LIVEANIMATE: STABLE LONG-FORM STREAMING HUMAN ANIMATION IN REAL-TIME

Anonymous authors
Paper under double-blind review

ABSTRACT

Pose-driven human animation synthesizes a video of a target person from a single reference image and a driving pose stream. Real-time generation is essential for interactive applications such as live streaming, telepresence, and virtual avatars, yet diffusion-based systems require minutes to hours per clip, precluding responsive interaction. We present LiveAnimate, to our knowledge the first pose-driven human animation system to combine real-time streaming with stable long-form generation at billion scale, built on a 14B-parameter video Diffusion Transformer (DiT). A two-stage training pipeline first adapts a pretrained bidirectional DiT into a block-causal autoregressive generator through Reference-Anchored Teacher-Forcing Adaptation, and then reduces the sampling budget to three steps through Block-wise Self-Forcing Distillation. To preserve appearance over extended streams, we introduce Pose-Retrieval Sink Attention (PR-Sink), a bounded KV-cache mechanism combining a Static Sink that permanently anchors the first generated block, a Dynamic Sink that holds a pose-retrieved historical block, and a three-slot Rolling Window. When a pose recurs, PR-Sink restores the relevant appearance context without retaining the entire sequence, so memory and per-block latency remain constant regardless of stream duration. Together with Ulysses sequence parallelism and operator fusion, these designs enable a DiT throughput of 19.63 FPS on two NVIDIA H100 GPUs, and an end-to-end throughput of 11.2 FPS in a pipelined four-GPU deployment. On a three-minute benchmark, LiveAnimate maintains nearly constant perceptual quality and identity from the first 30 seconds to the final minute (IQA 4.047 vs. 4.026), while prior systems degrade substantially or require hours of offline computation for the same rollout. These results establish a new operating point in quality, latency, and duration for interactive full-body animation.

1 INTRODUCTION

Pose-driven human animation synthesizes a video of a target person that follows a sequence of driving poses while preserving the appearance of a single reference image, with broad applications in virtual try-on, digital humans, live streaming, and telepresence. The core challenge is to simultaneously maintain appearance fidelity, pose accuracy, and temporal coherence over extended sequences. Video diffusion models Ho et al. (2020); Rombach et al. (2022); Peebles & Xie (2023) are the prevailing approach: Animate Anyone Hu (2024), MagicAnimate Xu et al. (2024c), and UniAnimate-DiT Wang et al. (2025) generate short clips via reference-based conditioning and temporal attention, Wan-Animate Wan et al. (2025); Cheng et al. (2025) and SCAIL Yan et al. (2026) scale this design to 14B-parameter DiT backbones, and EverAnimate Li et al. (2026b) extends generation to the minute scale through persistent latent propagation and restorative flow matching.

Table 1: **Capability comparison of human animation methods.** LiveAnimate is the first pose-driven system to achieve real-time streaming with stable long-form generation at billion scale.

Method	Streaming	Real-Time	Stable Long-Form	Param
Animate Anyone Hu (2024)	✗	✗	✗	–
UniAnimate-DiT Wang et al. (2025)	✗	✗	✗	14B
One-to-All Shi et al. (2026)	✗	✗	✗	14B
SCAIL Yan et al. (2026)	✗	✗	✗	14B
Wan-Animate Cheng et al. (2025)	✗	✗	✗	14B
EverAnimate Li et al. (2026b)	✗	✗	✓	14B
LiveAnimate (Ours)	✓	✓	✓	**14B**

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.

However, a critical gap remains: all of these methods are offline, requiring minutes to hours per clip and precluding responsive interaction. As shown in Table 1, no prior pose-driven full-body diffusion system simultaneously supports streaming generation, real-time inference, and stable long-form output. Most methods Hu (2024); Wang et al. (2025); Yan et al. (2026) generate fixed-length clips with no mechanism for an open-ended pose stream, and even the minute-scale EverAnimate Li et al. (2026b) requires 20 denoising steps per chunk. Closing this gap requires solving three coupled problems: *(i)* reducing the inference cost of a billion-scale DiT to an interactive latency budget, *(ii)* converting a bidirectional video model into a causal generator without sacrificing quality, and *(iii)* preventing identity and appearance drift over arbitrarily long rollouts.

To address these challenges, we present LiveAnimate, the first pose-driven system to enable real-time streaming human animation with stable long-form generation at billion scale, built on a 14B-parameter causal DiT. In Stage 1, *Reference-Anchored Teacher-Forcing Adaptation* converts the pretrained bidirectional DiT into a block-causal generator using ground-truth clean history, with a global Ref Sink that keeps the reference image visible to all generated blocks. In Stage 2, *Block-wise Self-Forcing Distillation* (BS-DMD), building on Self Forcing Huang et al. (2026), reduces the sampling budget to three steps: a Self-Forcing Rollout without gradient tracking followed by Block-wise DMD Optimization on one replayed block at a time provides a distribution-matching signal at every block position while keeping training feasible on a single 8×80GB GPU node. For long-term coherence, *Pose-Retrieval Sink Attention (PR-Sink)* maintains appearance consistency over long-form streams with a Static Sink, a pose-retrieved Dynamic Sink, and a bounded Rolling Window. With Ulysses-style sequence parallelism Fang & Zhao (2024) and operator fusion, LiveAnimate reaches 19.63 FPS DiT throughput on 2×H100 GPUs and 11.2 FPS end-to-end in a four-GPU deployment (Appendix B).

