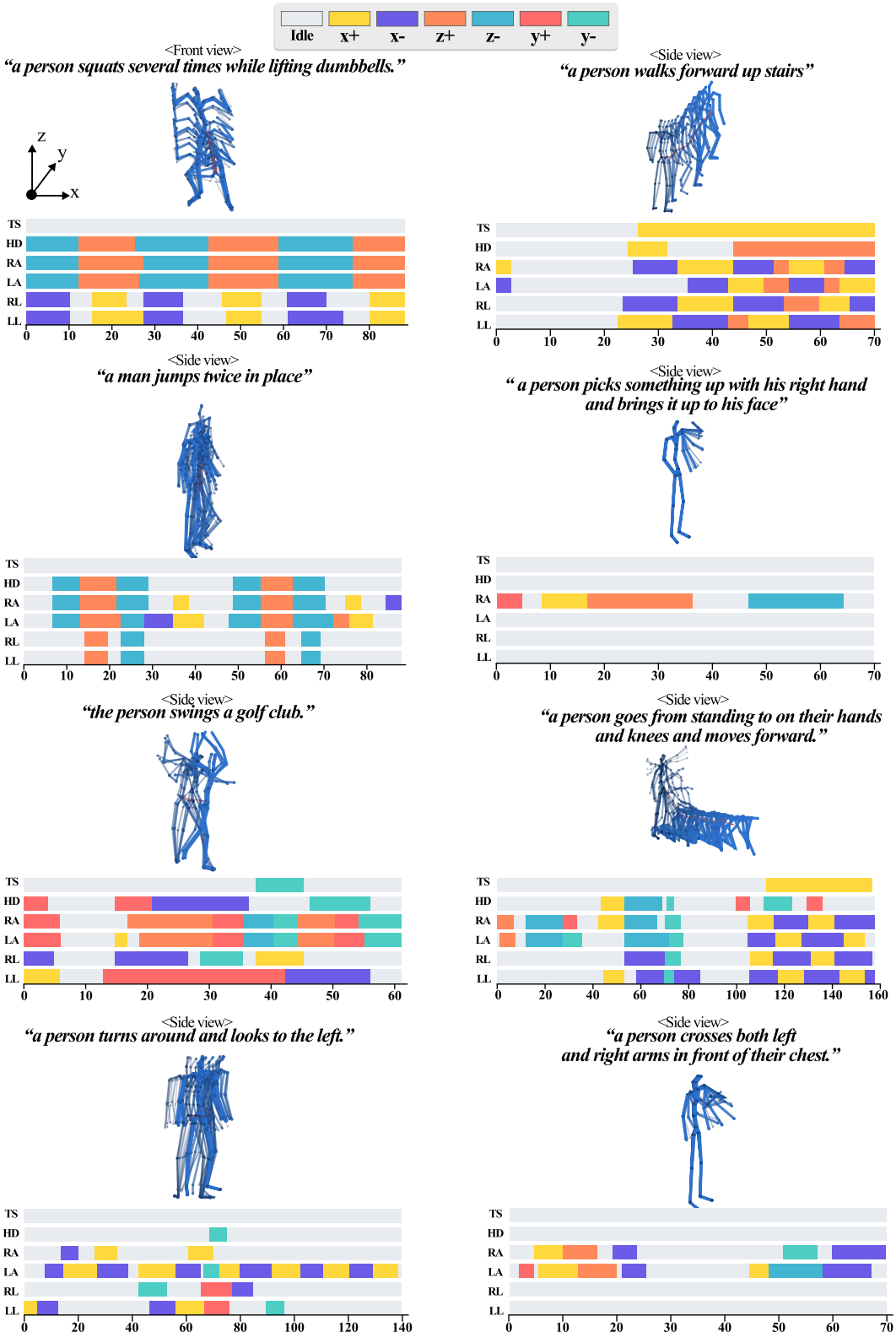


Figure 12: **Visualization of part-wise motion event labels.** Darker poses indicate later timesteps.