# ActionPlan: Future-Aware Streaming Motion Synthesis via Frame-Level Action Planning

Eric Nazarenus*[1], Chuqiao Li*†[1], Yannan He[1], Xianghui Xie[1,2], Jan Eric Lenssen[2], and Gerard Pons-Moll[1,2]

[1] Tübingen AI Center, University of Tübingen, Germany
[2] Max Planck Institute for Informatics, Saarland Informatics Campus, Germany
https://coral79.github.io/ActionPlan/

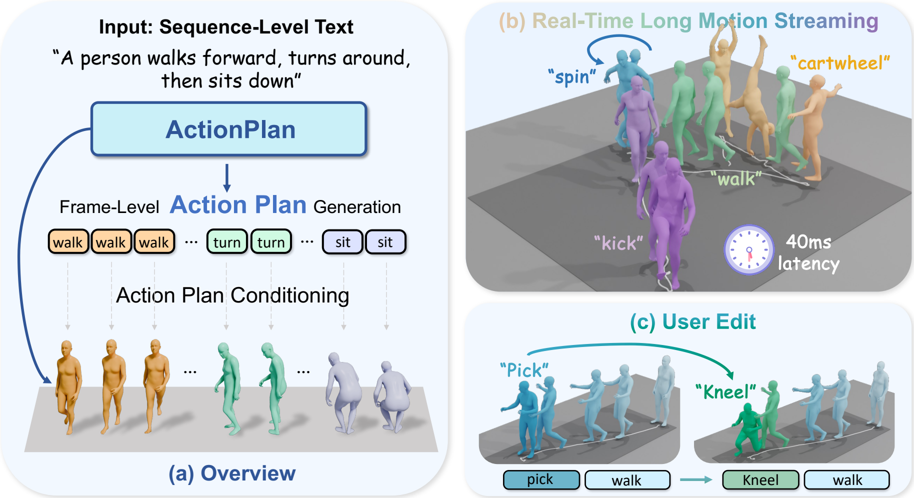


**Fig. 1:** ActionPlan decouples high-level action planning from low-level motion generation in a single generative model (a). By conditioning motion synthesis on generated action plans, ActionPlan achieves online generation (b) without the typical accuracy drop that happens in existing streaming methods and supports localized edits (c).

**Abstract.** We present **ActionPlan**, a unified motion diffusion framework that bridges real-time streaming with high-quality offline generation within a single model. The core idea is to introduce a per-frame action plan: the model predicts frame-level text latents that act as dense semantic anchors throughout denoising, and uses them to denoise the full motion sequence with combined semantic and motion cues. To support this structured workflow, we design latent-specific diffusion steps, allowing each motion latent to be denoised independently and sampled in flexible orders at inference. As a result, ActionPlan can run in a history-conditioned, future-aware mode for real-time streaming, while also supporting high-quality offline generation. The same mechanism further enables zero-shot motion editing and in-betweening without additional models. Experiments demonstrate that our real-time streaming is 5.25× faster while also achieving 18% motion quality improvement over the best previous method in terms of FID.

* Equal contribution. † Corresponding author

## 1 Introduction

Text-driven human motion generation [21,54,63,64] has emerged as a key technology for enabling realistic interactions in digital humans, virtual reality, and autonomous agents. An ideal motion synthesis framework should simultaneously satisfy three core requirements: high *semantic fidelity* to complex textual prompts, *on-the-fly streaming* for low-latency interaction, and sufficient *versatility* to support different tasks such as zero-shot editing and in-betweening.

Existing methods largely fall into two specialized and mutually exclusive paradigms. Offline frameworks [4, 11, 54] achieve high-quality results by leveraging global bidirectional attention to condition on the future with parallel or random order generation (c.f. Fig. 2). However, their non-causal design requires access to the entire sequence, making them unsuitable for real-time streaming. In contrast, recent streaming-compatible models [60] achieve low-latency generation by adopting causal latent spaces and unidirectional raster order inference. This strict causality, however, imposes a form of temporal myopia: without access to future context, streaming models often miss semantics from complex prompts, leading to less consistent motion generation. Furthermore, they are inherently incapable of bidirectional tasks such as editing and in-betweening.

In this work, we aim to preserve the low-latency benefits of streaming motion generation while equipping the model with awareness of global and future motion. We argue that effective online motion synthesis requires anticipating upcoming actions rather than reacting solely to past context.

To this end, we propose **ActionPlan** (c.f. Fig. 1), a hierarchical framework that first predicts an action plan, a sequence of fine-grained, frame-level textual action descriptors, which subsequently conditions motion generation. By *decoupling high-level semantic planning from low-level kinematic synthesis*, our approach enables state-of-the art generation quality and *future-aware streaming without sacrificing accuracy.* w Action plans are temporally aligned with motion at each frame and their generation is trained jointly with motion latent generation using latent-specific diffusion timesteps for text and individual motion frames [7,59]. These independent timesteps enable flexible denoising schedules at inference time. We first generate action plans and then denoise the motion using task-specific timestep schedules tailored to online streaming, motion editing, or in-betweening. As illustrated in Fig. 2, this flexible scheduling, guided by action plans, allows offline and online generation by choosing different overlapping denoising orders. We evaluate ActionPlan on HumanML3D-272 [21,60]. In offline mode, ActionPlan improves on the best offline competing method [41] by 22% in FID. In online streaming mode, ActionPlan surpasses even the best offline baseline [41] by 18% and the best streaming method [60] by 51% in FID, while achieving up to 5.25× faster token generation.

We also conduct comprehensive ablations which validate that our action plan generation consistently improves performance in both setups and that our single model supports editing and in-betweening. Our code and pretrained model will be made publicly available.

Our contributions are summarized as follows:

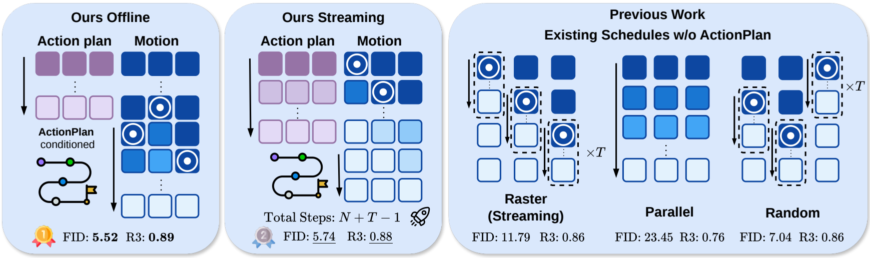


**Fig. 2: Comparison of generation paradigms**, where darker shading indicates higher noise levels. By introducing frame-level action plans as semantic conditioning, ActionPlan achieves significantly better FID and R-Precision compared to schedules without ActionPlan. Additionally, our streaming mode completes generation in only $N + T - 1$ total steps ($N$: motion tokens, $T$: flow matching steps), enabling efficient low-latency generation without sacrificing motion quality.

- We introduce ActionPlan, a hierarchical diffusion framework that decouples high-level action planning from low-level motion generation, by introducing varying noise levels for individual text and motion frames.
- We propose to generate frame-level textual action plans first to strengthen the semantic conditioning for subsequent motion generation. Comprehensive ablations validate the effectiveness of our action plan.
- We demonstrate that our action plan–guided sampling strategy supports both, offline and streaming generation, editing, and in-betweening within the same model without fine-tuning. Experiments show that our method significantly outperforms previous methods in both motion generation quality and inference speed.

## 2 Related Work

**Text-to-motion generation.** Text-conditioned motion generation aims to synthesize 3D human motions *conditioned on* natural language descriptions [21]. Early works learn cross-modal mappings by aligning text and motion in latent spaces [1,42,43,52], or by adopting diffusion-based models [11,13,29–31,50,54, 62,64,65,67] Following works [2,9,10,27,33,44,51,53,66,69–72,75] mainly *focus* on improving architectures or training strategies. Another line of motion generation methods utilizes Vector Quantized Variational Autoencoders (VQ-VAE) to discretize motions into tokens [8,15,18,19,22,26,28,34,37,38,45,57, 58,63,68,77], these methods mainly apply BERT-style bidirectional transformers [12,16,20,46,47] to predict the codebook entries via cross-entropy supervision, treating motions as a foreign language. More recent works [24,40,41,55,60,78] utilize continuous-valued motion latents to generate higher-quality human motions, these approaches still employ bidirectional attention mechanisms to denoise motion latents under textual guidance. While such offline designs enable high-quality generation and support flexible tasks such as motion editing and in-betweening [14,20,30,41,45,47,54], they fundamentally rely on non-causal inference and assume access to the complete motion sequence. Consequently, these methods are inherently unsuitable for low-latency interactive applications that require incremental, streaming capability.

**Real-time and online motion synthesis.** Interactive applications require motion to be generated incrementally under streaming text input [51]. Recent works apply diffusion models in an auto-regressive manner for real-time interactive character control, such as CAMDM [9] and A-MDM [51]. Ready-to-React [6] further extends this idea to online two-character interaction. In addition, CLoSD [53] and DART [72] explore real-time text-driven motion control with streaming prompts. However, these approaches are often not strictly causal because they rely on a fixed-length context window. As generation progresses over time, access to earlier history diminishes and semantic coherence gradually fades. More recently, MotionStreamer [60] proposes streaming text-to-motion generation with a continuous *causal* latent space and auto-regressive denoising, enabling online response with variable-length historical context. In contrast, we study a streaming-compatible formulation that additionally supports future-conditioned, bidirectional operations (e.g., editing and in-betweening) within a unified framework with action plans that clearly improve generation quality.

