STREAMING MULTI-TRACK TIMELINE CONTROL FOR 3D HUMAN MOTION GENERATION

Anonymous authors
Paper under double-blind review

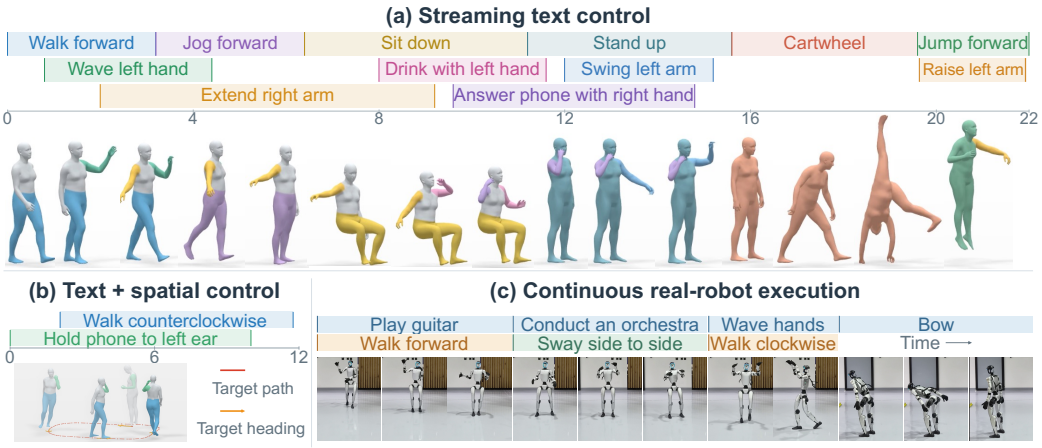

Figure 1: **Streaming multi-track motion generation with TimelineControl.** (a) Sequential and overlapping body-part instructions. (b) Joint text, root-path, and heading control. (c) Humanoid robot execution. In (a) and (b), body-part colors correspond to instructions.

ABSTRACT

Text-driven human motion generation has advanced substantially, yet most methods assume instructions are available before synthesis. Interactive applications require responding to new instructions while continuing ongoing actions, such as answering a phone while walking. Existing approaches address streaming generation or simultaneous composition without explicitly combining streaming instruction arrival with independently timed, overlapping actions. We introduce streaming multi-track timeline control and propose TimelineControl to incorporate new instructions alongside ongoing actions. Interval-aware conditioning preserves instruction timing, while causal part-structured representations and part-aware denoising coordinate concurrent actions across body regions. We also construct TimelineMotion, a dataset with overlapping instruction intervals and body-part annotations. Experiments on TimelineMotion and MTT demonstrate improved semantic alignment and temporal adherence over evaluated streaming baselines, including models retrained on the same data. Ablations and human evaluations validate our design, complemented by spatial conditioning and humanoid execution demonstrations. *Our code, data and models will become publicly available.*

1 INTRODUCTION

Text-driven human motion generation has made substantial progress in synthesizing realistic movements from natural language (Petrovich et al., 2022; Tevet et al., 2023; Chen et al., 2023; Guo et al., 2024), including action sequences and simultaneous motions (Athanasiou et al., 2022; Petrovich et al., 2024; Li et al., 2026). However, most approaches assume that instructions are available before generation begins. Interactive applications in gaming and robotics require both reactivity and compositionality: a character may need to answer a ringing phone while walking, or avoid a moving obstacle while continuing another activity. Such interactions require incorporating new actions while preserving compatible ongoing behaviors and their individual timings.

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.

This setting poses three coupled challenges: generating motion without access to future instructions, maintaining distinct temporal scopes for concurrent actions, and coordinating their effects across body regions while preserving whole-body coherence. Previous work addresses these challenges separately. Streaming methods such as MotionStreamer (Xiao et al., 2025), ActionPlan (Nazarenus et al., 2026), and ARDY (Zhao et al., 2026) support changing text conditions, but their described interfaces do not explicitly maintain independently timed, overlapping instructions. Conversely, SINC (Athanasiou et al., 2023), STMC (Petrovich et al., 2024), and FrankenMotion (Li et al., 2026) support simultaneous or temporal composition assume pre-specified conditions.

We introduce streaming multi-track timeline control, where instructions arrive at their start times, retain separate temporal intervals, and may overlap with ongoing actions, as illustrated in Figure 1. Our framework, TimelineControl, extends ActionPlan with online multi-prompt conditioning and part-aware denoising. Instructions are encoded separately, with interval encoding and visibility masks determining their temporal applicability. A causal part-structured autoencoder provides anatomical motion latents, while text-derived relevance estimates guide each body region toward applicable instructions. We combine body-part and global features to produce coordinated whole-body motion. New instructions guide future motion without modifying previously generated motions.

For training and evaluation, we construct TimelineMotion with overlapping instruction intervals and body-part annotations through STMC-based composition of HumanML3D motions, followed by refinement and filtering. Experiments on TimelineMotion and MTT demonstrate strong semantic alignment and temporal adherence, outperforming evaluated streaming baselines, including ActionPlan retrained on the same data. Ablations validate our conditioning and part-aware components, while human evaluations favor our text consistency over ActionPlan and offline STMC. We further validate our method by showing strong single-prompt generation, spatial conditioning, and humanoid execution through a motion-tracking controller. Our main contributions are:

- We introduce streaming multi-track timeline control and develop **TimelineControl** to incorporate newly arriving instructions alongside ongoing actions, combining interval-aware multi-prompt conditioning with part-aware motion generation.
- We construct **TimelineMotion**, a dataset of motion sequences paired with overlapping instruction intervals and body-part annotations, supporting training and evaluation of streaming compositional generation.
- We demonstrate strong semantic alignment and temporal adherence on TimelineMotion and MTT, supported by baseline comparisons, ablations, and human evaluations. We further demonstrate spatial conditioning and humanoid motion execution.

2 Related Work

Text-to-motion generation. Text-to-motion methods represent data with raw motion (Tevet et al., 2023), continuous latents (Petrovich et al., 2022; Chen et al., 2023) or discrete tokens (Guo et al., 2022b; Zhang et al., 2023a; 2026a). Generation strategies include autoregression (Guo et al., 2022b; Zhang et al., 2023a), diffusion (Tevet et al., 2023; Zhang et al., 2024; Chen et al., 2023), masked token prediction (Guo et al., 2024), and masked autoregression (Meng et al., 2025; He et al., 2026). For sequential composition, TEACH (Athanasiou et al., 2022) conditions successive actions on motion history, while DoubleTake (Shafir et al., 2024) and FlowMDM (Barquero et al., 2024) synthesize transitions through coordinated denoising. In discrete motion-token space, M2D2M (Chi et al., 2024) combines independent and joint denoising to compose action sequences. For simultaneous actions, SINC (Athanasiou et al., 2023) constructs spatial compositions, while MultiAct (Sala et al., 2026) uses attention guidance to strengthen underrepresented actions in composite prompts. MotionMaster (Jiang et al., 2026) combines quantized motion tokens with a pretrained multimodal language model for compositional generation and body-part editing. STMC (Petrovich et al., 2024) brings temporal and spatial composition together by assigning text intervals to body regions and combining denoising predictions across tracks, given a complete timeline before synthesis. Temporally localized text conditioning also appears in video generation (Wu et al., 2025b; Shu et al., 2026). None of these methods support streaming multi-track timeline control.

Part-aware motion modeling. Part-aware representations use hierarchical upper- and lower-body streams for action composition (Ghosh et al., 2021) or body-part attention in the motion encoder (Zhong et al., 2023). PB-MLD (Liu et al., 2026) uses separate upper- and lower-body varia-

tional autoencoders with a joint Transformer denoiser. SALAD (Hong et al., 2025) preserves skeletal structure in its latent representation and combines temporal, skeletal, and text attention for generation and editing. Part-specific generation uses inter-part communication in ParCo (Zou et al., 2024), part-specific text with full-body refinement in LGTM (Sun et al., 2024), and part-aware text grounding with holistic-part fusion in ParTY (Heo et al., 2026). Other approaches condition motion on spatial and temporal descriptions (Zhang et al., 2023c) or compose atomic motions for open-vocabulary generation (Fan et al., 2026). FineMotion (Wu et al., 2025a) provides detailed motion annotations, while FrankenMotion (Li et al., 2026) pairs hierarchical temporal annotations with a diffusion model conditioned on pre-specified, temporally structured part descriptions. We use six anatomical latents in a causal autoencoder to support part-wise denoising during streaming generation.

Streaming and interactive motion generation. Causal tokenizers encode motion as discrete tokens (Jiang et al., 2025) or continuous latents, as in MotionStreamer (Xiao et al., 2025). Autoregressive diffusion models generate motion incrementally from motion history (Shi et al., 2024; Chen et al., 2024b). Diffusion Forcing (Chen et al., 2024a) introduces token-specific noise levels, which CMDM (Yu et al., 2026) applies to causal motion generation and FloodDiffusion (Cai et al., 2026) adapts to changing text. ActionPlan (Nazarenus et al., 2026) jointly generates motion and frame-level action plans with latent-specific noise levels, while PRISM (Ling et al., 2026) combines per-joint causal latents with flow matching and Self Forcing (Huang et al., 2025). We extend ActionPlan's joint formulation to prompts that arrive independently during generation and may overlap in time. Spatial conditioning complements language through trajectories and keyframes (Karunratanakul et al., 2023), joint-position constraints (Xie et al., 2024), and constraints for masked motion synthesis (Pinyoanuntapong et al., 2025). DartControl (Zhao et al., 2025) supports interactive motion control, while ARDY (Zhao et al., 2026) combines explicit root trajectories with body latents for online text and spatial conditioning. Related settings include online two-character interaction in Ready-to-React (Cen et al., 2025) and motion generation with physics-based character control in CLoSD (Tevet et al., 2025). In this paper, we mainly focus on online streaming timeline control and treat spatial targets as optional conditions on the timeline generator.

3 PRELIMINARIES

Prompt-conditioned rectified flow. We follow Nazarenus et al. (2026) for our problem setup. Given a text prompt ℓ, we generate a latent motion sequence $z = (z_1, \ldots, z_N)$, where z_n represents the n-th motion segment, $n \in [1, N]$. The generator is synthesizing N latent motions, then decoded to poses. The latent space is defined by a pretrained motion autoencoder that remains frozen during generator training. The generated latent sequence is later decoded back to motion (Chen et al., 2023; Xiao et al., 2025; Nazarenus et al., 2026). Let z^0 denote a clean latent sequence and $t \in [0, 1]$ the rectified-flow time (Liu et al., 2023). We construct the noisy latent z^t and target velocity $v^\star$ as

$$z^t = (1-t)z^0 + t\epsilon, \qquad v^\star = \epsilon - z^0, \qquad \epsilon \sim \mathcal{N}(0, \mathbf{I}). \tag{1}$$

Here, $t = 0$ corresponds to clean motion and $t = 1$ to Gaussian noise. Then, we use flow matching (Lipman et al., 2023) to train a velocity predictor v_θ, with parameters θ, conditioned on ℓ:

$$\mathcal{L}_{\text{flow}} = \mathbb{E}_{(z^0,\ell),\, t,\, \epsilon}\left[\left\| v_\theta(z^t, t, \ell) - v^\star \right\|_2^2\right]. \tag{2}$$

At inference, we initialize $z^1 \sim \mathcal{N}(0, \mathbf{I})$ and progressively transport it toward $t = 0$. Let $\hat{z}^t$ be the current sample and $t' < t$ the next flow time. We obtain the next latent $\hat{z}^{t'}$ with a first-order update:

$$\hat{z}^{t'} \leftarrow \hat{z}^t + (t' - t)\, v_\theta(\hat{z}^t, t, \ell), \qquad 0 \le t' < t \le 1. \tag{3}$$

When $t = 0$, the resulting latent sequence $\hat{z}^0$ is decoded into motion.