In summary, our contributions are fourfold:

- We present LiveAnimate, the first pose-driven system to enable real-time streaming human animation with stable long-form generation at billion scale, achieving a DiT throughput of ∼20 FPS on 2×H100 GPUs and 11.2 FPS end-to-end in deployment.
- We design a two-stage training pipeline: Reference-Anchored Teacher-Forcing Adaptation converts a pretrained bidirectional DiT into a block-causal generator, and Block-wise Self-Forcing Distillation reduces the sampling budget to three steps via a one-block-at-a-time replay scheme, keeping training feasible on a single 8×80GB GPU node.
- We propose PR-Sink, a bounded KV-cache strategy that combines a Static Sink, a pose-retrieved Dynamic Sink, and a Rolling Window to maintain appearance consistency over long-form streams.
- On a three-minute benchmark, LiveAnimate preserves quality and identity across the rollout at ∼20 FPS DiT throughput, establishing a favorable quality–latency–duration trade-off against state-of-the-art offline systems.

2 Related Work

2.1 Video Diffusion Models

Video diffusion models have evolved from pixel-space U-Nets Ho et al. (2020; 2022) to latent-space architectures and Diffusion Transformers (DiTs) Peebles & Xie (2023). Video LDM Blattmann et al. (2023b) and Stable Video Diffusion Blattmann et al. (2023a) showed that operating in a compressed VAE latent space enables high-resolution video generation at manageable cost. More recently, billion-scale DiTs have become strong backbones for controllable generation; Wan Wan et al. (2025), for example, scales to 14B parameters with a causal 3D VAE and a flow-matching objective. Our work builds on Wan2.2-Animate-14BCheng et al. (2025) and adapts it from offline clip generation to causal, real-time streaming.

2.2 Pose-Driven Human Animation

Pose-driven human animation synthesizes video of a target person following a driving signal such as skeleton sequences or DensePose maps. Early GAN-based warping methods Siarohin et al. (2019);

Wang et al. (2018) run in real time but produce artifacts under large pose changes. Diffusion-based methods now prevail: Animate Anyone Hu (2024), MagicAnimate Xu et al. (2024c), and MusePose Tong et al. (2024) established the reference-and-pose conditioning paradigm, followed by extensions to 3D body guidance Zhu et al. (2024), long-form generation Wang et al. (2025); Gan et al. (2025); Li et al. (2026b), and improved control and fidelity Shi et al. (2026); Zhang et al. (2025); Yan et al. (2026), as well as large-scale or multi-person DiT frameworks Wan et al. (2025); Cheng et al. (2025); Hu et al. (2026); Jiang et al. (2025). Nearly all of these methods are designed for offline generation and do not jointly target low-latency streaming and minute-scale temporal stability.

2.3 Autoregressive Video Generation

Autoregressive chunk-wise generation extends video beyond a model's training horizon, but repeated rollouts accumulate visual degradation and semantic drift. A line of work reduces this exposure bias through per-token noise levels Chen et al. (2024), causal distillation of bidirectional teachers Yin et al. (2025); Zhu et al. (2026), self-forcing rollout training Huang et al. (2026); Yin et al. (2024), rolling-window or long-context alignment Liu et al. (2026); Chen et al. (2026), and error-recycling or sink-based cache strategies Li et al. (2026c;a). These methods mitigate rollout drift in general video generation, but none addresses pose-aware cache retention under a strict real-time latency budget.

Interactive avatars have been explored most actively in the audio-driven setting: real-time talking-face systems Xu et al. (2024b); Zhang et al. (2024) and diffusion-based portrait animators Tian et al. (2024); Xu et al. (2024a) synthesize speech-driven talking heads, and TaoAvatar Chen et al. (2025) renders full-body talking avatars in real time via 3D Gaussian splatting. A concurrent line of work streams audio-driven avatars from billion-scale diffusion models through causal few-step distillation Sun et al. (2026); Huang et al. (2025); Shen et al. (2025); Ki et al. (2026); Team et al. (2026). Our setting is complementary and distinct: we target pose-driven full-body animation from a skeleton stream rather than audio, retaining pose-relevant historical tokens under a bounded cache to sustain identity over minute-scale streams.