**Structured generation with diffusion.** In image generation, early representative methods [49] perform diffusion in a learned latent space. Recent approaches encode an image into a sequence of latents, enabling structured generation and explicitly exposing token ordering or masking as a design choice [17,32]. Following this line, motion diffusion models adopt learned motion latents with masked auto-regressive generation to improve realism and text–motion alignment [24,41]. For signals with strong spatial dependencies, SRMs [59] predict an uncertainty-driven adaptive generation order, yielding consistent gains in both fidelity and correctness. More recently, Latent Forcing [3] couples latent denoising with pixel-level denoising via separate noise schedules, producing an informative early "blueprint" and implicitly learning a underlying generation order for synthesis. A similar principle appears in motion generation: AutoKeyframe [73] generates keyframes first and then fills in the remaining frames. Diffusion Forcing [7] further generalizes structured sampling by assigning frame-specific noise levels, which facilitates flexible causal generation. Subsequent works leverage frame- or latent-wise timestep schedules together with tailored sampling strategies to enhance quality [5,61]. Motivated by these advances, we are the first method to generate textual action plans as a guiding blueprint and then perform structured real-time motion generation, achieving higher speed and quality while remaining plug-and-play for downstream applications.

## 3 ActionPlan: A Framework for Diverse Motion Tasks

We propose ActionPlan, a unified text-to-motion (T2M) generation framework based on latent diffusion that supports versatile tasks via latent-specific diffusion timesteps. An overview of the proposed framework is shown in Fig. 3. Our key

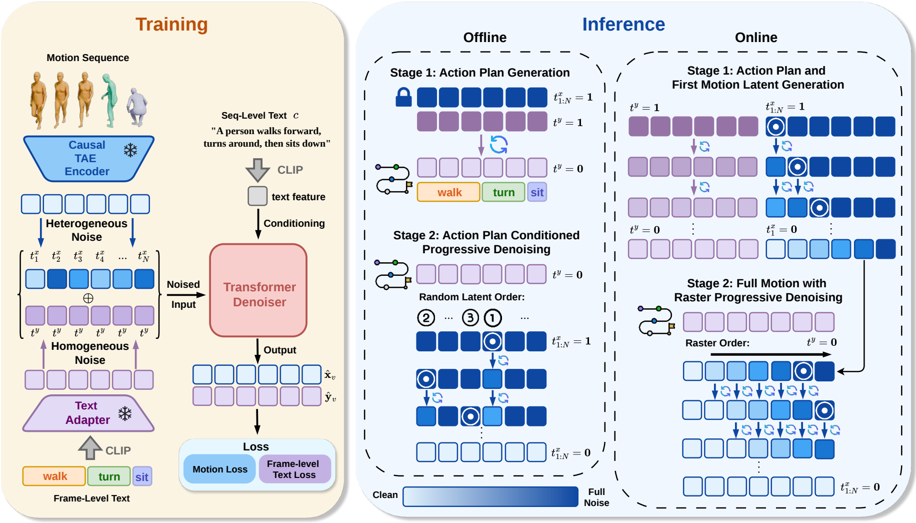


**Fig. 3: Overview of our ActionPlan**. (a) During training, motion latents are noised with per-frame heterogeneous timesteps while frame-level text latents share a single global timestep. A Transformer Denoiser is trained to jointly reconstruct both. During inference, the model operates in two modes: in offline mode (b), the action plan is fully generated first and then motion latents are denoised in random pyramid order; in streaming mode (c), the action plan is denoised alongside the first motion frame, followed by raster progressive denoising of the remaining latents.

idea is to predict textual action plans that are aligned with motion at each frame and to allow independent denoising of each motion latent. At inference time, we first generate the per-frame action plan, then denoise the motion latents using task-specific schedules tailored to different downstream applications. We first introduce the problem setup and motion autoencoder in Sec. 3.1 and then describe co-generation of motion and text in Sec. 3.2. We further discuss our latent-specific noise scheduling in Sec. 3.3 which allows flexible sampling guided by action plan generation in Sec. 3.4.

## 3.1 Preliminaries

**Pose representation.** We represent each pose $m \in \mathbb{R}^{272}$ using SMPL-based 6D rotations [60, 76] as:

$$m = \{\dot{r}^x, \dot{r}^z, \dot{r}^a, j^p, j^v, j^r\}, \tag{1}$$

where $(\dot{r}^x, \dot{r}^z) \in \mathbb{R}^2$ are root linear velocities on the ground plane, $\dot{r}^a \in \mathbb{R}^6$ is the root angular velocity in 6D rotation, $j^p \in \mathbb{R}^{3K}$, $j^v \in \mathbb{R}^{3K}$, and $j^r \in \mathbb{R}^{6K}$ are local joint positions, velocities, and rotations respectively, with $K = 22$ joints. Unlike the widely used 263D representation [21], which requires inverse kinematics to recover SMPL [36] parameters, the 272D representation directly drives the SMPL model without post-processing, avoiding conversion-induced rotation errors. We use this representation in all experiments.

**Motion autoencoder.** We adopt the Causal Temporal AutoEncoder (Causal TAE) [60] to encode raw motion sequences into a continuous latent space. Given a motion sequence $M = \{m_1, \ldots, m_N\}$ with $m_t \in \mathbb{R}^D$ ($D = 272$), the encoder $\mathcal{E}$ produces a sequence of latents $\mathbf{x} = \{x_1, \ldots, x_{N/l}\}$ with $x_i \in \mathbb{R}^{d_c}$, where $l$ is the temporal downsampling rate and $d_c$ is the latent dimension. Both the encoder and decoder are built from 1D causal convolutions, ensuring that each latent depends only on current and past frames. This causal property enables online decoding: motion frames can be reconstructed sequentially as latents become available, without waiting for the full sequence. The Causal TAE is trained using standard VAE loss with an additional root stability loss, for more details we refer to the original work [60]. We keep the the TAE frozen for later training.

### 3.2 Joint Diffusion on Action Plans and Kinematic Motion

We generate motion from action plans using diffusion in the Causal TAE latent space. Prior methods rely on a single sequence-level text description, offering only coarse conditioning and failing to capture the temporal structure of multi-action sequences (e.g., “walk forward then sit down”). Since both global intent and per-frame actions matter, we advocate for a hierarchical text representation by additionally predicting action plans aligned with motion at every frame. Thus, conditioned on a sequence-level text description $c$ encoded by CLIP, our diffusion model $G_\theta$ learns to generate a sequence of action plans $\{y_1, ..., y_N\}$ with temporally aligned motion latent codes $\{x_1, ..., x_N\}$.

**Action and motion representations.** Action plans $\{y_1, ..., y_N\}$ are obtained by encoding per-frame descriptions with CLIP and compressing them via a pre-trained MLP autoencoder to a 16-D space matching the motion latent dimension. Unlike the Causal TAE introduced in Sec. 3.1, the action autoencoder operates independently per latent, preserving fine-grained temporal alignment. Training combines three objectives: an MSE reconstruction loss in the original embedding space, a neighbor-preservation loss that encourages correct action label retrieval, and a variance regularization term that discourages dimensional collapse by pushing each latent dimension toward unit variance. More details are provided in supplementary. Inspired by UniMotion [31], the text latents are downsampled by a factor of 4 in the temporal axis and we align frame-level text with motion by concatenating the latents at each frame as one vector $(x_i, y_i)$.

**Text and motion generation.** Our denoiser $G_\theta$, a Transformer network, takes a sequence of concatenated vectors $\{(x_i, y_i)\}_1^N$ together with diffusion timesteps as input and predicts the velocity of the denoising trajectory: $\hat{\mathbf{x}}_v, \hat{\mathbf{y}}_v$. We train our diffusion network with carefully designed noise scheduling to allow flexible sampling at test time, which is introduced in the following section.

### 3.3 Training with Latent-specific Noise Levels

We assign each latent an independent noise level, allowing the model to handle heterogeneous denoising states, enabling conditioning on partially clean latents at inference and allowing flexible sampling strategies.

**Heterogeneous noise scheduling.** We assign each motion latent $x_i$ an independent timestep $t_i^x \in [0, 1]$, sampled according to the $\bar{t}$ algorithm [59], avoiding collapse of the mean to a Bates distribution (c.f. appendix for details). For action plan latents $y_i$, a single global timestep $t^y \sim \mathcal{U}(0, 1)$ is sufficient and reduces complexity. We follow Rectified Flow [35] with velocity parameterization, i.e. our learned denoising function $G_\theta$ is formulated as:

$$(\hat{\mathbf{x}}_v, \hat{\mathbf{y}}_v) = G_\theta(\mathbf{x}_{\mathbf{t}^x}, \mathbf{y}_{t^y}; \mathbf{t}^x, t^y, c), \tag{2}$$

where $\mathbf{t}^x = \{t_1^x, \dots, t_N^x\}$ and $\mathbf{x}_{\mathbf{t}^x}, \mathbf{y}_{t^y}$ are noisy states of motion and action latents, according to the Rectified Flow forward process [35].

**Training with mixed datasets.** To supervise action plans $y_i$, we require a dataset with both sequence level text $c$ and frame-level text annotations. HumanML3D-272 [60] provides sequence level text while BABEL-272 [48, 60] additionally provides frame-level annotations for a subset of samples. To leverage the best of two datasets, we propose mixed training using both datasets with masking. The standard Rectified Flow loss for text prediction is $\mathcal{L}_{\text{text}} = \|\hat{\mathbf{y}}_v - (\mathbf{y}_0 - \epsilon)\|_2^2$, where $\epsilon \sim \mathcal{N}(0, I)$ is the pure noise and $\mathbf{y}_0$ the clean latents. We introduce an indicator $w \in \{0, 1\}$ to dynamically disable the text loss:

$$\mathcal{L}_{\text{text}} = w\, \|\hat{\mathbf{y}}_v - (\mathbf{y}_0 - \epsilon)\|_2^2 \tag{3}$$

where $w = 1$ if frame-level text annotations are available for this sequence. The training objective of our model is then the reconstruction loss for motion and text, and per-latent variance loss $\mathcal{L}_{\text{var}}$ [59]:

$$\mathcal{L} = \lambda_x\, \mathcal{L}_{\text{motion}} + \lambda_y\, \mathcal{L}_{\text{text}}, \tag{4}$$

where $\mathcal{L}_{\text{motion}} = \|\hat{\mathbf{x}}_v - (\mathbf{x}_0 - \epsilon)\|_2^2$ and $\mathbf{x}_0$ denotes the clean motion latents.