Online multi-track timeline control. We follow ActionPlan (Nazarenus et al., 2026) for the streaming setup, but we also assume that motion instructions *may arrive at arbitrary times, including when another instruction is not completed*. We aim to generate motion of each instruction in the correct interval. An instruction i is represented as $e_i = (\ell_i, [s_i, u_i))$, where ℓ_i is a textual prompt describing the action and s_i, u_i are start and end interval positions in the same window-local latent-index coordinates as n. During streaming, z belongs to the *current denoising window*, containing

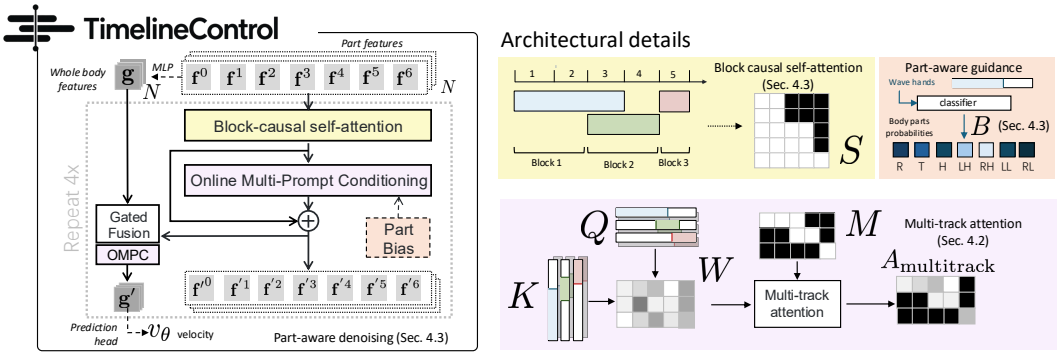

Figure 2: **TimelineControl.** We process part features and whole-body features separately (left). The part features undergo block-causal self-attention (yellow). Both use online multi-prompt conditioning (pink, OMPC) to allow for multi-track streaming inference. We add a part bias (orange) to enforce better mapping to body parts. Details (right) show examples of elements used in text.

previously emitted motion as fixed history and a continuation to be generated. Thus, N counts latent entries within this window, with index n. At a streaming step τ, the received instructions overlapping the current denoising window $\mathcal{W}_\tau$ form $C_\tau = \{e_i \mid s_i \leq \tau,\ [s_i, u_i) \cap \mathcal{W}_\tau \neq \varnothing\}$, where $\mathcal{W}_\tau$ is the current denoising window. The streaming time τ is expressed in the same window-local latent-index coordinates as n, s_i, and u_i; these coordinates are updated whenever the window advances. Let n_τ denote the first of the N latent entries to be generated in causal generation: $z_{<n_\tau}$ is retained as context, while $z_{n_\tau:N}$ is to be generated. The denoiser $v_\theta(z^t, t, \mathcal{C}_\tau)$ processes the whole window, but noise updates apply only to the continuation. As completed motion is emitted, it becomes fixed history and the window advances. When a new instruction arrives at time τ', we update the conditions to $\mathcal{C}_{\tau'}$ and regenerate the continuation $z_{n_{\tau'}:N}$ in the current window. We now describe how we enable the estimation of $v_\theta(z^t, t, \mathcal{C}_\tau)$.

4 Method

We introduce TimelineControl and show its architecture in Fig. 2. Our method is built on Action-Plan (Nazarenus et al., 2026), and trained with Diffusion Forcing (Chen et al., 2024a) to impose streaming generation. To allow multi-track conditioning, we use online multi-prompt conditioning, aligning motion features with available streamed prompts (Sec. 4.2). We also find that part-aware representations help disambiguate overlapping instructions, as different instructions may impact different body parts. We exploit this with a part-structured autoencoder (Sec. 4.1) and with part-aware and global features (Sec. 4.3) to guide individual regions while maintaining whole-body awareness.

4.1 Causal Part-Structured Autoencoder

Part-structured representation. For multi-track control, we assume that different instructions may impact different parts of the body. Hence, we benefit from part-structured representations in the latent space in which the motion generator is operating, similarly to related works (Heo et al., 2026; Hong et al., 2025). In practice, we partition body motion into six regions: torso, head, left and right arms, and left and right legs. Each entry z_n of the sequence introduced in Section 3 contains six body latents $b_n^k \in \mathbb{R}^8$, where $k \in 1, \ldots, 6$ indexes these regions. Following ARDY (Zhao et al., 2026), root motion bypasses the body encoder, keeping global position and heading explicit for spatial conditioning. The three position coordinates and the sine and cosine of heading over the same four frames form $r_n \in \mathbb{R}^{20}$. The motion representation is $z_n = [r_n; b_n^1; \ldots; b_n^6] \in \mathbb{R}^{68}$, with $z \in \mathbb{R}^{N \times 68}$.

Architecture. We define our autoencoder for causal streaming-oriented generation. The encoder and decoder each comprise three causal temporal-convolution stages, compressing the motion by a factor of 4. Within these stages, skeleton-graph message passing connects the torso stream to the head and each limb (Hong et al., 2025). Temporal convolutions capture each region's motion history, while graph communication provides context from connected regions without merging their repre-

sentations. The decoder additionally conditions on root yaw rate, horizontal velocity, and height derived from the explicit trajectory. Following MotionStreamer (Xiao et al., 2025), we train the autoencoder with reconstruction and KL losses, and add additional acceleration loss to reduce jitter, then freeze it during generator training. The detailed architecture is illustrated in Appendix B.1.

4.2 Online Multi-Prompt Conditioning

Interval-aware text attention. To implement the conditioning $v_\theta(z^t, t, \mathcal{C}_\tau)$, we must allow different instructions to influence different portions of the motion, with overlaps. We assume text cross attention for conditioning. Let $F = [\mathbf{f}_1; \ldots; \mathbf{f}_N]$ be the hidden features entering the cross-attention layer. These are computed from the noisy motion input z^t by the preceding layers. In the first layer, they are z_n. Each row $\mathbf{f}_n$ retains the temporal index n of the motion sequence and supplies the query for that position. Section 4.3 specifies how F evolves inside the network.

We encode each prompt $\ell_i \in \mathcal{C}_\tau$ independently with a frozen CLIP (Radford et al., 2021) text encoder and stack the resulting sentence embeddings h_i into $H = [h_1; \ldots; h_J]$, where $J = |\mathcal{C}_\tau|$ is the number of received instructions for a streaming window. For one attention head of dimension d, learned projections map F to queries $Q \in \mathbb{R}^{N\times d}$ and H to keys and values *of all the instructions, for all N steps*: $K, V \in \mathbb{R}^{J\times d}$. To align motion features with each instruction's temporal scope, we apply rotary position encoding (RoPE) (Su et al., 2024) to queries using their motion positions and rotary interval encoding (RoTE) to keys using each instruction's midpoint and duration, as in TIE (Shu et al., 2026). We then define the score matrix $W = \mathrm{RoPE}(Q)\,\mathrm{RoTE}(K)^\top/\sqrt{d} \in \mathbb{R}^{N\times J}$, which is the standard dot-product attention without softmax normalization. Each of its J columns corresponds to a distinct prompt, so every motion position is scored separately and aggregated in W. In short, we extract relevance scores *for each prompt in the input batch* with the current motion features.

Multi-track attention. Receiving an instruction does not make it applicable to the entire denoising window: it should contribute only between its start and end times. Positional encoding supplies timing bias but does not enforce this restriction. We hence define a visibility matrix $M \in \{0,1\}^{N\times J}$:

$$M_{ni} = \begin{cases} 1, & e_i \in \mathcal{C}_\tau \text{ and } s_i \le n < u_i, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

We show M in Fig. 2. At streaming time τ, this permits direct text contributions only from instructions whose intervals cover the queried motion. Hence, different rows can admit different subsets of the same instructions. Let $c_n = \sum_i M_{ni}$ count the visible instructions for the latent position n, and let the column vector $\mathbf{c}$ collect these counts. Whereas W contains pairwise compatibility scores, the final attention matrix $A_{\text{multitrack}} \in \mathbb{R}^{N\times J}$ determines how much each instruction contributes to each group of generated frames. Given an optional bias matrix $B \in \mathbb{R}^{N\times J}$, defined in Sec. 4.3, we have:

$$A_{\text{multitrack}} = \frac{M \odot \sigma(W + B)}{\sqrt{\max(1, \mathbf{c})}}. \tag{5}$$

Here, σ is sigmoid and $\odot$ is element-wise multiplication. We use sigmoid attention (Ramapuram et al., 2025) to preserve instruction importance. B adjusts part relevance, M excludes inapplicable instructions, and the normalization limits the text contribution in case of overlap. The resulting cross-attention is $\mathrm{CA}_{\text{multitrack}}(F, H; B, M) = A_{\text{multitrack}} V \in \mathbb{R}^{N\times d}$. Note in this operation, aggregation over timelines happens. The responses from multiple heads are concatenated and projected back to F dimensionality. Sec. 4.3 explains how we integrate this in TimelineControl.

4.3 Part-Aware Denoising

Parts and whole body. We now describe the flow of features F inside TimelineControl, and illustrate that in Fig. 2 (left). Multi-track control requires different body parts to follow overlapping instructions without losing whole-body coordination. For it, we join part-aware feature processing with a global stream mixing information across different parts. At the first layer, a multilayer perceptron initializes the whole-body features as $\mathbf{g}_n = \mathrm{MLP}([\mathbf{f}_n^0; \ldots; \mathbf{f}_n^6])$. This combines information across the root and six body regions while preserving their separate part-aware information.

Block-causal self-attention. Then, we process F with self-attention. During training, when instruction e_i arrives at streaming time $\tau = s_i$, it defines a generation block $\mathcal{B}_\tau = \{n \in \{1, \ldots, N\} :$

$\tau \le n < \tau^+\}$, where τ^+ is the next distinct arrival position within the window, or $N+1$ if there is none. Motion within a block is denoised jointly using the instructions available in $\mathcal{C}_\tau$, allowing bidirectional attention within the block and attention to all previous motion. Bidirectional attention coordinates motion within the block, so that earlier poses can anticipate later movements and form a temporally coherent action. For each query position $n \in \mathcal{B}_\tau$ and key position m, we add a mask to the self-attention logits, with $S^{(\tau)}_{nm} = 0$ if $m < \tau$ or $m \in \mathcal{B}_\tau$, and $S^{(\tau)}_{nm} = -\infty$ otherwise. We *only enforce this at training time* to improve learning. We show an example of S in Fig. 2.