### 3.4 Flexible Sampling with Action Plan Generation

The flexible denoising formulation in Sec. 3.3 enables inference-time sampling strategies with independently scheduled text and motion timesteps at each frame. Since global semantic structure, *i.e.*, what action occurs when, is crucial for maintaining consistency in long and complex motions, we first generate an action plan and then synthesize the full motion using overlapping denoising windows with random (offline) or raster (streaming) orders.

**Stage 1: Action plan generation**. We keep all motion latents at pure noise ($t_i^x = 1, \forall i$) and denoise the action latents $\mathbf{y}$ over $T$ steps: $t_s^y = 1 - s\,\Delta t, \quad s = 0, \dots, T$. At the end of this stage we obtain clean frame-level text latents $\hat{\mathbf{y}}_0$, the *action plan*, which specify what action occurs at each temporal position before any motion is generated.

**Stage 2: Progressive denoising schedule.** Given the action plan, we generate the full motion while conditioning on frame-level semantics. Although sequentially denoising frame-after-frame is a natural solution, it is computationally prohibitive, see Fig. 2. Instead, we overlap the denoising intervals of individual frames [7, 59]: motion frames are partitioned into a denoising set and a waiting set. After a few iterations, one additional noisy latent is moved from waiting set to the denoising set (in  Fig. 2 and Fig. 3 such activation is indicated as a white dot). Thus, multiple active latents are updated simultaneously with individual denoising progress, increasing throughput.

**Offline:** Motion latents are activated in random order without temporal constraints. We find that random activation yields the best generation quality.

**Online / Streaming:** Latents are activated sequentially in temporal order. While such causal activation typically leads to degraded motion quality due to limited future context, our action plan conditioning preserves global semantic consistency during streaming and heavily reduces this degradation. In online mode, we denoise the action plan alongside the first motion latent.

## 4 Experiments

In this section, we compare ActionPlan with state-of-the-art (SOTA) T2M generation methods, ablate key design choices and showcase downstream applications enabled by our flexible architecture. The results show that ActionPlan outperforms all prior methods in both online streaming and offline mode, and that our action plan generation improves motion quality.

### 4.1 Experimental Setup

**Dataset.** We follow the evaluation protocol introduced by MotionStreamer [60] and conduct experiments on HumanML3D-272 [60] with additional frame-level text labels from BABEL [48]. MotionStreamer corrects inaccuracies in original HumanML3D [21] by using 272-D pose representation and resamples all motions to 30 FPS, yielding higher-quality data. BABEL includes fine-grained frame-level text annotation and partially overlaps with the motion sequences in HumanML3D. We train our method with the same set of training motions from HumanML3D-272 as used in previous methods [21, 60] and query frame-text labels when they are available from BABEL to supervise action plan generation.

**Evaluation metrics.** We report a comprehensive set of metrics following [21, 60]. To assess generation quality, we measure Fréchet Inception Distance (FID) [23]. For semantic alignment, we report R-Precisions and Multi-modal Matching Score (MatchS), which evaluate retrieval accuracy and feature-space proximity between generated motions and their text descriptions, respectively [21]. We additionally measure Diversity [21] to quantify variation across generated samples.

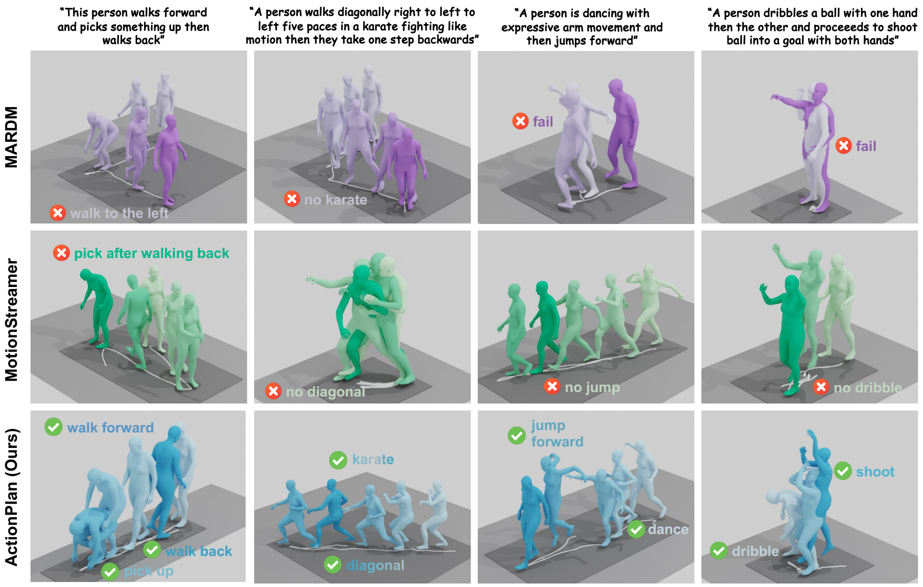

"This person walks forward and picks something up then walks back"
"A person walks diagonally right to left to left five paces in a karate fighting like motion then they take one step backwards"
"A person is dancing with expressive arm movement and then jumps forward"
"A person dribbles a ball with one hand then the other and proceeeds to shoot ball into a goal with both hands"
MARDM
✗ walk to the left
✗ no karate
✗ fail
✗ fail
MotionStreamer
✗ pick after walking back
✗ no diagonal
✗ no jump
✗ no dribble
ActionPlan (Ours)
✓ walk forward
✓ walk back
✓ pick up
✓ karate
✓ diagonal
✓ jump forward
✓ dance
✓ shoot
✓ dribble

**Fig. 4: Qualitative comparison** with MARDM [41] and MotionStreamer [60] on four text prompts. The color varies from light to dark representing the time flow. Incorrectly generated or missing actions are marked with ×, and correctly executed actions with ✓. By generating a frame-level action plan prior to motion synthesis, ActionPlan faithfully executes all specified actions in the correct order, while baselines frequently miss or misorder key actions. See Supp. for videos.

**Implementation details.** The latent denoising model is implemented as a lightweight Transformer [56] with 16 layers, 16 attention heads, and a hidden dimension of 1024 . We use CLIP to embed both frame level text and sequence-level text. For the frame level text, we use an autoencoder to reduce the original textual dimension to 16. In order to retrieve the text from the latent space, we decode the 16-d latents and use a simple KNN method on the CLIP embeddings, following [31]. During inference, we use 2 denoising steps for progressive sampling (Tab. 3) and 25 total denoising steps for rectified flow.

### 4.2 Baseline Comparison on Text-to-motion Generation

**Quantitative evaluation.** We follow the training and evaluation setup of <MotionStreamer [60]: all models are trained on the HumanML3D-272 [21, 60] training set, and evaluated with their pretrained TMR-based evaluator [43, 60]. All baselines are re-trained from scratch using their own official implementations and the same 272-D representation for fair comparison.

As shown in Tab. 1, a single ActionPlan checkpoint supports both offline and streaming generation simply by switching the sampling strategy at inference time, no retraining or fine-tuning is required. Both variants consistently outperform all competing methods across all metrics, including diffusion- [11, 54],

| **Method** | **FID** ↓ | **R-Precision**↑ | | | **MatchS** ↓ | **Div** → |
|---|---|---|---|---|---|---|
| | | **Top 1** | **Top 2** | **Top 3** | | |
| Real | 0.002 | 0.702 | 0.864 | 0.914 | 15.151 | 27.492 |
| MDM [54] | 23.454 | 0.523 | 0.692 | 0.764 | 17.423 | 26.325 |
| MLD [11] | 18.236 | 0.546 | 0.730 | 0.792 | 16.638 | 26.352 |
| T2M-GPT [63] | 12.475 | 0.606 | 0.774 | 0.838 | 16.812 | 27.275 |
| MotionGPT [28] | 14.375 | 0.456 | 0.598 | 0.628 | 17.892 | 27.114 |
| MoMask [20] | 12.232 | 0.621 | 0.784 | 0.846 | 16.138 | 27.127 |
| AttT2M [74] | 15.428 | 0.592 | 0.765 | 0.834 | 15.726 | 26.674 |
| MotionStreamer [60] | 11.790 | 0.631 | 0.802 | 0.859 | 16.081 | 27.284 |
| MARDM [41] | 7.044 | 0.669 | 0.806 | 0.860 | 15.892 | 26.235 |
| **ActionPlan-Streaming** | 5.735 | 0.672 | 0.822 | 0.877 | 15.315 | **27.287** |
| **ActionPlan-Offline** | **5.522** | **0.687** | **0.838** | **0.892** | **15.09** | 27.272 |

**Table 1: Quantitative comparison** with SOTA T2M generation methods on HumanML3D-272 [21, 60] test set. MatchS and Div denote the matching score and diversity respectively. **Bold** indicates best results, underline indicates second best. ActionPlan achieves better performance in both offline and online-streaming mode, maintaining both efficiency and high quality motion generation.

VQ- [20, 28, 63], streaming- [60], and masked auto-regressive [41] approaches. Notably, our offline variant achieves substantial gains over the best baseline, with 22% improvement in FID and R-Precision@3 reaching 0.892. Our online mode, being comparable to our offline mode, is 51% better in FID than the best streaming method MotionStreamer [60].