Part-aware guidance. After self-attention, the features then receive the residual contribution of $\mathrm{CA}_{\mathrm{multitrack}}$ from Section 4.2, using the temporally updated features as queries and B^k as guidance. To guide each part toward relevant instructions, a text classifier is trained jointly with the denoiser using dataset body-part labels obtained through an LLM. See Appendix A for details. For all J received instructions, it predicts probabilities $\Pi \in [0,1]^{J\times 6}$, with one column per body part. For part k, the attention bias is $B^k_{ni} = \lambda_{\mathrm{r}}(\Pi_{ik} - \bar{\Pi}_i)$, where $\bar{\Pi}_i$ averages instruction i's six probabilities and λ_{r} controls guidance strength. Repeating these biases over the N motion positions gives $B^k \in \mathbb{R}^{N\times J}$. The root uses $B^0 = 0$. Classifier predictions are detached when biasing attention, while temporal applicability remains governed by M. See Fig. 2 for a visualization.

Gated part-global fusion We finally fuse part-aware processed features with the global branch. Given the part-aware states $\mathbf{f}'^k_n$, we fuse them into the whole-body state: $\bar{\mathbf{g}}_n = \mathbf{g}_n + \alpha_n \,\mathrm{Attn}\big(\mathbf{g}_n, \{\mathbf{f}'^k_n\}_{k=0}^6\big)$, where the learned gate $\alpha_n \in [0,1]$ controls the contribution and Attn is cross-attention. The whole-body state supplies the query, and the updated part-aware states supply keys and values. The fused features $\bar{\mathbf{g}}_n$ then pass through four Transformer layers with block-causal self-attention and online text attention. Whole-body text attention uses $B = 0$ and replaces the normalization of Section 4.2 with the row-wise sum of masked sigmoid weights (Ramapuram et al., 2025). After four stages, an output head maps the global features to the velocity.

4.4 Training Data

Since the most relevant timeline dataset MTT (Petrovich et al., 2024) has only testing data and there is no training data for multi-track streaming motion generation, we construct TimelineMotion, a dataset specific for the task of 44,944 motion sequences with overlapping text intervals, from HumanML3D (Guo et al., 2022a) using STMC-based motion composition (Petrovich et al., 2024). Each timeline uses source clips from a single HumanML3D split. We first use Qwen3-8B (Yang et al., 2025) to extract impacted body parts and motion attributes for HumanML3D captions. We then compose text labels in multiple tracks, using annotated labels for assessing motion compatibility. For instance, it is not feasible to simultaneously clap and point a finger with the left hand, because both actions impact that hand. Composition of different instructions can be sequential or parallel. The body part labels are also used to train the classifier in Section 4.3. We then generate motion associated to the composed instructions. We first initialize non-overlapping intervals from their HumanML3D source-motion crops. Within overlapping intervals, instead, we use STMC to perform body-part stitching and temporal composition to combine source motions according to the annotated body-part assignments. Composing motions with STMC from different clips can potentially introduce abrupt changes around the boundaries. We refine these transition regions to improve continuity, by applying a similar refinement pipeline to InterPose (Zhang et al., 2026b). In short, we check for discontinuities by thresholding joints' position and rotation differences across frames, and smooth detected discontinuities with interpolation. Finally, we also discard sequences which still present abrupt translations or excessive jitter. Further dataset construction details are provided in Appendix A.

5 Experiments

5.1 Experimental Setup

Datasets and baselines. We evaluate on the test sets of timeline dataset TimelineMotion and MTT (Petrovich et al., 2024), the most relevant dataset with multi-track instructions available. We further evaluate the performance on non-streaming text dataset HumanML3D (Guo et al., 2022a). Note *no baseline natively supports streaming multi-track conditioning*. For streaming baselines, we compare with ActionPlan (Nazarenus et al., 2026), which also serves as our backbone, Motion-

Table 1: **Benchmarks with streaming instructions.** We outperform all the baselines on fine-grained evaluation about sequence alignment, body parts, motion quality and temporal adherence with the sequence, on both TimelineMotion and MTT.

	Method	Stream	Sequence alignment					Body-part alignment			Motion quality			Temporal adherence	
			R@1_{32} ↑	R@2_{32} ↑	R@3_{32} ↑	M2T ↑	M2M ↑	R@3_{32} ↑	M2T ↑	M2M ↑	FID ↓	FID+ ↓	Trans ↓	AP ↑	AUC ↑
TimelineMotion	STMC	✗	0.329	0.452	0.525	0.627	0.630	0.353	0.538	0.633	0.055	0.228	0.92	0.803	0.732
	FrankenMotion	✗	0.106	0.178	0.232	0.538	0.551	0.270	0.526	0.568	0.159	0.246	1.01	0.639	0.545
	ActionPlan	✓	0.248	0.344	0.405	0.585	0.585	0.303	0.547	0.589	0.274	0.330	2.16	0.733	0.653
	ActionPlan*	✓	0.320	0.436	0.502	0.622	0.627	0.347	0.546	0.634	0.055	0.209	0.92	0.784	0.711
	MotionStreamer*	✓	0.222	0.321	0.383	0.585	0.592	0.304	0.541	0.608	0.049	0.218	1.09	0.734	0.653
	DartControl*	✓	0.263	0.381	0.454	0.604	0.575	0.341	0.541	0.591	0.244	0.368	1.27	0.793	0.728
	ARDY	✓	0.165	0.246	0.303	0.562	0.575	0.275	0.524	0.573	0.340	0.383	**0.90**	0.697	0.607
	TimelineControl	✓	**0.406**	**0.531**	**0.603**	**0.653**	**0.656**	**0.374**	**0.553**	**0.659**	**0.017**	**0.192**	1.31	**0.832**	**0.772**
MTT	STMC	✗	0.330	0.491	0.566	0.678	0.668	0.421	0.560	**0.654**	–	**0.476**	**0.49**	0.769	0.729
	FrankenMotion	✗	0.140	0.234	0.309	0.572	0.577	0.370	0.549	0.578	–	0.543	1.08	0.629	0.579
	ActionPlan	✓	0.296	0.429	0.514	0.645	0.628	0.399	0.551	0.615	–	0.589	1.01	0.754	0.716
	ActionPlan*	✓	0.333	0.497	0.574	0.678	0.656	0.421	0.560	0.633	–	0.535	0.64	0.778	0.742
	MotionStreamer*	✓	0.215	0.357	0.432	0.612	0.615	0.358	0.543	0.612	–	0.489	0.90	0.710	0.670
	DartControl*	✓	0.311	0.449	0.543	0.651	0.614	0.426	0.567	0.594	–	0.604	1.16	0.792	0.764
	ARDY	✓	0.207	0.321	0.369	0.598	0.601	0.320	0.521	0.583	–	0.687	0.57	0.674	0.621
	TimelineControl	✓	**0.368**	**0.543**	**0.628**	**0.696**	**0.682**	**0.433**	**0.569**	**0.654**	–	0.478	0.68	**0.809**	**0.777**

Methods with ∗ are trained on both HumanML3D and TimelineMotion

Streamer (Xiao et al., 2025), DartControl (Zhao et al., 2025), and ARDY (Zhao et al., 2026). Our model and all baselines marked with * are trained jointly on TimelineMotion and HumanML3D. Streaming methods receive each prompt at its annotated start time. For baselines without native conditioning on multiple text intervals, we update a single caption at every instruction start and end: one active prompt is passed unchanged, and multiple active prompts are joined with *"while"*. Retrained baselines also use *"while"* to join overlapping descriptions within each training window. STMC (Petrovich et al., 2024) and FrankenMotion (Li et al., 2026) are evaluated offline with the complete timeline with explicit body-part labels and body text intervals, respectively. FrankenMotion additionally receives a sequence-level caption formed from the original event descriptions. On HumanML3D, it receives only the sequence-level caption.

Evaluation metrics. We use TMR-based R-Precision, text-motion similarity (M2T), and similarity to source (M2M) to assess sequence and body-part alignment following (Petrovich et al., 2023; Li et al., 2026). We use FID (Heusel et al., 2017) to compare distributions of complete generated and reference motions, also including FID+ following STMC (Petrovich et al., 2024). Transition distance (Trans) (Petrovich et al., 2024) assesses continuity at instruction boundaries and it is reported in centimeters. Autoencoder reconstruction is evaluated with FID and mean per-joint position error (MPJPE, in millimeters) following Xiao et al. (2025). We also measure if generated motions express each instruction within its requested interval. To do so, we score each instruction against sliding motion windows across the complete timeline using TMR embeddings. We then quantify with average precision (AP) and area-under-the-curve (AUC) if windows centered inside the requested instruction interval receive higher scores than those outside. More details in Appendix C.

Training/inference. We train jointly on TimelineMotion and HumanML3D and each timeline window contains up to 78 prediction latents and 8 clean history latents. On a NVIDIA RTX 6000 Ada at batch size one, sampling takes 280 ms for the first motion token and 35 ms per subsequent token. We also use the plan branch of ActionPlan unchanged from theirs. Full details are in App. B.

5.2 Comparison with baselines

Quantitative results. As presented in Table 1, TimelineControl achieves state-of-the-art performance on the two benchmarks supporting streaming inferences. Methods supporting online streaming inference are indicated in the Stream column. Notably, TimelineControl also outperforms offline STMC on TimelineMotion, despite STMC having access to the complete timeline and being used to compose our training data. For example, FID decreases from 0.055 to 0.017. This shows that training on STMC-composed data does not limit TimelineControl to the performance of the composition procedure. Our data construction strategy alone helps with the task. Indeed, retraining ActionPlan jointly on TimelineMotion and HumanML3D (ActionPlan* in Table 1) improves almost all the metrics on the two timeline benchmarks. However, our gains in sequence alignment and body-part alignment persist against retrained baselines, showing the impact of our

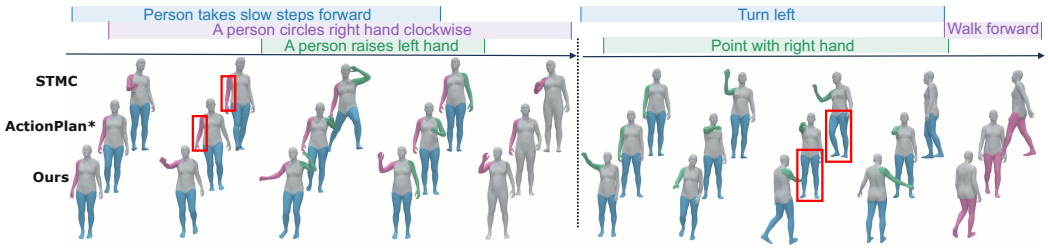

Figure 3: **Qualitative evaluation.** We compare STMC, ActionPlan* and TimelineControl on TimelineMotion (left) and MTT (right) dataset. Red box marks the failure part. TimelineControl is the only one not failing by ignoring instructions or confusing them.

contributions. Appendix D.4 further compares with ActionPlan, our backbone, with separately encoded instructions and interval masks.

We include HumanML3D in our evaluation to assess whether our model preserves performance in the standard single-prompt setting. As shown in Table 2, TimelineControl achieves the highest retrieval performance (0.874 vs 0.830) and the second-best FID among the evaluated baselines (0.027 vs 0.022). These results suggest that its improvements on timeline benchmarks do not significantly impact text-to-motion generation.

Qualitative results. We show qualitatives in Figure 3, comparing with ActionPlan* and STMC, the best baselines. In presence of overlapping instructions, *none of the baselines is able to correctly render the motion*. In particular, on the left, the instruction is fully ignored by ActionPlan*, while STMC renders it only in the very last frame, when there is no overlapping. On the right example, instead, ActionPlan ignores the instruction, while STMC mixes information in different labels and turn the human right rather than left. TimelineControl is instead able to correctly generate the desired motion, and to map each instruction into the correct interval. More examples are in Appendix E.1 and also the supplementary video.

Table 2: **Results on non-streaming generation.** We compare with motion generation methods on HumanML3D dataset.

	Method	R@1_{32} ↑	R@2_{32} ↑	R@3_{32} ↑	M2T ↑	M2M ↑	FID ↓
HumanML3D	STMC	0.611	0.756	0.815	<u>0.741</u>	0.697	0.224
	FrankenMotion	0.432	0.566	0.643	0.677	0.689	0.079
	ActionPlan	0.618	0.768	0.829	0.724	**0.769**	**0.022**
	ActionPlan*	<u>0.645</u>	<u>0.778</u>	<u>0.830</u>	0.734	<u>0.765</u>	0.044
	MotionStreamer*	0.568	0.706	0.772	0.726	0.742	0.047
	DartControl*	0.536	0.699	0.775	0.711	0.657	0.345
	ARDY	0.413	0.566	0.649	0.657	0.649	0.349
	TimelineControl	**0.687**	**0.827**	**0.874**	**0.769**	0.753	<u>0.027</u>

User study. We conduct a user study to complement the evaluation in three aspects: text alignment, motion quality and temporal adherence. TimelineControl consistently outperforms other baselines. In particular, TimelineControl wins 66.2% and 72.1% on text-motion consistency against STMC and ActionPlan*, respectively. More detailed results and the study design can be found in Appendix C.5.

Autoencoder design. Table 3 positions our autoencoder design with published alternatives. We compare with 16- and 48-dimensional temporal autoencoders (TAE, 16-D and 48-D respectively) (Xiao et al., 2025), SALAD (Hong et al., 2025), and our part-structured autoencoder. We evaluate both reconstruction quality and downstream generation effectiveness. All the 48-D autoencoders use 48-D body latents with 20-dimensional explicit root features (68 dimensions total). At the same 48-D body latent size, our part-aware autoencoder achieves comparable reconstruction FID on TimelineMotion while reducing generation FID from 0.040 to 0.017 and improving retrieval on both datasets. This result supports preserving anatomical structure in the features supplied to the denoising network. Finally, we show that removing our skeleton graph substantially degrades metrics (ours w/o sk.graph, Table 3), further justifying our design choices.

5.3 Ablation Studies

Table 4 evaluates our architectural choices on TimelineMotion, to assess streaming performance, and on HumanML3D, for evaluating loss of non-streaming capabilities. Further ablations on attention normalization, self-conditioning, and training-window sampling can be found in Appendix D.

Table 3: **Autoencoder benchmarks.** Quantitative comparison of different autoencoder designs in reconstruction and generation in both HumanML3D and TimelineMotion.

Autoencoder	Dims	Params (M)	*HumanML3D*				*TimelineMotion*						
			Reconstruction		Generation		Reconstruction		Generation			Temporal adherence	
			FID ↓	MPJPE ↓	FID ↓	R@3_{32} ↑	FID ↓	MPJPE ↓	FID ↓	FID+ ↓	R@3_{32} ↑	AP ↑	AUC ↑
TAE (Xiao et al., 2025)	16	76.20	0.0028	21.9	0.047	0.833	0.0020	21.7	0.051	0.203	0.533	0.823	0.755
TAE (Xiao et al., 2025)	48	79.42	0.0005	3.19	0.049	0.851	**0.0002**	3.77	0.040	0.197	0.576	0.827	0.762
SALAD (Hong et al., 2025)	48	0.51	0.0034	24.1	0.030	0.858	0.0036	27.7	0.023	0.208	0.575	0.820	0.755
Ours w/o sk.graph	48	4.80	0.0012	7.46	0.038	0.851	0.0007	8.27	0.025	0.209	0.600	**0.832**	0.767
Ours	48	5.59	**0.0004**	**3.09**	**0.027**	**0.874**	**0.0002**	**3.58**	**0.017**	**0.192**	**0.603**	**0.832**	**0.772**

Table 4: **Architectural ablations.** Ablation of temporal conditioning and part-aware denoising on HumanML3D and TimelineMotion. Each variant removes the indicated component from TimelineControl. All our contributions together yield the best performance.

Setup	*HumanML3D*			*TimelineMotion*								
				Sequence alignment		Body-part alignment		Motion quality			Temporal adherence	
	R@3_{32} ↑	M2T ↑	FID ↓	R@3_{32} ↑	M2T ↑	R@3_{32} ↑	M2T ↑	FID ↓	FID+ ↓	Trans ↓	AP ↑	AUC ↑
Full model (Ours)	**0.874**	**0.769**	**0.027**	**0.603**	**0.653**	**0.374**	**0.553**	**0.017**	**0.192**	1.31	**0.832**	**0.772**
RoTE only	0.830	0.753	0.044	0.586	0.649	0.370	0.552	0.028	0.197	1.28	0.820	0.752
w/o RoTE	0.820	0.748	0.061	0.572	0.648	0.372	0.550	0.037	0.195	**1.09**	0.801	0.734
w/o part-aware bias B	0.850	0.764	0.038	0.591	0.650	0.357	0.547	0.025	0.193	1.20	0.830	0.769
w/o part branch	0.850	0.763	0.042	0.588	0.649	0.350	0.545	0.030	0.198	1.26	0.829	0.767

We ablate conditioning components in Table 4 (top). We first remove the M and S masks and rely naively on RoTE only for multi-timeline conditioning (RoTE only). This reduces retrieval while increasing FID on both datasets, demonstrating the need of our attention masking for streaming multi-track motion generation. Removing RoTE (*w/o RoTE*) leaves the masks intact but removes temporal sequence information from the encoded text. This has significant effects, with R@3 decreasing from **0.603** to 0.572 on TimelineMotion. This shows that to benefit from our conditioning mechanism, we need appropriate encoding of the timeline arrival times and order.

We now ablate components of our part-aware denoising algorithm in Table 4 (bottom). We first remove the bias B from the multi-prompt conditioning (*w/o part-aware bias* B). Most metrics decrease, showing that our representation exploits correctly the attention on multiple body parts, which B enhances. We also propose a *w/o part branch* setup that only uses the global state $\mathbf{g}$ to predict velocity. Note that $\mathbf{g}$ still receives information on the prompts with online multi-prompt conditioning. From the results, we considerably increase FID (**0.027**→0.042 on HumanML3D, **0.017**→0.030 on TimelineMotion), showing that the task highly benefits from part-aware processing.

5.4 Applications

We propose some applications built on TimelineControl, to demonstrate the versatility of our method. In Figure 1(b), we show spatial conditioning, with a strategy like in Zhao et al. (2026). We can use the obtained model for position control on animation and robotics. We also deployed TimelineControl on a humanoid robot, executing motions in real time using SONIC (Luo et al., 2026) as motion tracker (Figure 1(c)). Additional results and details are in Appendices E and F.

6 Conclusion

We introduced streaming multi-track timeline control, where independently timed instructions arrive during motion generation and may overlap. Our framework, TimelineControl, combines interval-aware text conditioning with part-aware denoising to coordinate concurrent actions. We also constructed TimelineMotion to support training and evaluation in this setting. Experiments show improved performance over the evaluated baselines on TimelineMotion and MTT. We further demonstrated spatial conditioning and execution of generated motions on a humanoid robot. Future work will explore online revision and cancellation of instructions, as well as resolving conflicting requests while preserving motion continuity.

AI USE STATEMENT

Generative AI tools assisted with manuscript polishing, code implementation and debugging, and body-part annotation of motion descriptions. The annotation procedure and label review are described in Section 4.4 and Appendix A.

REPRODUCIBILITY STATEMENT

The model architecture and conditioning mechanisms are given in Sections 4.1–4.3, with implementation, training-objective, and streaming-inference details in Appendix B.2. The TimelineMotion construction procedure is given in Section 4.4 and Appendix A, including the annotation, compatibility, transition-refinement, and filtering rules. The experimental setup and evaluation metrics are given in Section 5.1 and Appendix C. Additional training strategies and ablations are provided in Appendix D. Our code, data and models will be released upon acceptance.

REFERENCES

Nikos Athanasiou, Mathis Petrovich, Michael J. Black, and Gül Varol. TEACH: Temporal action composition for 3D humans. In *3DV*, 2022.

Nikos Athanasiou, Mathis Petrovich, Michael J. Black, and Gül Varol. SINC: Spatial composition of 3D human motions for simultaneous action generation. In *ICCV*, 2023.

German Barquero, Sergio Escalera, and Cristina Palmero. Seamless human motion composition with blended positional encodings. In *CVPR*, 2024.

Yiyi Cai, Yuhan Wu, Kunhang Li, You Zhou, Bo Zheng, and Haiyang Liu. FloodDiffusion: Tailored diffusion forcing for streaming motion generation. In *CVPR*, 2026.

Zhi Cen, Huaijin Pi, Sida Peng, Qing Shuai, Yujun Shen, Hujun Bao, Xiaowei Zhou, and Ruizhen Hu. Ready-to-React: Online reaction policy for two-character interaction generation. In *ICLR*, 2025.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion Forcing: Next-token prediction meets full-sequence diffusion. In *NeurIPS*, 2024a.

Rui Chen, Mingyi Shi, Shaoli Huang, Ping Tan, Taku Komura, and Xuelin Chen. Taming diffusion probabilistic models for character control. In *ACM SIGGRAPH Conf*, 2024b.

Xin Chen, Biao Jiang, Wen Liu, Zilong Huang, Bin Fu, Tao Chen, and Gang Yu. Executing your commands via motion diffusion in latent space. In *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 *ECCV*, 2024.

Ke Fan, Jiangning Zhang, Ran Yi, Jingyu Gong, Yabiao Wang, Yating Wang, Xin Tan, Chengjie Wang, and Lizhuang Ma. Open the motion door: Atomic motion decomposition and recomposition for open-vocabulary motion generation. In *CVPR*, 2026.

Anindita Ghosh, Noshaba Cheema, Cennet Oguz, Christian Theobalt, and Philipp Slusallek. Synthesis of compositional animations from textual descriptions. In *ICCV*, 2021.

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 *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 *ECCV*, 2022b.

Chuan Guo, Yuxuan Mu, Muhammad Gohar Javed, Sen Wang, and Li Cheng. MoMask: Generative masked modeling of 3D human motions. In *CVPR*, 2024.

Yannan He, Garvita Tiwari, Xiaohan Zhang, Pankaj Bora, Tolga Birdal, Jan Eric Lenssen, and Gerard Pons-Moll. MoLingo: Motion–language alignment for text-to-human motion generation. In *CVPR*, 2026.

KunHo Heo, SuYeon Kim, Yonghyun Gwon, Youngbin Kim, and MyeongAh Cho. ParTY: Part-guidance for expressive text-to-motion synthesis. In *CVPR*, 2026.

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. In *NeurIPS*, 2017.

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 *CVPR*, 2025.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self Forcing: Bridging the train-test gap in autoregressive video diffusion. In *NeurIPS*, 2025.

Biao Jiang, Xin Chen, Ailing Zeng, Xinru Sun, Fukun Yin, Xianfang Zeng, Xuanyang Zhang, Gang Yu, and Tao Chen. Causal motion tokenizer for streaming motion generation. In *ICCVW*, 2025.