**Runtime performance.** To evaluate practical efficiency, we measure the latency for generating one motion token (4 frames) and compare our method in online streaming mode with baseline MARDM [41] and MotionStreamer [60] in the table below (unit: ms). All experiments are done on a single NVIDIA A100 GPU. MARDM [41] and MotionStreamer [60] generate motions in an autoregressive manner which is more efficient than full motion sampling yet they still require full denoising for every token. In contrast, our overlapping schedule requires a small overhead for the first token but is significantly faster on the following tokens, achieving a 1.44×–2.47× speedup for the initial token and 5.25×–9× during continuous streaming. Notably, our streaming mode achieves significant speedup compared to baselines while maintaining much better motion quality, demonstrating robust adaptability of our method.

| Method | First ↓ | Others ↓ |
|---|---|---|
| MARDM | 210 | 210 |
| MotionStreamer | 360 | 360 |
| Ours | **146** | **40** |

**Qualitative comparisons.** Fig. 4 presents qualitative comparisons with MotionStreamer [60] and MARDM [41] on the text-to-motion task. Despite its expensive masked auto-regressive inference, MARDM frequently fails to capture fine-grained semantic details: given the prompt *"a person walks forward and*

| **Configurations** | **Training** | **ActionPlan** | **Mode** | **FID** ↓ | **R-Precision**↑ | | | **MatchS**↓ | **Div** → |
|---|---|---|---|---|---|---|---|---|---|
| | **data** | **generation** | | | **Top 1** | **Top 2** | **Top 3** | | |
| Real motion | | | | 0.002 | 0.702 | 0.864 | 0.914 | 15.151 | 27.492 |
| **(A)** No frame-level text | partial | ✗ | Offline | 9.674 | 0.634 | 0.790 | 0.851 | 15.764 | 27.207 |
| | | | Streaming | 10.693 | 0.627 | 0.785 | 0.848 | 15.931 | 27.212 |
| **(B)** Frame-level text w/ actionplan | partial | ✓ | Offline | 7.863 | 0.625 | 0.791 | 0.854 | 15.849 | 27.232 |
| | | | Streaming | 8.018 | 0.625 | 0.789 | 0.854 | 15.866 | 27.220 |
| **(C)** No frame-level text | full | ✗ | Offline | 6.952 | 0.661 | 0.814 | 0.873 | 15.390 | 27.275 |
| | | | Streaming | 8.648 | 0.655 | 0.810 | 0.869 | 15.586 | **27.287** |
| **(D)** Frame-level text T&M co-gen | full | ✗ | Offline | 6.093 | 0.677 | 0.822 | 0.878 | 15.234 | 27.186 |
| **(E)** Frame-level text w/ actionplan | full | ✓ | Offline | **5.522** | **0.687** | **0.838** | **0.892** | **15.086** | 27.272 |
| | | | Streaming | 5.878 | 0.675 | 0.821 | 0.875 | 15.369 | 27.228 |

**Table 2: Ablation studies** for both Offline and Streaming modes. Training on the full dataset (C–E) consistently outperforms the partial intersection subset (A–B), validating our masked loss design. Frame-level text prediction (D, E) improves over no frame text (C). Action plan generation (E) further outperforms joint co-generation (D). These gains hold across both inference modes. **Bold**: best, underline: second best.

*picks something up then walks back"*, it generates a leftward walk without the pick-up action. MotionStreamer, as a strictly causal streaming method, suffers from action drift and incompleteness due to the lack of future context: it omits the dribbling action in *"a person dribbles a ball then shoots into a goal"* and misorders actions in sequential prompts. In contrast, ActionPlan faithfully executes all specified actions in the correct order across all examples, demonstrating that the frame-level action plan effectively grounds generation to the full semantic content of the input prompt.

## 4.3 Ablation Studies

We conduct comprehensive ablation studies to validate the key design choices of our framework and report results in Tab. 2 and Tab. 3. All methods are evaluated on the same HumanML3D-272 [60] test set.

**Frame-level text semantics prediction.** We propose to additionally predict text for each frame, enabling more comprehensive semantic alignment between text and motion. We train models with and without frame text prediction on the full HumanML3D-272 training set. Even with simple joint generation of per-frame text and motion, our frame-level text prediction (Tab. 2D) already improves over no text prediction (Tab. 3C).

**Mixed dataset training.** To train on sequences that do not have frame-level text annotation, we propose a simple mask loss ( Eq. (3)) to fully leverage the complete HumanML3D [21] dataset. Compared to models (Tab. 2A, B) trained on the overlapping subset between HumanML3D and BABEL, only 30% of the full data, models trained on full data (Tab. 2C, E) are consistently better. This underscores the importance of data scale and effective curation strategies.

**Action Plan generation.** At inference time, we propose to first generate action plans and then complete the full motion. We show in Tab. 2 that this two-stage generation (Tab. 2E) is better than simple joint generation (Tab. 2D) of both

| **Progressive Sampling** | **FID↓** | **R-Precision↑** | | | **MatchS↓** | **Div→** |
|---|---|---|---|---|---|---|
| | | **Top 1** | **Top 2** | **Top 3** | | |
| Real | 0.002 | 0.702 | 0.864 | 0.914 | 15.151 | 27.492 |
| Fully overlap (Parallel) | **5.420** | 0.681 | 0.836 | 0.886 | 15.104 | 26.989 |
| 2-step non-overlap | 5.522 | **0.687** | **0.838** | **0.892** | **15.086** | **27.272** |
| 5-steps non-overlap | 5.721 | **0.687** | 0.834 | 0.885 | 15.128 | 27.179 |
| 10-steps non-overlap | 5.605 | 0.685 | 0.832 | 0.885 | 15.128 | 27.261 |
| 15-steps non-overlap | 5.612 | 0.686 | 0.834 | 0.883 | 15.175 | 27.113 |
| Fully non-overlap (Random) | 5.566 | 0.683 | 0.828 | 0.880 | 15.160 | 27.138 |

**Table 3: Ablation on sampling strategy.** All rows use the same trained model with a fixed total of 25 denoising steps, differing only in the overlap between consecutively activated motion latents, ranging from fully parallel activation to fully sequential (non-overlapping) denoising. Our chosen schedule denoises each activated latent for 2 steps before activating the next, achieving the best trade-off between motion quality (FID) and text-motion alignment (R-Precision).

text and motion with single timestep. This suggests that making motion aware of the future through action plans is crucial for good performance.

**Progressive sampling strategy.** To support diverse tasks while maintaining efficiency, we propose to denoise in a pyramid fashion: gradually add one more latent to denoise multiple motion latents together, each with different timesteps. This independent timestep improves efficiency while maintaining flexibility. Alternatively, one can change the overlap window size $K = 2$ between two consecutively added latents to enable faster (more latents are denoised at the same time) or slower (less latents are denoised at the same time). We ablate this design choice in Tab. 3. It can be seen that the chosen size $K = 2$ achieves the best balance between motion quality and text alignment.

### 4.4 User Study

To further evaluate qualitative results, we conduct user studies on randomly selected HumanML3D-272 test prompts to evaluate text-to-motion generation and long motion generation. For each question, we present side-by-side animations in randomized order, asking them to select the best motion based on two criteria: (1) *semantic consistency*, how well the motion matches the given text description; and (2) *realism*, how natural, realistic, and smooth the motion appears.

**Text-to-motion generation.** We compare against MotionStreamer [60] and MARDM [41] on 20 prompts and release the user study to 30 participants. ActionPlan is preferred by 67.5% of participants, with MARDM and MotionStreamer receiving 12.2% and 20.3% of preferences respectively.

**Long motion generation.** We randomly select 20 long-horizon sequences, each composed of multiple prompts and compare against MotionStreamer [60] with

30 participants. ActionPlan is preferred by 67.7% of participants over 32.3% for MotionStreamer.

ActionPlan is consistently preferred by participants by a large margin in both user studies, closely matching the quantitative improvements reported in Tab. 1. The strong alignment between human judgments and automatic metrics highlights not only superior motion quality, realism, and smoothness, but also more accurate semantic alignment with the input text. Together, these results provide compelling evidence that our method substantially outperforms prior approaches across both perceptual and objective evaluations.

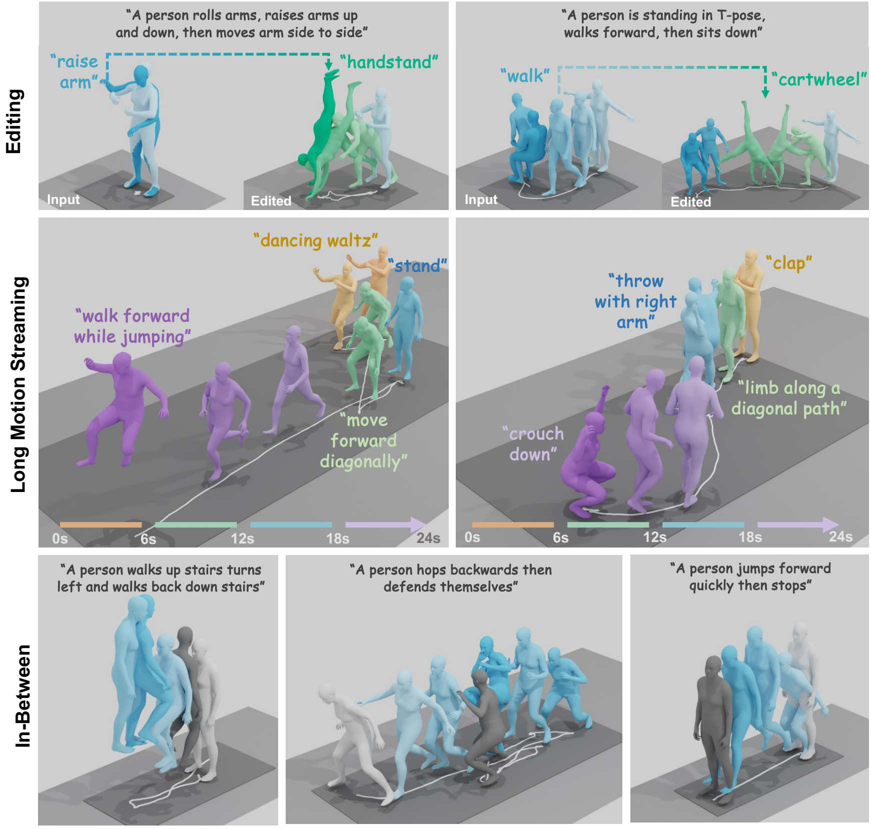

Editing
"A person rolls arms, raises arms up and down, then moves arm side to side"
"raise arm"
"handstand"
Input
Edited
"A person is standing in T-pose, walks forward, then sits down"
"walk"
"cartwheel"
Input
Edited
Long Motion Streaming
"dancing waltz"
"stand"
"walk forward while jumping"
"move forward diagonally"
0s
6s
12s
18s
24s
"clap"
"throw with right arm"
"limb along a diagonal path"
"crouch down"
0s
6s
12s
18s
24s
In-Between
"A person walks up stairs turns left and walks back down stairs"
"A person hops backwards then defends themselves"
"A person jumps forward quickly then stops"