Nan Jiang, Yunhao Li, Lexi Pang, Zimo He, Siyuan Huang, and Yixin Zhu. MotionMaster: Generalizable text-driven motion generation and editing. In *CVPR*, 2026.

Korrawe Karunratanakul, Konpat Preechakul, Supasorn Suwajanakorn, and Siyu Tang. Guided motion diffusion for controllable human motion synthesis. In *ICCV*, 2023.

Chuqiao Li, Xianghui Xie, Yong Cao, Andreas Geiger, and Gerard Pons-Moll. FrankenMotion: Part-level human motion generation and composition. In *CVPR*, 2026.

Zeyu Ling, Qing Shuai, Teng Zhang, Shiyang Li, Bo Han, and Changqing Zou. PRISM: Streaming human motion generation with per-joint latent decomposition. *arXiv*, 2026.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *ICLR*, 2023.

Jiahao Liu, Huiyang Xiao, Yuhang Li, Yuning Qiu, and Guoxu Zhou. PB-MLD: Part-based motion latent diffusion model. *Image and Vision Computing*, 2026.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023.

Zhengyi Luo, Ye Yuan, Tingwu Wang, Chenran Li, Fernando Castañeda, Sirui Chen, Zi-Ang Cao, Jiefeng Li, David Minor, Qingwei Ben, Jinhyung Park, David Sami, Zi Wang, Xingye Da, Runyu Ding, Cyrus Hogg, Lina Song, Edy Lim, Eugene Jeong, Tairan He, Haoru Xue, Wenli Xiao, Simon Yuen, Jan Kautz, Yan Chang, Umar Iqbal, Linxi "Jim" Fan, and Yuke Zhu. SONIC: Supersizing motion tracking for natural humanoid whole-body control. *Science Robotics*, 2026.

Zichong Meng, Yiming Xie, Xiaogang Peng, Zeyu Han, and Huaizu Jiang. Rethinking diffusion for text-driven human motion generation: Redundant representations, evaluation, and masked autoregression. In *CVPR*, 2025.

Eric Nazarenus, Chuqiao Li, Yannan He, Xianghui Xie, Jan Eric Lenssen, and Gerard Pons-Moll. ActionPlan: Future-aware streaming motion synthesis via frame-level action planning. *arXiv*, 2026.

Mathis Petrovich, Michael J. Black, and Gül Varol. TEMOS: Generating diverse human motions from textual descriptions. In *ECCV*, 2022.

Mathis Petrovich, Michael J. Black, and Gül Varol. TMR: Text-to-motion retrieval using contrastive 3D human motion synthesis. In *ICCV*, 2023.

Mathis Petrovich, Or Litany, Umar Iqbal, Michael J. Black, Gül Varol, Xue Bin Peng, and Davis Rempe. Multi-track timeline control for text-driven 3D human motion generation. In *CVPRW*, 2024.

Ekkasit Pinyoanuntapong, Muhammad Saleem, Korrawe Karunratanakul, Pu Wang, Hongfei Xue, Chen Chen, Chuan Guo, Junli Cao, Jian Ren, and Sergey Tulyakov. MaskControl: Spatio-temporal control for masked motion synthesis. In *ICCV*, 2025.

Abhinanda R. Punnakkal, Arjun Chandrasekaran, Nikos Athanasiou, Alejandra Quiros-Ramirez, and Michael J. Black. BABEL: Bodies, action and behavior with english labels. In *CVPR*, 2021.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

Jason Ramapuram, Federico Danieli, Eeshan Dhekane, Floris Weers, Dan Busbridge, Pierre Ablin, Tatiana Likhomanenko, Jagrit Digani, Zijin Gu, Amitis Shidani, and Russ Webb. Theory, analysis, and best practices for sigmoid self-attention. In *ICLR*, 2025.

Nathan Sala, Ofir Abramovich, Ariel Shamir, Daniel Cohen-Or, Andreas Aristidou, and Sigal Raab. MultiAct: Text-to-motion generation from composite text via tailored attention guidance. In *ACM SIGGRAPH Conf*, 2026.

Yoni Shafir, Guy Tevet, Roy Kapon, and Amit Haim Bermano. Human motion diffusion as a generative prior. In *ICLR*, 2024.

Yi Shi, Jingbo Wang, Xuekun Jiang, Bingkun Lin, Bo Dai, and Xue Bin Peng. Interactive character control with auto-regressive motion diffusion models. *ACM TOG*, 2024.

Zhilei Shu, Shangwen Zhu, Zihang Liang, Xiaofan Li, Qianyu Peng, Xinyu Cui, Bo Ye, Yiming Li, Fan Cheng, Jian Zhao, Yang Cao, Zheng-Jun Zha, and Ruili Feng. TIE: Time interval encoding for video generation over events. *arXiv*, 2026.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 2024.

Haowen Sun, Ruikun Zheng, Haibin Huang, Chongyang Ma, Hui Huang, and Ruizhen Hu. LGTM: Local-to-global text-driven human motion diffusion model. In *ACM SIGGRAPH Conf*, 2024.

Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-Or, and Amit Haim Bermano. Human motion diffusion model. In *ICLR*, 2023.

Guy Tevet, Sigal Raab, Setareh Cohan, Daniele Reda, Zhengyi Luo, Xue Bin Peng, Amit Haim Bermano, and Michiel van de Panne. CLoSD: Closing the loop between simulation and diffusion for multi-task character control. In *ICLR*, 2025.

Bizhu Wu, Jinheng Xie, Meidan Ding, Zhe Kong, Jianfeng Ren, Ruibin Bai, Rong Qu, and Linlin Shen. FineMotion: A dataset and benchmark with both spatial and temporal annotation for fine-grained motion generation and editing. In *ICCV*, 2025a.

Ziyi Wu, Aliaksandr Siarohin, Willi Menapace, Ivan Skorokhodov, Yuwei Fang, Varnith Chordia, Igor Gilitschenski, and Sergey Tulyakov. Mind the time: Temporally-controlled multi-event video generation. In *CVPR*, 2025b.

Lixing Xiao, Shunlin Lu, Huaijin Pi, Ke Fan, Liang Pan, Yueer Zhou, Ziyong Feng, Xiaowei Zhou, Sida Peng, and Jingbo Wang. MotionStreamer: Streaming motion generation via diffusion-based autoregressive model in causal latent space. In *ICCV*, 2025.

Yiming Xie, Varun Jampani, Lei Zhong, Deqing Sun, and Huaizu Jiang. OmniControl: Control any joint at any time for human motion generation. In *ICLR*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang

Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv*, 2025.

Qing Yu, Akihisa Watanabe, and Kent Fujiwara. Causal motion diffusion models for autoregressive motion generation. In *CVPR*, 2026.

Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Yong Zhang, Hongwei Zhao, Hongtao Lu, Xi Shen, and Ying Shan. Generating human motion from textual descriptions with discrete representations. In *CVPR*, 2023a.

Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Xi Shen, Xiaojian Shen, Hehe Fan, and Yi Yang. Text-to-motion generation with discrete representations and large language models. *IEEE TPAMI*, 2026a.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023b.

Mingyuan Zhang, Huirong Li, Zhongang Cai, Jiawei Ren, Lei Yang, and Ziwei Liu. FineMoGen: Fine-grained spatio-temporal motion generation and editing. In *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 TPAMI*, 2024.

Yangsong Zhang, Abdul Ahad Butt, Gül Varol, and Ivan Laptev. InterPose: Learning to generate human–object interactions from large-scale web videos. In *3DV*, 2026b.

Kaifeng Zhao, Gen Li, and Siyu Tang. DartControl: A diffusion-based autoregressive motion model for real-time text-driven motion control. In *ICLR*, 2025.

Kaifeng Zhao, Mathis Petrovich, Haotian Zhang, Tingwu Wang, Siyu Tang, and Davis Rempe. ARDY: Autoregressive diffusion with hybrid representation for interactive human motion generation. *ACM TOG*, 2026.

Chongyang Zhong, Lei Hu, Zihao Zhang, and Shihong Xia. AttT2M: Text-driven human motion generation with multi-perspective attention mechanism. In *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 *ECCV*, 2024.

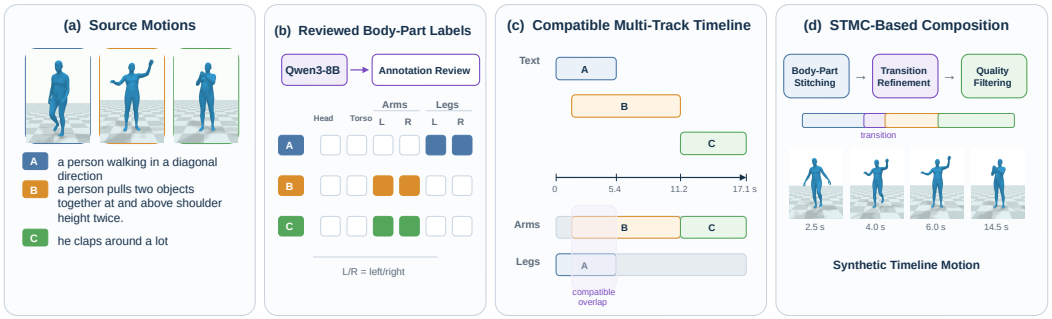

Figure 4: Construction of TimelineMotion. (a) HumanML3D source motions and captions. (b) Body-part annotations. (c) Compatible instructions arranged into multi-track timelines. (d) STMC-based body-part composition, transition refinement, and quality filtering.

A TimelineMotion Construction Details

Figure 4 summarizes the TimelineMotion construction pipeline; below, we provide the annotation and filtering details omitted from Section 4.4.

A.1 Body-Part Annotation

Qwen3-8B (Yang et al., 2025) annotates captions from HumanML3D dataset with body-part labels and motion attributes. The primary labels specify the anatomical regions used for motion composition, part supervision, and body-part evaluation. Primary labels that refer to the upper, lower, or full body are expanded into the corresponding anatomical regions when explicit regional labels are absent.

The annotation prompt below uses `{text}` for the input motion caption.

```
You are an expert motion-text parser.
Parse the input action text into one JSON
object with only these fields:
- primary_parts: list[str] from
  ["torso", "head", "left_arm",
   "right_arm", "left_leg", "right_leg",
   "upper_body", "lower_body", "full_body"]
- secondary_parts: list[str] with same
  allowed labels
- coarse_parts: list[str] from
  ["upper_body", "lower_body", "full_body"]
- locomotion: bool
- stance_change: bool
- whole_body: bool
- dynamic_level: one of
  ["low", "medium", "high", "unknown"]
- stance_type: one of
  ["standing", "sitting", "crouching",
   "lying", "unknown"]

Rules:
- Do not output explanation.
- Output JSON only.
- If text mentions hand/arm gestures
  (wave, clap, throw, catch, point, raise
  hands, etc.), include explicit arm labels
  in primary_parts.
- If side is unspecified for arm gestures,
  include both "left_arm" and "right_arm".
- If text mentions leg/foot/lower-body
```

```
  actions (kick, step, walk, run, jump,
  squat, etc.), include explicit leg labels
  in primary_parts.
- If side is unspecified for leg/foot
  actions, include both "left_leg" and
  "right_leg".
- Do not use only coarse labels when
  explicit limbs are inferable from text
  (same principle for leg/foot actions).

Input text:
{text}
```

A.2 Timeline Construction

We randomly sample source clips from the same HumanML3D split and consider sequential or overlapping temporal arrangements. For overlapping clips, we exclude pairs that require the same body regions, contain conflicting root motion or locomotion, involve incompatible poses or large stance changes, or contain full-body actions. The primary body-part labels determine the source motion assigned to each anatomical region during STMC composition, as described in Section 4.4.