**Fig. 5: ActionPlan supports diverse downstream applications zero-shot**. Darker shading indicates later time steps. **Editing** (top): regenerates selected latents conditioned on a new prompt (green) while preserving others. **Long motion streaming** (middle): generates coherent long-horizon motion in successive chunks across prompts. **In-betweening** (bottom): given fixed start (white) and end (dark grey) poses, fills in the intermediate motion. See Supp. for video results.

### 4.5 Applications

Benefiting from our flexible training strategy, ActionPlan supports multiple generation modes without retraining or fine-tuning. We present three downstream applications below and refer readers to the supplementary for video results.

**Motion editing.** Given a motion sequence generated by ActionPlan, we fix a subset of latents and regenerate the rest conditioned on them and a new prompt. This is achieved by setting the fixed latent's timesteps to $t_i^x = 0$ during sampling. As shown in Fig. 5, the edited regions reflect the new intent while smoothly blending with preserved segments.

**Motion in-betweening.** Given fixed start and end poses and a text prompt, ActionPlan fills in the intermediate motion by treating boundary frames as fully denoised anchors and applying progressive denoising the remaining latents conditioned on both the boundaries and text. As shown in Fig. 5, the infilled motions are temporally smooth, and faithful to the input prompt.

**Long motion generation.** Given a sequence of prompts, ActionPlan generates long-horizon motion in successive chunks, each conditioned on its prompt and the previous chunk's final tokens for continuity. As shown in Fig. 5, it produces temporally smooth motions that follow the intended action sequence.

## 5 Conclusion and Limitations

We presented ActionPlan, a hierarchical diffusion framework that unifies offline and streaming text-to-motion generation within a single model. By decoupling frame-level action planning from motion synthesis and introducing latent-specific noise schedules, ActionPlan enables future-aware semantic conditioning across a diverse set of tasks, such as streaming generation, offline generation, motion editing, and in-betweening, all in one single model without fine tuning.

Experiments on HumanML3D-272 demonstrate that our offline mode outperforms prior methods by 21% in FID and online mode achieves 51% improvement in FID compared to the best streaming method while being 5.2× faster. Our method is consistently favored by human evaluators, showing more than 67% preference in all user studies. We further conduct comprehensive ablations which validate the importance of our action plan generation in various setups. Results also show that our method enables zero-shot applications for diverse tasks while keeping the generated motions smooth and faithful to diverse text control. Our code and model will be publicly released.

**Limitations.** While ActionPlan enables flexible text-to-motion generation and supports multiple downstream tasks, there are several limitations. Firstly, it cannot yet model finger articulation or facial expressions which requires additional datasets with paired text descriptions for hands and faces. Secondly, our method generates human motion without scene or object awareness which limits its application in interactive agents or robotics that require models to actually interact with environment to finish complex daily tasks. We leave these for future works.

**Acknowledgments:** Special thanks RVH members for the help and discussion. Prof. Gerard Pons-Moll is member of the Machine Learning Cluster of Excellence, EXC number 2064/1 - Project number 390727645. Gerard Pons-moll is endowed by the Carl Zeiss Foundation. Jan Eric Lenssen is supported by the German Research Foundation (DFG) - 556415750 (Emmy Noether Programme, project: Spatial Modeling and Reasoning). Xianghui Xie is partially funded by NVIDIA fellowship.

EN and CL contributed equally as joint first authors; they are allowed to change their order freely on their resume and website. CL initialized the core idea, organized the project, co-developed the method, co-supervised the experiments, and wrote the draft along with figures. EN co-developed the method, led the implementation of most prototypes and demo development, and conducted most experiments. YH co-supervised the experiments, contributed to draft refining and result visualization, including the teaser and result figures. XX co-supervised the experiments, contributed to draft refining and the application demo.

## References

1. Ahuja, C., Morency, L.P.: Language2pose: Natural language grounded pose forecasting. In: 2019 International conference on 3D vision (3DV). pp. 719–728. IEEE (2019) 3
2. Azadi, S., Shah, A., Hayes, T., Parikh, D., Gupta, S.: Make-an-animation: Large-scale text-conditional 3d human motion generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15039–15048 (2023) 3
3. Baade, A., Chan, E.R., Sargent, K., Chen, C., Johnson, J., Adeli, E., Fei-Fei, L.: Latent forcing: Reordering the diffusion trajectory for pixel-space image generation. arXiv preprint arXiv:2602.11401 (2026) 4
4. Barquero, G., Escalera, S., Palmero, C.: Seamless human motion composition with blended positional encodings. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 457–469 (2024) 2
5. Cai, Y., Wu, Y., Li, K., Zhou, Y., Zheng, B., Liu, H.: Flooddiffusion: Tailored diffusion forcing for streaming motion generation. arXiv preprint arXiv:2512.03520 (2025) 4
6. Cen, Z., Pi, H., Peng, S., Shuai, Q., Shen, Y., Bao, H., Zhou, X., Hu, R.: Ready-to-react: Online reaction policy for two-character interaction generation. In: ICLR (2025) 4
7. Chen, B., Martí Monsó, D., Du, Y., Simchowitz, M., Tedrake, R., Sitzmann, V.: Diffusion forcing: Next-token prediction meets full-sequence diffusion. Advances in Neural Information Processing Systems **37**, 24081–24125 (2024) 2, 4, 8, 22
8. Chen, C., Zhang, J., Lakshmikanth, S.K., Fang, Y., Shao, R., Wetzstein, G., Fei-Fei, L., Adeli, E.: The language of motion: Unifying verbal and non-verbal language of 3d human motion. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 6200–6211 (2025) 3
9. Chen, R., Shi, M., Huang, S., Tan, P., Komura, T., Chen, X.: Taming diffusion probabilistic models for character control. In: ACM SIGGRAPH 2024 Conference Papers. pp. 1–10 (2024) 3, 4
10. Chen, W., Jia, H., Lai, S., Wu, K., Xiao, H., Hu, L., Yue, Y.: Free-t2m: Frequency enhanced text-to-motion diffusion model with consistency loss. arXiv preprint arXiv:2501.18232 (2025) 3

11. Chen, X., Jiang, B., Liu, W., Huang, Z., Fu, B., Chen, T., Yu, G.: Executing your commands via motion diffusion in latent space. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 18000–18010 (2023) 2, 3, 9, 10
12. Cho, J., Kim, J., Kim, J., Kim, M., Kang, M., Hong, S., Oh, T.H., Yu, Y.: Discord: Discrete tokens to continuous motion via rectified flow decoding. arXiv preprint arXiv:2411.19527 (2024) 3
13. Dabral, R., Mughal, M.H., Golyanik, V., Theobalt, C.: Mofusion: A framework for denoising-diffusion-based motion synthesis. In: Computer Vision and Pattern Recognition (CVPR) (2023) 3
14. Dai, W., Chen, L.H., Wang, J., Liu, J., Dai, B., Tang, Y.: Motionlcm: Real-time controllable motion generation via latent consistency model. In: European Conference on Computer Vision. pp. 390–408. Springer (2024) 4
15. Fan, K., Lu, S., Dai, M., Yu, R., Xiao, L., Dou, Z., Dong, J., Ma, L., Wang, J.: Go to zero: Towards zero-shot motion generation with million-scale data. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13336–13348 (2025) 3
16. Fan, K., Zhang, J., Yi, R., Gong, J., Wang, Y., Wang, Y., Tan, X., Wang, C., Ma, L.: Textual decomposition then sub-motion-space scattering for open-vocabulary motion generation. arXiv preprint arXiv:2411.04079 (2024) 3
17. Fan, L., Li, T., Qin, S., Li, Y., Sun, C., Rubinstein, M., Sun, D., He, K., Tian, Y.: Fluid: Scaling autoregressive text-to-image generative models with continuous tokens. arXiv preprint arXiv:2410.13863 (2024) 4
18. Ghosh, A., Zhou, B., Dabral, R., Wang, J., Golyanik, V., Theobalt, C., Slusallek, P., Guo, C.: Duetgen: Music driven two-person dance generation via hierarchical masked modeling. In: Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers. pp. 1–11 (2025) 3
19. Guo, C., Hwang, I., Wang, J., Zhou, B.: Snapmogen: Human motion generation from expressive texts. arXiv preprint arXiv:2507.09122 (2025) 3
20. Guo, C., Mu, Y., Javed, M.G., Wang, S., Cheng, L.: Momask: Generative masked modeling of 3d human motions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1900–1910 (2024) 3, 4, 10
21. Guo, C., Zou, S., Zuo, X., Wang, S., Ji, W., Li, X., Cheng, L.: Generating diverse and natural 3d human motions from text. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022) 2, 3, 5, 8, 9, 10, 11, 22
22. Guo, C., Zuo, X., Wang, S., Cheng, L.: Tm2t: Stochastic and tokenized modeling for the reciprocal generation of 3d human motions and texts. In: European Conference on Computer Vision. pp. 580–597. Springer (2022) 3
23. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016) 8
24. He, Y., Tiwari, G., Zhang, X., Bora, P., Birdal, T., Lenssen, J.E., Pons-Moll, G.: Molingo: Motion–language alignment for text-to-human motion generation. In: CVPR (2026) 3, 4
25. Ho, J., Salimans, T.: Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598 (2022) 22
26. Hong, F., Guzov, V., Kim, H.J., Ye, Y., Newcombe, R., Liu, Z., Ma, L.: Egolm: Multi-modal language model of egocentric motions. arXiv preprint arXiv:2409.18127 (2024) 3