We refine transition and overlap regions with interpolation to improve continuity while retaining the surrounding source motion. Following InterPose (Zhang et al., 2026b), refinement uses spherical interpolation for joint rotations and cubic Hermite interpolation where root trajectories require smoothing. We additionally identify composition artifacts around timeline boundaries from decoded joint motion. We first express joint positions relative to the pelvis and compute the magnitude of their temporal second differences, averaged over joints. For each composition boundary, we take the maximum value within two frames of the boundary as the boundary peak. As a local reference, we compute the 95th percentile within 12 frames on either side of the boundary, excluding the three frames closest to the boundary on each side. We reject a sequence when the boundary peak is at least 0.03, exceeds the local reference by at least 0.025, and is at least $1.5\times$ the local reference. We also remove sequences containing non-finite motion values.

B Objectives and Implementation Details

B.1 Causal Part-Structured Autoencoder

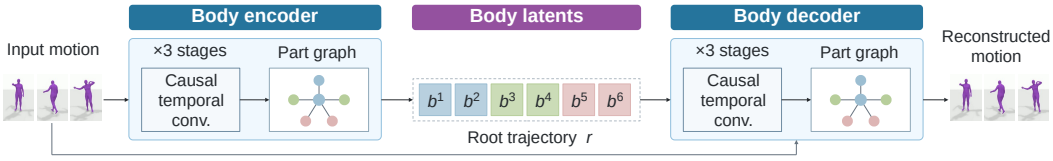

Figure 5: Causal part-structured autoencoder. Causal temporal convolutions and skeleton-graph message passing encode and reconstruct six anatomical body regions. The root trajectory bypasses the body encoder and conditions the decoder.

Figure 5 illustrates the architecture described in Section 4.1. Body features comprise 21 non-root joint positions, 22 joint rotations in 6D, joint velocities, and four foot-contact indicators. Positions retain world-space height but are horizontally centered at the pelvis. Rotations and velocities use world coordinates. The torso includes root rotation and velocity, and each leg includes its foot-contact indicators. Following MotionStreamer (Xiao et al., 2025), the autoencoder objective is

$$\mathcal{L}_{\mathrm{AE}} = \mathcal{L}_{\mathrm{rec}} + \mathcal{L}_{\mathrm{KL}} + \mathcal{L}_{\mathrm{acc}}. \tag{6}$$

Here, $\mathcal{L}_{\mathrm{rec}}$ is the Gaussian negative log-likelihood reconstruction loss and $\mathcal{L}_{\mathrm{KL}}$ regularizes the latent posterior. The temporal loss $\mathcal{L}_{\mathrm{acc}}$ matches second-order differences.

B.2 Denoiser Architecture and Training Settings

The global branch has four stages of four Transformer layers, with width 1024, feed-forward width 4096, and 16 attention heads. The input token encoder has one attention layer with width 128 and four heads. The classifier-predicted attention bias uses $\lambda_{\mathrm{r}} = 0.25$. The classifier takes projected prompt embeddings as input. Body-part tokens use centered relevance probabilities to form B^k, while the root uses $B^0 = 0$. We stop gradients at the classifier input and through the probabilities used to form B^k in Section 4.3. The generator learning rate is 1e-4, with weight decay 0.01. HumanML3D examples are capped at 78 latents with a latent rate of 7.5 fps. RoTE uses frequencies $\omega_j = 10000^{-(j-1)/D_{\mathrm{rot}}}$ for $j = 1, \ldots, D_{\mathrm{rot}}$, where D_{rot} is the number of rotary pairs. A common timestamp scale $\gamma = 4$ is used for motion queries and interval centers and radii.

B.3 Backbone Objectives and details

Following ActionPlan (Nazarenus et al., 2026), we construct action-plan targets from atomic action annotations in BABEL (Punnakkal et al., 2021), aligned to the source motion clips and transferred to the composed timelines. We average the CLIP embeddings of distinct active action labels at each frame, temporally smooth the features, and compress them with a frozen autoencoder. We apply the interpolation and velocity target in Eq. (1) to both streams. Each motion latent uses a noise level t_n^{m}, while the action-plan stream shares one level t^{p} per training example.

The weighted flow loss is computed over scalar targets in the minibatch and normalized by the total weight jointly across the motion and action-plan streams. Valid motion targets use importance weights that compensate for their nonuniform noise-level distribution. Valid action-plan targets have unit weight. The shared action-plan noise level is sampled uniformly. Motion supervision excludes the fixed prefix, while available action-plan targets at prefix positions remain supervised. Padding, positions beyond the supervised horizon, and targets replaced by detached self-conditioned predictions have zero weight.

The main generation objective is

$$\mathcal{L}_{\mathrm{gen}} = \mathcal{L}_{\mathrm{flow}} + 0.2\mathcal{L}_{\mathrm{part}} + 0.02\mathcal{L}_{\mathrm{cls}} + \mathcal{L}_{\mathrm{root}}. \tag{7}$$

Here, $\mathcal{L}_{\mathrm{part}}$ is the mean squared error of the auxiliary body-latent flow-velocity prediction. Let $Y \in \{0,1\}^{J\times 6}$ denote the annotated body-part labels, with the same instruction and part ordering as Π. $\mathcal{L}_{\mathrm{cls}}$ uses binary cross-entropy with Y. The root loss $\mathcal{L}_{\mathrm{root}}$ applies smooth L1 supervision to frame-to-frame changes in horizontal position and heading.

B.4 Streaming Inference

Following ActionPlan (Nazarenus et al., 2026), each sampling window retains up to P generated motion latents as a fixed prefix and initializes future motion and all action-plan latents with Gaussian noise. The action-plan stream is first denoised jointly with the first motion token, while later motion positions remain at higher noise levels. Once the action plan is complete, its latents remain fixed and motion is denoised progressively. Prompt attention follows Eq. (4) throughout sampling. Completed motion is committed to the output and extends the fixed prefix. When a new instruction arrives, committed motion is preserved, while uncommitted predictions are discarded and regenerated using the new instruction together with earlier instructions that remain active. For open-ended interactive commands without a specified end time, we use a 5-second default interval, which can be extended successively for continuous execution. We provide instructions in seconds and then map them to the window interval that the network is operating on.

C Evaluation Protocols and Metrics

C.1 Baseline Text Conditioning

For baselines that accept only a single caption, we split the inference timeline at instruction start and end times. Using the notation of Section 3, instruction $e_i = (\ell_i, [s_i, u_i))$ contributes its text exactly when $s_i \le \tau < u_i$. Within each resulting segment, a single active prompt is passed unchanged,

and multiple active prompts are joined with *while*. The caption is recomputed at each boundary, removing instructions that have ended and adding instructions that have just arrived. For example, prompts A on $[0, 5)$ and B on $[3, 8)$ produce A on $[0, 3)$, A *while* B on $[3, 5)$, and B on $[5, 8)$. Future prompts are withheld until their annotated start times, and caption updates retain the already emitted motion through the baseline's native continuation mechanism. This inference procedure is separate from the window-level caption construction used during retraining.

C.2 Evaluation Metrics

Sequence alignment. We use TMR (Petrovich et al., 2023) to measure motion-to-text R-Precision at ranks 1, 2, and 3, motion-to-text similarity (M2T), and similarity to the corresponding source motion (M2M), with both similarities rescaled to $[0, 1]$. On timeline benchmarks, we evaluate text-motion alignment within each prompt's annotated interval. Overlapping prompts are scored separately over their respective intervals, so shared frames contribute to the evaluation of each prompt. On HumanML3D, alignment is evaluated over complete motions.

Body-part alignment. For body-part alignment, we follow FrankenMotion's released protocol (Li et al., 2026), scoring each prompt's full-body motion crop against its original text with the evaluators for its primary body-part labels. The same evaluators measure M2M between each generated interval and its corresponding source motion. We report R@3, M2T, and M2M averaged equally over the six body regions on TimelineMotion and MTT.

Motion quality. FID measures distributional similarity using whole motion sequences. FID+ compares five crops of up to five seconds per generated timeline with ground truth source motions, following STMC (Petrovich et al., 2024). We report both on TimelineMotion, FID on HumanML3D, and FID+ on MTT. Because MTT has no complete reference timelines, we omit its whole-motion FID and reference-motion temporal-adherence scores. Transition distance (Trans) measures root-relative joint displacement at instruction boundaries (Athanasiou et al., 2022; Petrovich et al., 2024).

Reconstruction. FID measures distributional similarity between reconstructed and reference motions. MPJPE measures the Euclidean distance between corresponding reconstructed and reference joint positions, averaged over joints and all valid frames and reported in millimeters.

Temporal adherence. For temporal adherence, we measure whether each instruction is reflected within its requested interval using frozen whole-body TMR embeddings. For every instruction $e_i = (\ell_i, [s_i, u_i))$, we score its text against 2-second motion windows sampled every 0.2 seconds across the complete timeline. A window is positive if its center lies in $[s_i, u_i)$ and negative otherwise. We report non-interpolated average precision (AP) and the area under the receiver operating characteristic curve (ROC-AUC). AP is computed from the ranking induced by TMR similarity scores, without selecting a fixed similarity threshold. We score overlapping instructions independently and average each metric equally over instructions with both positive and negative windows. Temporal adherence is not applicable to HumanML3D because its captions span complete motions.

C.3 Temporal Adherence Validation

To assess sensitivity to temporal misalignment, we keep each motion and its text-motion similarity scores fixed and shift the requested instruction interval earlier or later by 0.5, 1, or 2 seconds. We evaluate GT instruction intervals from the TimelineMotion test set. Interval lengths and scoring windows are unchanged across shifts. Table 5 shows that mean AP and ROC-AUC are highest at the original intervals for GT. On average, shifting intervals in either direction lowers both metrics, with larger shifts producing larger decreases. These results support using Temporal AP and ROC-AUC to evaluate alignment between motion content and requested instruction intervals.

C.4 Runtime Measurement

Runtime is evaluated on an NVIDIA RTX 6000 Ada at batch size one in FP32 with a sampling window of 78 latents. After five warm-up runs, we report the mean latency to the first clean motion

Table 5: Sensitivity of temporal adherence metrics under shifts of the requested instruction intervals on TimelineMotion. Best and second-best scores are bold and underlined.

Interval shift	AP ↑	ROC-AUC ↑
- 2 s	0.440	0.564
- 1 s	0.580	0.675
- 0.5 s	<u>0.630</u>	0.712
0	**0.647**	**0.728**
+ 0.5 s	0.622	<u>0.713</u>
+ 1 s	0.556	0.671
+ 2 s	0.417	0.568

Table 6: Effect of self-conditioned training on motion generation on HumanML3D and TimelineMotion. Best results are bold.

Variant	HumanML3D			TimelineMotion								
				Sequence alignment		Body-part alignment		Motion quality			Temporal adherence	
	$R@3_{32}$ ↑	M2T ↑	FID ↓	$R@3_{32}$ ↑	M2T ↑	$R@3_{32}$ ↑	M2T ↑	FID ↓	FID+ ↓	Trans (cm) ↓	AP ↑	ROC-AUC ↑
Full model (Ours)	**0.874**	**0.769**	**0.027**	**0.603**	**0.653**	**0.374**	**0.553**	**0.017**	**0.192**	**1.31**	**0.832**	**0.772**
w/o self-conditioned training	0.862	0.767	0.035	0.594	0.650	0.365	0.551	0.024	0.198	1.35	**0.832**	0.770

token and the mean interval between subsequent clean tokens over 100 sequences. Each token represents four motion frames.

C.5 User Study Details

We randomly generated 20 multi-track human motion videos with STMC (Petrovich et al., 2024), ActionPlan* (Nazarenus et al., 2026) and TimelineControl. As shown in Figure 6, for each question, we put the videos from TimelineControl and one of the baseline methods in random order to form 40 questions in total. For each timeline input, 18 participants compare each pair with respect to text alignment, motion realism, and temporal adherence. As presented in Figure 7, TimelineControl consistently outperforms other baselines in all aspects.

D Training Strategies and Additional Experiments

D.1 Self-Conditioned Training

During streaming inference, previously generated latents become context for subsequent predictions. Following MotionStreamer (Xiao et al., 2025), we train with detached model predictions as context. For examples selected for replacement, a first forward pass estimates clean motion and action-plan latents without gradient propagation. We insert the selected predictions into the second-pass input as clean latents and set their noise levels to zero. Gradients are computed only through the second pass. Replaced positions receive zero flow-loss weight in the corresponding stream, while the remaining valid targets retain supervision.

Table 6 compares the full model with its variant without self-conditioned training. Removing self-conditioned training lowers retrieval and increases FID on both datasets, and increases boundary displacement on TimelineMotion. Temporal AP is unchanged at the reported precision, while ROC-AUC decreases slightly.

D.2 Training-Window Strategies

For examples from TimelineMotion, we choose the prediction start either uniformly across the sequence (*Random start*) or at the start of a text instruction (*Instruction start*), with equal probability. Random-start crops cover different stages of motion, while instruction-start crops emphasize responses to newly received instructions. Each crop includes up to P preceding clean motion latents as context. Action-plan latents remain prediction targets. Table 7 compares the full training-window

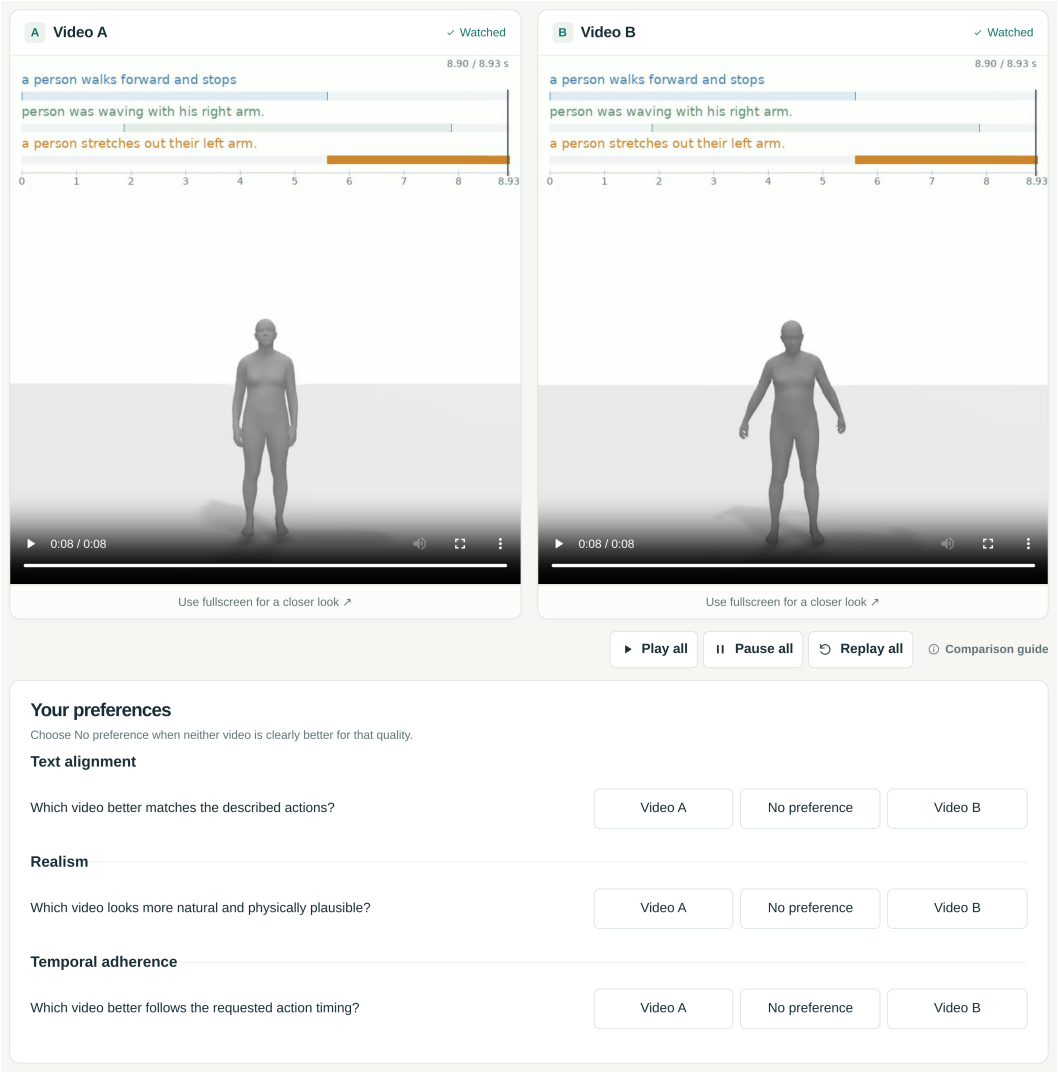

Figure 6: User study interface. We present timeline with videos generated by TimelineControl and one of the baselines. Users are asked to compare the two videos in terms of text-motion alignment, motion quality, and temporal adherence.

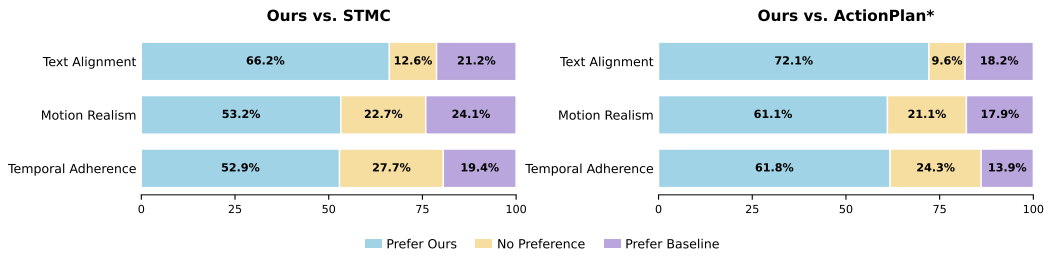

Figure 7: User study. Comparison for timeline human motion videos generated by TimelineControl, STMC and ActionPlan*.

Table 7: Comparison of training-window strategies on HumanML3D and TimelineMotion.

Training-window strategy	HumanML3D			TimelineMotion								
				Sequence alignment		Body-part alignment		Motion quality			Temporal adherence	
	R@3$_{32}$ ↑	M2T ↑	FID ↓	R@3$_{32}$ ↑	M2T ↑	R@3$_{32}$ ↑	M2T ↑	FID ↓	FID+ ↓	Trans (cm) ↓	AP ↑	ROC-AUC ↑
Random start	0.859	0.762	0.035	0.597	0.651	**0.374**	**0.553**	0.027	**0.187**	1.48	0.829	0.765
Mixture (Ours)	**0.874**	**0.769**	**0.027**	**0.603**	**0.653**	**0.374**	**0.553**	**0.017**	0.192	1.31	**0.832**	**0.772**
Instruction start	0.866	0.764	0.029	0.596	0.652	0.371	0.551	0.019	0.196	**1.28**	0.831	0.768

Table 8: Comparison of sigmoid weighting and softmax normalization in multi-track attention on HumanML3D and TimelineMotion.

Variant	HumanML3D			TimelineMotion								
				Sequence alignment		Body-part alignment		Motion quality			Temporal adherence	
	R@3$_{32}$ ↑	M2T ↑	FID ↓	R@3$_{32}$ ↑	M2T ↑	R@3$_{32}$ ↑	M2T ↑	FID ↓	FID+ ↓	Trans (cm) ↓	AP ↑	ROC-AUC ↑
Sigmoid (Ours)	**0.874**	**0.769**	**0.027**	**0.603**	**0.653**	**0.374**	**0.553**	**0.017**	0.192	1.31	**0.832**	**0.772**
Softmax	0.847	0.754	0.041	0.580	0.644	**0.374**	0.551	0.027	**0.190**	**1.23**	0.825	0.757

policy with Random start and Instruction start variants. The mixture achieves the highest sequence retrieval and lowest FID on both datasets, and the highest Temporal AP and ROC-AUC on TimelineMotion. The Random start baseline has the lowest FID+, while the Instruction start baseline has the lowest boundary displacement.

D.3 Attention Normalization

Table 8 compares sigmoid weighting with softmax normalization in text attention. The softmax variant replaces the weighting in both the part-aware and global branches, while retaining the temporal interval encodings, visibility masks, and part-aware biases. Sigmoid weighting yields higher sequence retrieval and lower whole-motion FID on both datasets, and higher Temporal AP and ROC-AUC on TimelineMotion. Softmax yields slightly lower FID+ and boundary displacement on TimelineMotion.

D.4 Multi-Text Conditioning in ActionPlan

Table 9 compares ActionPlan, its jointly retrained variant, and a variant with multi-text conditioning. The multi-text variant encodes each instruction separately and masks its text cross-attention outside the annotated interval, while retaining the backbone depth and width and the same motion autoencoder. It uses softmax text attention without our temporal interval encoding or part-aware branches. Relative to the jointly retrained single-caption baseline, multi-text conditioning improves sequence retrieval and temporal adherence on TimelineMotion. On TimelineMotion, our model further improves sequence and body-part alignment, FID, and temporal adherence, while both models have the same boundary displacement at the reported precision. These results show that multi-text conditioning improves ActionPlan, but does not account for the full gain of our method.

Table 9: Comparison of joint retraining and multi-text conditioning variants of ActionPlan with TimelineControl on HumanML3D and TimelineMotion. * denotes joint retraining on both datasets. Multi-text uses separately encoded instructions with an interval visibility mask. Best and second-best results are bold and underlined.