27. Huang, Y., Yang, H., Luo, C., Wang, Y., Xu, S., Zhang, Z., Zhang, M., Peng, J.: Stablemofusion: Towards robust and efficient diffusion-based motion generation framework. In: Proceedings of the 32nd ACM International Conference on Multimedia. pp. 224–232 (2024) 3
28. Jiang, B., Chen, X., Liu, W., Yu, J., Yu, G., Chen, T.: Motiongpt: Human motion as a foreign language. Advances in Neural Information Processing Systems **36**, 20067–20079 (2023) 3, 10
29. Karunratanakul, K., Preechakul, K., Suwajanakorn, S., Tang, S.: Guided motion diffusion for controllable human motion synthesis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2151–2162 (2023) 3
30. Kim, J., Kim, J., Choi, S.: Flame: Free-form language-based motion synthesis & editing. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 8255–8263 (2023) 3, 4
31. Li, C., Chibane, J., He, Y., Pearl, N., Geiger, A., Pons-Moll, G.: Unimotion: Unifying 3d human motion synthesis and understanding. In: 2025 International Conference on 3D Vision (3DV). pp. 240–249. IEEE (2025) 3, 6, 9, 22
32. Li, T., Tian, Y., Li, H., Deng, M., He, K.: Autoregressive image generation without vector quantization. arXiv preprint arXiv:2406.11838 (2024) 4
33. Liang, H., Zhang, W., Li, W., Yu, J., Xu, L.: Intergen: Diffusion-based multi-human motion generation under complex interactions. International Journal of Computer Vision pp. 1–21 (2024) 3
34. Liu, P., Song, L., Huang, J., Liu, H., Xu, C.: Gesturelsm: Latent shortcut based co-speech gesture generation with spatial-temporal modeling. arXiv preprint arXiv:2501.18898 (2025) 3
35. Liu, X., Gong, C., Liu, Q.: Flow straight and fast: Learning to generate and transfer data with rectified flow. In: The Eleventh International Conference on Learning Representations (2023), https://openreview.net/forum?id=XVjTT1nw5z 7
36. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: SMPL: A skinned multi-person linear model. ACM Transactions on Graphics, (Proc. SIGGRAPH Asia) **34**(6), 248:1–248:16 (Oct 2015) 5
37. Lu, S., Chen, L.H., Zeng, A., Lin, J., Zhang, R., Zhang, L., Shum, H.Y.: Humantomato: Text-aligned whole-body motion generation. arXiv preprint arXiv:2310.12978 (2023) 3
38. Lu, S., Wang, J., Lu, Z., Chen, L.H., Dai, W., Dong, J., Dou, Z., Dai, B., Zhang, R.: Scamo: Exploring the scaling law in autoregressive motion generation model. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 27872–27882 (2025) 3
39. Mahmood, N., Ghorbani, N., Troje, N.F., Pons-Moll, G., Black, M.J.: Amass: Archive of motion capture as surface shapes. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 5442–5451 (2019) 22
40. Meng, Z., Han, Z., Peng, X., Xie, Y., Jiang, H.: Absolute coordinates make motion generation easy. arXiv preprint arXiv:2505.19377 (2025) 3
41. Meng, Z., Xie, Y., Peng, X., Han, Z., Jiang, H.: Rethinking diffusion for text-driven human motion generation: Redundant representations, evaluation, and masked autoregression. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 27859–27871 (2025) 2, 3, 4, 9, 10, 12
42. Petrovich, M., Black, M.J., Varol, G.: Temos: Generating diverse human motions from textual descriptions. In: European Conference on Computer Vision. pp. 480–497. Springer (2022) 3

43. Petrovich, M., Black, M.J., Varol, G.: Tmr: Text-to-motion retrieval using contrastive 3d human motion synthesis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9488–9497 (2023) 3, 9
44. Petrovich, M., Litany, O., Iqbal, U., Black, M.J., Varol, G., Bin Peng, X., Rempe, D.: Multi-track timeline control for text-driven 3d human motion generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1911–1921 (2024) 3
45. Pinyoanuntapong, E., Saleem, M., Karunratanakul, K., Wang, P., Xue, H., Chen, C., Guo, C., Cao, J., Ren, J., Tulyakov, S.: Maskcontrol: Spatio-temporal control for masked motion synthesis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9955–9965 (2025) 3, 4
46. Pinyoanuntapong, E., Saleem, M.U., Wang, P., Lee, M., Das, S., Chen, C.: Bamm: Bidirectional autoregressive motion model. In: European Conference on Computer Vision. pp. 172–190. Springer (2024) 3
47. Pinyoanuntapong, E., Wang, P., Lee, M., Chen, C.: Mmm: Generative masked motion model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1546–1555 (2024) 3, 4
48. Punnakkal, A.R., Chandrasekaran, A., Athanasiou, N., Quiros-Ramirez, A., Black, M.J.: BABEL: Bodies, action and behavior with english labels. In: Proceedings IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 722–731 (Jun 2021) 7, 8, 22
49. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models (2021) 4
50. Shafir, Y., Tevet, G., Kapon, R., Bermano, A.H.: Human motion diffusion as a generative prior. arXiv preprint arXiv:2303.01418 (2023) 3
51. Shi, Y., Wang, J., Jiang, X., Lin, B., Dai, B., Peng, X.B.: Interactive character control with auto-regressive motion diffusion models. ACM Transactions on Graphics (TOG) **43**(4), 1–14 (2024) 3, 4
52. Tevet, G., Gordon, B., Hertz, A., Bermano, A.H., Cohen-Or, D.: Motionclip: Exposing human motion generation to clip space. In: European Conference on Computer Vision. pp. 358–374. Springer (2022) 3
53. Tevet, G., Raab, S., Cohan, S., Reda, D., Luo, Z., Peng, X.B., Bermano, A.H., van de Panne, M.: Closd: Closing the loop between simulation and diffusion for multi-task character control. arXiv preprint arXiv:2410.03441 (2024) 3, 4
54. Tevet, G., Raab, S., Gordon, B., Shafir, Y., Cohen-Or, D., Bermano, A.H.: Human motion diffusion model. arXiv preprint arXiv:2209.14916 (2022) 2, 3, 4, 9, 10
55. Tu, L., Meng, L., Li, Z., Ling, H., Huang, S.: Autoregressive motion generation with gaussian mixture-guided latent sampling. In: The Thirty-ninth Annual Conference on Neural Information Processing Systems 3
56. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017) 9
57. Wan, W., Dou, Z., Komura, T., Wang, W., Jayaraman, D., Liu, L.: Tlcontrol: Trajectory and language control for human motion synthesis. In: European Conference on Computer Vision. pp. 37–54. Springer (2024) 3
58. Wang, Y., chuan guo, Mu, Y., Javed, M.G., Zuo, X., Lu, J., Jiang, H., cheng, L.: Motiondreamer: One-to-many motion synthesis with localized generative masked transformer. In: The Thirteenth International Conference on Learning Representations (2025), https://openreview.net/forum?id=d23EVDRJ6g 3

59. Wewer, C., Pogodzinski, B., Schiele, B., Lenssen, J.E.: Spatial reasoning with denoising models. In: International Conference on Machine Learning (ICML) (2025) 2, 4, 7, 8, 22
60. Xiao, L., Lu, S., Pi, H., Fan, K., Pan, L., Zhou, Y., Feng, Z., Zhou, X., Peng, S., Wang, J.: Motionstreamer: Streaming motion generation via diffusion-based autoregressive model in causal latent space. arXiv preprint arXiv:2503.15451 (2025) 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 22
61. Yu, Q., Watanabe, A., Fujiwara, K.: Causal motion diffusion models for autoregressive motion generation. In: CVPR (2026) 4
62. Yuan, Y., Song, J., Iqbal, U., Vahdat, A., Kautz, J.: Physdiff: Physics-guided human motion diffusion model. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 16010–16021 (2023) 3
63. Zhang, J., Zhang, Y., Cun, X., Huang, S., Zhang, Y., Zhao, H., Lu, H., Shen, X.: 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) (2023) 2, 3, 10
64. Zhang, M., Cai, Z., Pan, L., Hong, F., Guo, X., Yang, L., Liu, Z.: Motiondiffuse: Text-driven human motion generation with diffusion model. IEEE transactions on pattern analysis and machine intelligence **46**(6), 4115–4128 (2024) 2, 3
65. Zhang, M., Guo, X., Pan, L., Cai, Z., Hong, F., Li, H., Yang, L., Liu, Z.: Remodiffuse: Retrieval-augmented motion diffusion model. arXiv preprint arXiv:2304.01116 (2023) 3
66. Zhang, M., Jin, D., Gu, C., Hong, F., Cai, Z., Huang, J., Zhang, C., Guo, X., Yang, L., He, Y., et al.: Large motion model for unified multi-modal motion generation. arXiv preprint arXiv:2404.01284 (2024) 3
67. Zhang, M., Li, H., Cai, Z., Ren, J., Yang, L., Liu, Z.: Finemogen: Fine-grained spatio-temporal motion generation and editing. NeurIPS (2023) 3
68. Zhang, P., Liu, P., Garrido, P., Kim, H., Chaudhuri, B.: Kinmo: Kinematic-aware human motion understanding and generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11187–11197 (2025) 3
69. Zhang, Z., Liu, A., Reid, I., Hartley, R., Zhuang, B., Tang, H.: Motion mamba: Efficient and long sequence motion generation. In: European Conference on Computer Vision. pp. 265–282. Springer (2024) 3
70. Zhang, Z., Wang, Y., Li, D., Gong, D., Reid, I., Hartley, R.: Flashmo: Geometric interpolants and frequency-aware sparsity for scalable efficient motion generation. In: The Thirty-ninth Annual Conference on Neural Information Processing Systems 3
71. Zhang, Z., Liu, R., Hanocka, R., Aberman, K.: Tedi: Temporally-entangled diffusion for long-term motion synthesis. In: ACM SIGGRAPH 2024 Conference Papers. pp. 1–11 (2024) 3
72. Zhao, K., Li, G., Tang, S.: Dartcontrol: A diffusion-based autoregressive motion model for real-time text-driven motion control. arXiv preprint arXiv:2410.05260 (2024) 3, 4
73. Zheng, B., Chen, K., Yao, Y., Zeng, Z., Jiang, X., Wang, H., Lasenby, J., Jin, X.: Autokeyframe: Autoregressive keyframe generation for human motion synthesis and editing. In: Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers. pp. 1–12 (2025) 4
74. Zhong, C., Hu, L., Zhang, Z., Xia, S.: Attt2m: Text-driven human motion generation with multi-perspective attention mechanism. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 509–519 (2023) 10

75. Zhou, W., Dou, Z., Cao, Z., Liao, Z., Wang, J., Wang, W., Liu, Y., Komura, T., Wang, W., Liu, L.: Emdm: Efficient motion diffusion model for fast and high-quality motion generation. In: European Conference on Computer Vision. pp. 18–38. Springer (2024) 3
76. Zhou, Y., Barnes, C., Lu, J., Yang, J., Li, H.: On the continuity of rotation representations in neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5745–5753 (2019) 5
77. Zhou, Z., Wan, Y., Wang, B.: Avatargpt: All-in-one framework for motion understanding planning generation and beyond. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1357–1366 (2024) 3
78. Zhu, B., Jiang, B., Wang, S., Tang, S., Chen, T., Luo, L., Zheng, Y., Chen, X.: Motiongpt3: Human motion as a second modality. arXiv preprint arXiv:2506.24086 (2025) 3

In the following, we present additional details and results that complement the main paper. We first describe the action plan autoencoder in Sec. A, followed by the heterogeneous timestep sampling algorithm in Sec. B, architecture and training details in Sec. C, and sampling pseudocode in Sec. D. Finally, we present details of our user study in Sec. E.

## A Action Plan Autoencoder

As described in Sec. 3.2, we pre-train a lightweight MLP autoencoder to project CLIP text embeddings into a 16-dimensional latent space. This section provides the full architectural and training details.

**Architecture.** The encoder and decoder are both 3-layer MLPs with hidden dimensions 256 and 128, BatchNorm and GELU activations. The encoder maps from the CLIP embedding dimension ($d_{\text{CLIP}} = 512$) to 16, and the decoder maps back from 16 to $d_{\text{CLIP}}$.

**Training objectives.** The autoencoder is trained with three objectives. Given an input CLIP embedding $e \in \mathbb{R}^{d_{\text{CLIP}}}$, the encoder produces $z = f_{\text{enc}}(e) \in \mathbb{R}^{16}$ and the decoder reconstructs $\hat{e} = f_{\text{dec}}(z)$.
*(1) Reconstruction loss.* We minimize the MSE between the original and reconstructed embeddings in CLIP space:

$$\mathcal{L}_{\text{recon}} = \|\hat{e} - e\|_2^2. \tag{5}$$

*(2) Neighbor-preservation loss.* To preserve semantic similarity in the latent space, we encourage correct action label retrieval. For each input $e \in \mathbb{R}^{d_{\text{CLIP}}}$, we pre-compute its ground-truth label $y$ as the index of the nearest neighbor of $e$ in a fixed label bank $L \in \mathbb{R}^{6133 \times d_{\text{CLIP}}}$. We then treat retrieval as classification: for the reconstructed embedding $\hat{e}$, we compute cosine similarity scores $s_i = (\hat{e}^\top \ell_i)/(\tau \|\hat{e}\| \|\ell_i\|)$ against each normalized label $\ell_i$, and minimize cross-entropy against $y$:

$$\mathcal{L}_{\text{neighbor}} = -\log \frac{\exp(s_y/\tau)}{\sum_{j=1}^{K} \exp(s_j/\tau)} = \text{CE}\left(\text{softmax}\left(\frac{\hat{e}^\top L^\top}{\tau \|\hat{e}\| \|L\|}\right), y\right), \tag{6}$$

where $L$ is the $6133 \times d_{\text{CLIP}}$ label bank (rows $\ell_i$), $y$ is the ground-truth nearest-neighbor index, and $\tau$ is a temperature hyperparameter (default $\tau = 0.07$).
*(3) Variance regularization.* To prevent dimensional collapse, we encourage each latent dimension to maintain unit variance across the training batch:

$$\mathcal{L}_{\text{var}} = \sum_{j=1}^{16} \left(\text{Var}(z_j) - 1\right)^2, \tag{7}$$

where $z_j$ denotes the $j$-th dimension across all training samples in a batch. The total autoencoder loss is:

$$\mathcal{L}_{\text{AE}} = \mathcal{L}_{\text{recon}} + \lambda_{\text{n}} \mathcal{L}_{\text{neighbor}} + \lambda_{\text{v}} \mathcal{L}_{\text{var}}. \tag{8}$$

We use $\lambda_{\mathrm{n}} = 1.0$ and $\lambda_{\mathrm{v}} = 0.01$ by default.

**Text retrieval at inference.** Following [31], we decode the 16-d latents back to CLIP space using the decoder and retrieve the nearest action label via KNN over the CLIP embeddings of all action labels in the BABEL vocabulary.

## B Heterogeneous Timestep Sampling

As discussed in Sec. 3.3, each motion latent $x_i$ is assigned an independent timestep $t_i^x$ during training. Naively sampling each $t_i^x \sim \mathcal{U}(0,1)$ independently would cause the sequence mean $\bar{t}^x = \frac{1}{N}\sum_i t_i^x$ to concentrate around 0.5 due to the Bates distribution, undertraining the model at early ($\bar{t} \approx 0$) and late ($\bar{t} \approx 1$) denoising stages [7].

Following SRM [59], we instead first sample a mean timestep $\bar{t}^x \sim \mathcal{U}(0,1)$ and then perturb individual timesteps around this mean:

$$t_i^x = \mathrm{clip}(\bar{t}^x + \delta_i,\ 0,\ 1)\,, \quad \delta_i \sim \mathcal{N}(0, \sigma_t^2), \tag{9}$$

where $\sigma_t$ controls the spread of individual timesteps. This ensures that $\bar{t}^x$ remains uniformly distributed across training, providing balanced coverage of all denoising stages. We set $\sigma_t = 1.0$ in all experiments.

## C Architecture and Training Details

**Transformer denoiser architecture.** As stated in the main paper, our denoiser $G_\theta$ is a Transformer with 16 layers, 16 attention heads, and a hidden dimension of 1024. At each temporal position, the motion latent $x_i$ and action plan latent $y_i$ are concatenated into a single vector and linearly projected to the hidden dimension.

**Training hyperparameters.** We train with the Adam optimizer with a learning rate of 1e-4 and batch size of 32 on 4 NVIDIA A100 GPUs for 10,000 epochs, which takes approximately 26 hours. The loss weights are set to $\lambda_x = 1$ and $\lambda_y = 1$. We apply classifier-free guidance [25] by randomly dropping the sequence-level text condition $c$ with probability 0.1 during training and using a guidance scale of 5.5 at inference.

**Dataset details.** We utilize both HumanML3D [21] and BABEL [48], which independently annotate different subsets of AMASS [39]. The full HumanML3D-272 [60] training set contains 21,466 motion sequences with sequence-level text descriptions. Of these, only 8,829 sequences overlap with BABEL and have additional frame-level action labels. We do not apply left-right flipping augmentation when computing this overlap statistic.

## D Sampling Pseudocode

We provide detailed pseudocode for both offline (Algorithm 1) and streaming inference (Algorithm 2) modes. Both share the same Stage 1 (action plan generation) but differ in how motion latents are denoised in Stage 2.