Variant	HumanML3D			TimelineMotion								
				Sequence alignment		Body-part alignment		Motion quality			Temporal adherence	
	R@3_{32} ↑	M2T ↑	FID ↓	R@3_{32} ↑	M2T ↑	R@3_{32} ↑	M2T ↑	FID ↓	FID+ ↓	Trans (cm) ↓	AP ↑	ROC-AUC ↑
ActionPlan	0.829	0.724	**0.022**	0.405	0.585	0.303	<u>0.547</u>	0.274	0.330	2.16	0.733	0.653
ActionPlan*	0.830	<u>0.734</u>	0.044	0.502	0.622	0.347	0.546	0.055	0.209	**0.92**	0.784	0.711
ActionPlan* + multi-text	<u>0.845</u>	0.731	<u>0.027</u>	<u>0.566</u>	<u>0.640</u>	<u>0.352</u>	<u>0.547</u>	<u>0.036</u>	<u>0.198</u>	<u>1.31</u>	<u>0.805</u>	<u>0.745</u>
TimelineControl	**0.874**	**0.769**	<u>0.027</u>	**0.603**	**0.653**	**0.374**	**0.553**	**0.017**	**0.192**	<u>1.31</u>	**0.832**	**0.772**

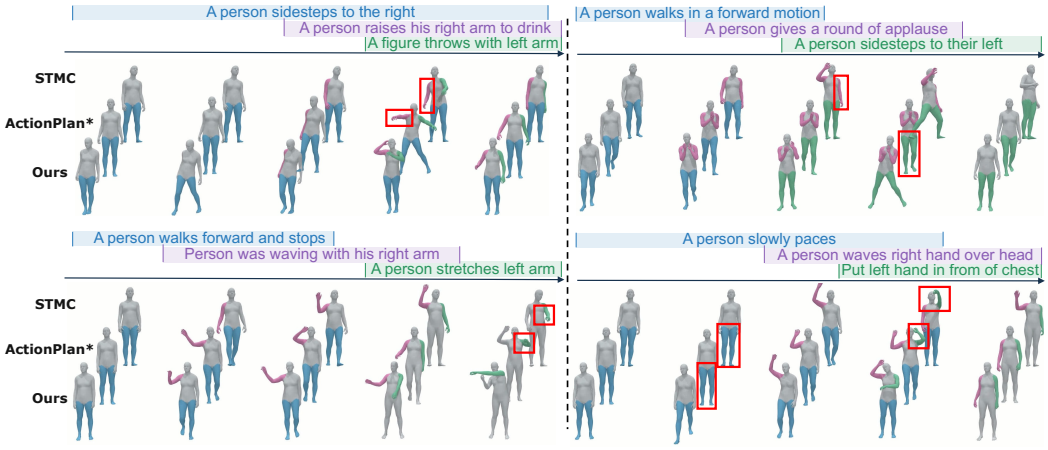

Figure 8: **Additional qualitative evaluation.** Red box marks the failure part.

E Extension to Spatial Control

We extend the text-conditioned generator to accept spatial targets alongside text instructions. At latent position n, we zero out unavailable target values and concatenate the masked, normalized targets with their binary availability masks to form $\boldsymbol{\xi}_n^{\text{ctrl}}$. A zero-initialized, bias-free linear projection (Zhang et al., 2023b) adds $W_c \boldsymbol{\xi}_n^{\text{ctrl}}$ to the initial global state before part-global fusion. The adapted model retains the streaming sampler and supports text-only generation with empty spatial masks.

E.1 Additional Qualitative Results

Figure 8 presents additional qualitative comparisons between TimelineControl, STMC, and ActionPlan* on multi-track motion generation. These examples complement Figure 3 in the main paper and illustrate model behavior under sequential and overlapping instructions.

E.2 Spatial Conditions

Spatial targets are represented in the root and body feature space with binary availability masks. Only available controls on future frames are supplied, using five control types:

- *Root waypoints*: horizontal positions at sparse frames, optionally with heading.
- *Root paths*: horizontal positions and heading at future frames with supplied root controls.
- *Body keyframes*: root features and non-root joint positions at sparse frames.
- *End-effector keyframes*: root features and selected foot or wrist positions and orientations.
- *Foot contacts*: the four binary contact channels at selected frames.

Table 10: Comparison with ARDY for motion generation conditioned jointly on text and spatial targets on HumanML3D and TimelineMotion.

Testing Dataset	Method	Horizontal root (cm) ↓	Body (cm) ↓	End-effector (cm) ↓	Contact F1 ↑	M2T ↑	FID ↓	FID+ ↓	Temporal adherence	
									AP ↑	ROC-AUC ↑
HumanML3D	ARDY	**2.55**	8.15	14.06	0.857	0.650	0.246	–	–	–
	TimelineControl	4.01	**5.42**	**7.95**	**0.939**	**0.786**	**0.020**	–	–	–
TimelineMotion	ARDY	**4.13**	9.98	16.88	0.897	0.565	0.221	0.308	0.715	0.630
	TimelineControl	6.31	**8.05**	**12.63**	**0.942**	**0.644**	**0.013**	**0.216**	**0.820**	**0.759**

Training samples position-only, orientation-only, or combined end-effector conditions. Evaluation uses combined conditions. Combined conditioning uses the union of the masks for two distinct control types.

E.3 Adaptation

Spatial adaptation initializes from the exponential-moving-average weights of the trained generator. We jointly fine-tune the denoiser and condition projection while keeping the autoencoder frozen, with learning rates 2×10^{-5} for the denoiser and 10^{-4} for the condition projection. Following the spatial-conditioning setup of ARDY (Zhao et al., 2026), the objective is

$$\mathcal{L}_{\text{ctrl}} = \mathcal{L}_{\text{gen}} + 0.25\mathcal{L}_{\text{dec}} + \mathcal{L}_{\text{goal}} + \mathcal{L}_{\text{kin}}. \tag{8}$$

The reconstruction term uses L1 error on decoded, normalized body features, while the goal term matches supplied root and body targets. Kinematic consistency compares the explicit and forward-kinematic positions of the 21 non-root joints using mean Euclidean distance. Training mixes single, combined, and absent spatial conditions, gradually increasing the maximum number of sparse keyframes. Reconstruction and kinematic losses use valid future frames, while the goal loss applies only to supplied targets.

E.4 Evaluation

Each motion is randomly assigned to one of six control settings including five individual control types and one combined setting. On TimelineMotion, control availability is synchronized with text-prompt arrivals. When a control becomes available, its world-space targets are translated by the horizontal displacement between the generated and source root positions at that time. The heading is unchanged. Errors are measured relative to the transformed targets. We report position errors for root, body, and end-effector targets in centimeters, and F1 for specified foot-contact labels. Control metrics average applicable examples and evaluate only frames and joints with supplied targets.

E.5 Results

Table 10 compares our extension with ARDY (Zhao et al., 2026). Our extension achieves lower body and end-effector errors and higher contact F1 and M2T on both datasets, while ARDY remains more accurate for horizontal root targets. Figure 9 presents further examples with dense root trajectories, end-effector targets, and full-body keyframes.

F Robot Motion Execution

We execute the generated motions on a physical humanoid robot in real time using SONIC's motion-tracking controller (Luo et al., 2026). Figure 1(c) shows upper-body gestures accompanying walking before a transition to a bow on the robot. The additional sequence in Figure 11 combines hand gestures with locomotion and stepping. Figure 10 presents text-driven G1 motions in simulation, including kicking, phone gestures, and violin playing.

G Limitations

TimelineControl has several limitations. First, TimelineMotion is constructed by composing HumanML3D source motions rather than collecting motions from real streaming multi-track interac-



Figure 9: Motion generation with text and spatial controls: (a) root trajectories, (b) root waypoints, (c) left-wrist targets, (d) full-body keyframes, and root trajectories combined with (e) left-wrist targets or (f) full-body keyframes. Poses share a world coordinate frame within each panel and darken over time. Bars mark instruction intervals.

tions. Although we filter incompatible actions and refine composition boundaries, the resulting data may not capture the full diversity and complexity of naturally co-occurring actions. Second, our formulation uses explicit temporal intervals to determine when each instruction is active. When an end time is not provided, the system can use a fixed-duration interval (e.g., 5 seconds) and extend the instruction in successive intervals for continuous execution. However, it does not currently infer action duration or termination directly from language or motion state. Finally, our part-aware supervision relies on automatically generated body-part annotations and a fixed decomposition into six anatomical regions; annotation errors or actions involving more complex cross-part interactions may therefore limit the part-aware guidance. Extending the framework to naturally collected interactive data, learned action termination, and more flexible part grounding provides promising directions for future work.

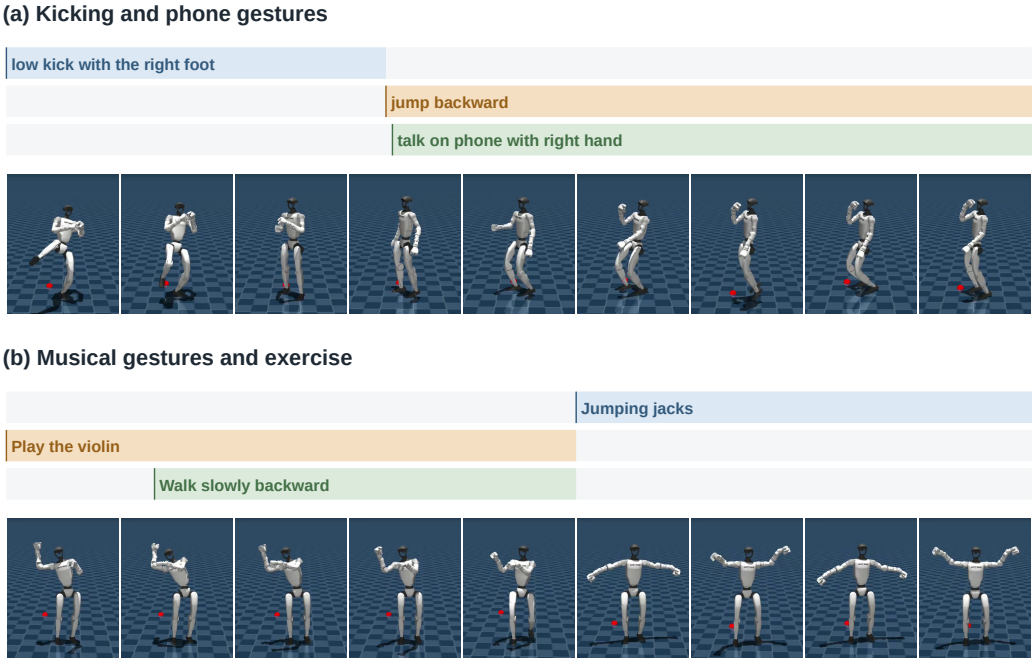

Figure 10: Text-driven G1 simulation. (a) A right-foot kick followed by overlapping instructions to jump backward and talk on the phone with the right hand. (b) Overlapping violin-playing and backward-walking instructions followed by jumping jacks. Bars indicate instruction intervals.

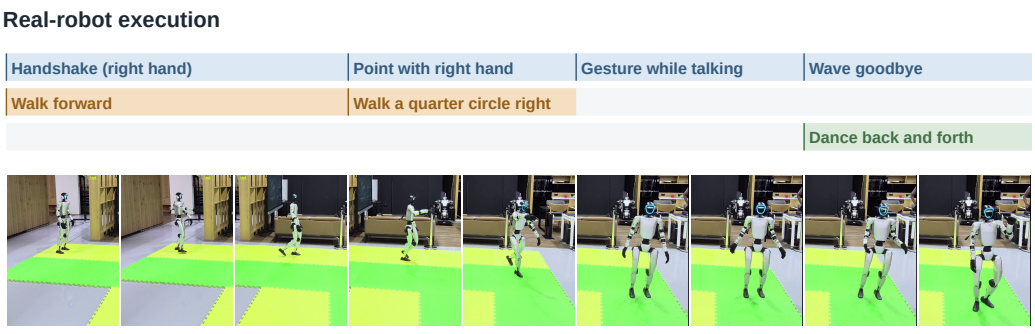

Figure 11: Text-driven real-robot execution combining upper-body gestures with walking and dancing. Bars indicate instruction intervals.