**Algorithm 1** Offline Mode Sampling

```
Input: Text condition $c$, denoising steps $T$, overlap steps $K$, sequence length $N$
Output: Clean motion latents $\hat{\mathbf{x}}_0$
Stage 1: Action plan generation
Sample $\mathbf{x} \sim \mathcal{N}(0, \mathbf{I}), \mathbf{y} \sim \mathcal{N}(0, \mathbf{I})$, $t_i^x = 1$ for $\forall i$    ▷ Initialization
for $s = 0$ to $T - 1$ do
    $t^y = 1 - s/T$
    $(\_, \hat{\mathbf{y}}_v) = G_\theta(\mathbf{x}_1, \mathbf{y}; \mathbf{1}, t^y, c)$    ▷ cf. Eq. (2),$\mathbf{t}_x = \mathbf{1}$ and $\mathbf{x_1}$ are fixed for Stage 1
    $\mathbf{y} \leftarrow \mathbf{y} - \hat{\mathbf{y}}_v / T$    ▷ Update $\mathbf{y}$ with the predicted velocity
end for
$\hat{\mathbf{y}}_0 \leftarrow \mathbf{y}$    ▷ The clean action plan

Stage 2: Progressive motion denoising in random order
Generate random permutation $\pi$ of $\{1, \dots, N\}$
Initialize active set $\mathcal{A} = \emptyset$, per-latent progress $p_i = 0$ for $\forall i$, step counter $S = 0$
while any $p_i < T$ do
    if $\pi \neq \emptyset$ and $S \bmod K = 0$ then    ▷ Activate the next latent every $K$ steps
        $\mathcal{A} \leftarrow \mathcal{A} \cup \{\pi_k\}$    ▷ Activate the next random latent $\pi_k$
        $\pi \leftarrow \pi \setminus \{\pi_k\}$    ▷ Remove
    end if
    for each $i \in \mathcal{A}$ with $p_i < T$ do    ▷ Active latents NOT fully denoised
        $t_i^x = 1 - p_i / T$    ▷ Update timesteps
    end for
    $(\hat{\mathbf{x}}_v, \_) = G_\theta(\mathbf{x}, \hat{\mathbf{y}}_0; \mathbf{t}^x, \mathbf{0}, c)$    ▷ cf. Eq. (2), action plan $\hat{\mathbf{y}}_0$ is fixed for Stage 2
    for each $i \in \mathcal{A}$ with $p_i < T$ do
        $x_i = x_i - \hat{x}_{v,i} / T$    ▷ Update the active motion latents only
        $p_i \leftarrow p_i + 1$
    end for
    $S \leftarrow S + 1$
end while
$\hat{\mathbf{x}}_0 \leftarrow \mathbf{x}$
```

# E User Study Details

We provide additional details on the user study interface and design. The detailed instructions and interface layouts are shown in Fig. 6.

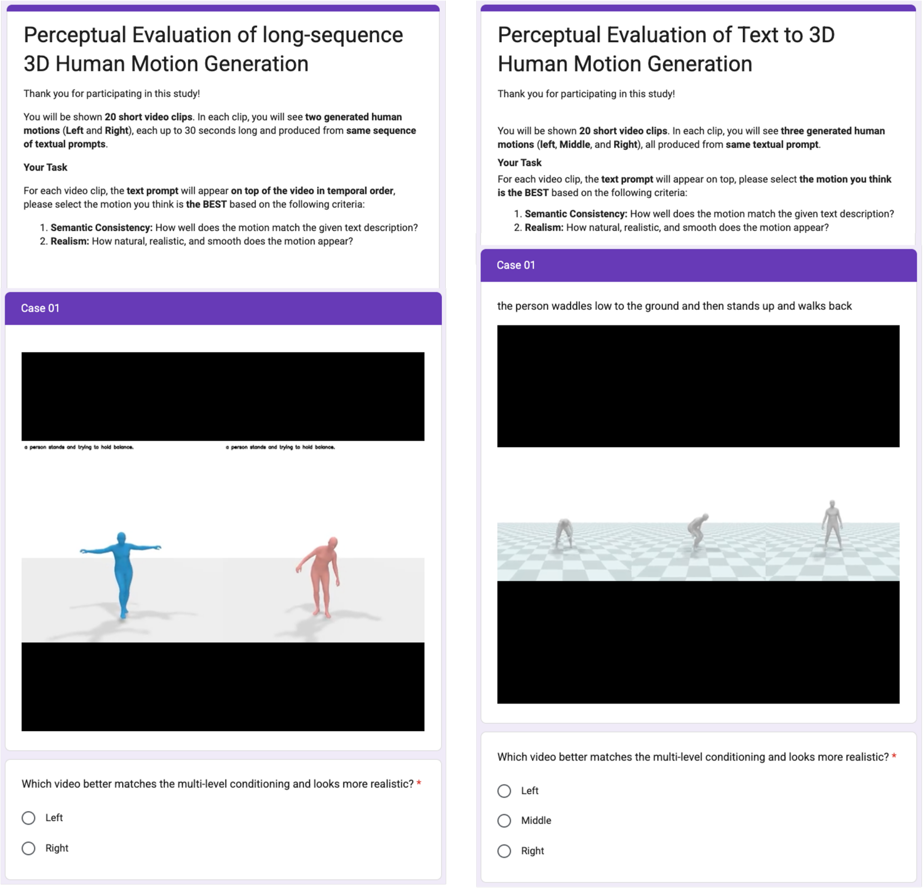
**Text-to-motion generation study (Fig. 6b).** Each trial presents three side-by-side animations produced from the same textual prompt, with method assignment to left, middle, and right positions randomized. The text prompt is displayed on top of the video. Participants are asked to select the motion that best matches the text description and looks most realistic.

**Long motion generation study (Fig. 6a).** Each trial presents two side-by-side animations generated from the same sequence of textual prompts (up to 30 seconds), with left-right assignment randomized. The text prompts are displayed on top of the video in temporal order so that participants can verify whether each action is faithfully executed. Participants select the preferred motion using the same two criteria.

**Algorithm 2** Streaming Mode Sampling

```
Input: Text condition $c$, denoising steps $T$, latent length $N$ $(T < N)$
Output: Streamed clean motion frames $\hat{\mathbf{x}}_0$
Stage 1: Action plan generation with the first motion latent fully denoised
Sample $\mathbf{x} \sim \mathcal{N}(0, \mathbf{I}), \mathbf{y} \sim \mathcal{N}(0, \mathbf{I})$        ▷ Initialization
Initialize active set $\mathcal{A} = \emptyset$, per-motion latent progress $p_i = 0$ for $\forall i$
for $s = 0$ to $T - 1$ do
    $t^y = 1 - s/T$
    $\mathcal{A} \leftarrow \mathcal{A} \cup \{x_s\}$        ▷ Activate the next latent $x_s$
    for each $i \in \mathcal{A}$ with $p_i < T$ do        ▷ Active latents NOT fully denoised
        $t_i^x = 1 - p_i/T$        ▷ Update timesteps
    end for
    $(\hat{\mathbf{x}}_v, \hat{\mathbf{y}}_v) = G_\theta(\mathbf{x}, \mathbf{y}; \mathbf{t}^x, t^y, c)$        ▷ cf. Eq. (2)
    for each $i \in \mathcal{A}$ with $p_i < T$ do
        $x_i = x_i - \hat{x}_{v,i}/T$        ▷ Update the active motion latents only
        $p_i \leftarrow p_i + 1$
    end for
    $\mathbf{y} \leftarrow \mathbf{y} - \hat{\mathbf{y}}_v/T$        ▷ Update the action plan
end for
$\hat{\mathbf{y}}_0 \leftarrow \mathbf{y}$        ▷ The clean action plan
$s \leftarrow s + 1$
Decode $x_1$ with the Causal TAE decoder to obtain motion frames

Stage 2: Sequential progressive denoising in raster order
while any $p_i < T$ do
    if $s < N$ then
        $\mathcal{A} \leftarrow \mathcal{A} \cup \{x_s\}$        ▷ Activate the next latent $x_s$
    end if
    for each $i \in \mathcal{A}$ with $p_i < T$ do        ▷ Active latents NOT fully denoised
        $t_i^x = 1 - p_i/T$        ▷ Update timesteps
    end for
    $(\hat{\mathbf{x}}_v, \_) = G_\theta(\mathbf{x}, \hat{\mathbf{y}}_0; \mathbf{t}^x, \mathbf{0}, c)$        ▷ cf. Eq. (2), action plan is fixed
    for each $i \in \mathcal{A}$ with $p_i < T$ do
        $x_i = x_i - \hat{x}_{v,i}/T$        ▷ Update the active motion latents only
        $p_i \leftarrow p_i + 1$
        if $p_i = T$ then
            Decode $x_i$ with the Causal TAE decoder to obtain motion frames
        end if
    end for
    $s \leftarrow s + 1$
end while
$\hat{\mathbf{x}}_0 \leftarrow \mathbf{x}$
```



Perceptual Evaluation of long-sequence 3D Human Motion Generation

Thank you for participating in this study!

You will be shown **20 short video clips**. In each clip, you will see **two generated human motions** (**Left** and **Right**), each up to 30 seconds long and produced from **same sequence of textual prompts**.

**Your Task**

For each video clip, the **text prompt** will appear **on top of the video in temporal order**, please select the motion you think is **the BEST** based on the following criteria:

1. **Semantic Consistency:** How well does the motion match the given text description?
2. **Realism:** How natural, realistic, and smooth does the motion appear?

Case 01

a person stands and trying to hold balance.
a person stands and trying to hold balance.

Which video better matches the multi-level conditioning and looks more realistic? *

○ Left

○ Right

Perceptual Evaluation of Text to 3D Human Motion Generation

Thank you for participating in this study!

You will be shown **20 short video clips**. In each clip, you will see **three generated human motions** (**left**, **Middle**, and **Right**), all produced from **same textual prompt**.

**Your Task**

For each video clip, the **text prompt** will appear on top, please select **the motion you think is the BEST** based on the following criteria:

1. **Semantic Consistency:** How well does the motion match the given text description?
2. **Realism:** How natural, realistic, and smooth does the motion appear?

Case 01

the person waddles low to the ground and then stands up and walks back

Which video better matches the multi-level conditioning and looks more realistic? *

○ Left

○ Middle

○ Right

**(a) Streaming user study interface**

**(b) T2M user study interface**

**Fig. 6: User study interfaces.** (a) Long-sequence streaming evaluation: participants compare two animations (left vs. right) generated from the same sequence of textual prompts. Note that the left side is always shown in blue and the right side is always shown in red, while the actual method assigned to each side is randomized. (b) Text-to-motion evaluation: participants compare three animations (left, middle, right) generated from the same single prompt. In both studies, method assignment is randomized and participants select the best motion based on semantic consistency and realism.