3 Method

3.1 Overview

In this section, we present LiveAnimate, a framework for real-time streaming pose-driven human animation using a billion-scale causal Diffusion Transformer. As illustrated in Figure 1, given a reference image I_{ref} and a stream of driving pose skeletons $\{P_t\}$, our system generates video blocks autoregressively, each denoised in 3 steps and followed by a clean KV cache update. The framework consists of two components: (1) a two-stage training pipeline comprising Reference-Anchored Teacher-Forcing Adaptation (Sec. 3.2) and Block-wise Self-Forcing Distillation (Sec. 3.3); and (2) Pose-Retrieval Sink Attention (PR-Sink, Sec. 3.4), a bounded KV-cache mechanism for stable long-form streaming. System-level optimizations that enable a DiT throughput of 19.63 FPS on 2×H100 GPUs are described in Appendix A, and the end-to-end deployment pipeline is analyzed in Appendix B.

3.2 Stage 1: Reference-Anchored Teacher-Forcing Adaptation

Standard video diffusion models denoise complete sequences with bidirectional attention and therefore cannot directly emit temporal blocks causally. We adapt the pretrained DiT into a block-causal autoregressive generator through Reference-Anchored Teacher-Forcing Adaptation.

We partition the training video into temporal blocks of B_f latent frames, and denoise each block b while attending to ground-truth clean target blocks $z^{\text{GT}}_{<b}$ rather than to its own predictions, preventing autoregressive errors from entering the conditioning context; exposure to student-generated history is introduced only in Stage 2.

A key distinction from standard teacher forcing in unconditional generation is the *Ref Sink*, a global conditional-cache region that stores the reference-image KV states. We encode the reference latent

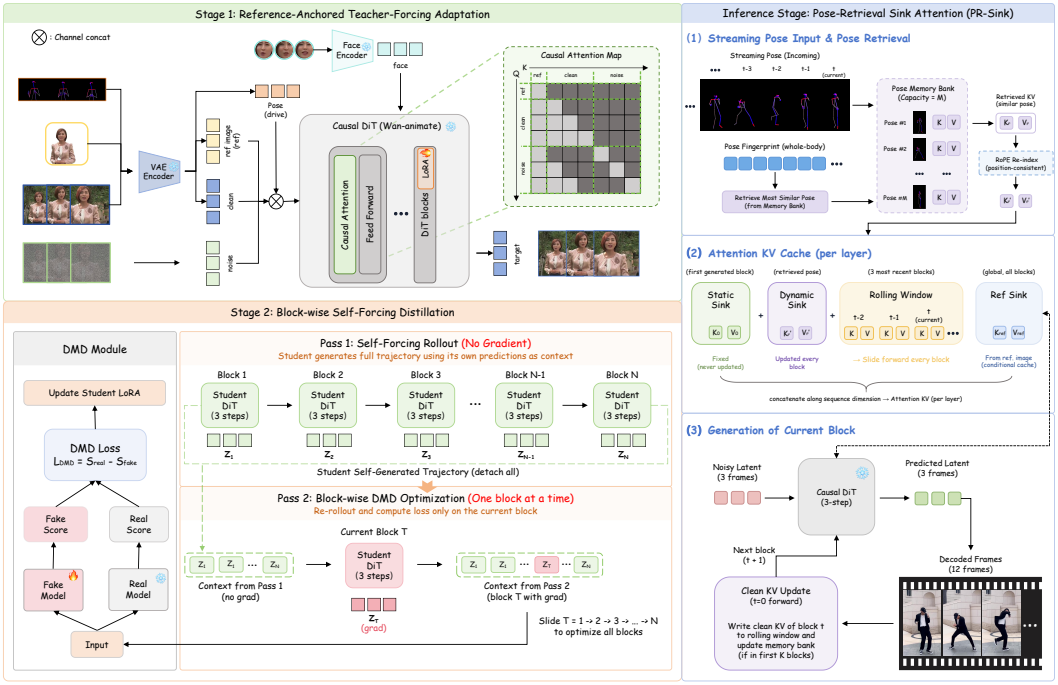

Figure 1: **Overview of LiveAnimate.** Given a reference image and streaming pose signals, our system generates video blocks autoregressively. Each block undergoes 3-step denoising followed by a clean KV update. PR-Sink augments a three-block rolling window with the first generated block and a pose-matched historical block selected from a compact memory bank. Ulysses sequence parallelism distributes attention computation across GPUs.

z_{ref} through a forward pass at the clean timestep (t=0) and make the resulting Ref Sink visible to all subsequent blocks, so that block-causal self-attention for target-block queries Q_b reads the Ref Sink, the ground-truth clean history, and the current block:

$$\text{Attn}(Q_b, K, V) = \text{softmax}\left(\frac{Q_b[K_{\text{ref}}; K_{<b}]^T}{\sqrt{d}}\right)[V_{\text{ref}}; V_{<b}] \tag{1}$$

where $K_{\text{ref}}, V_{\text{ref}}$ are the reference key–value pairs, and $K_{<b}, V_{<b}$ are the teacher-provided clean KV states from blocks preceding b. The reference frame thus serves as a permanent context anchor, analogous to attention sinks in LLMs Xiao et al. (2024).

At causal inference and during the Self-Forcing Rollout in Stage 2, a denoised block $\hat{z}_b^0$ is written to the historical cache through the *Clean KV Update*, which performs an additional forward pass at the clean timestep:

$$\text{KV}_{\text{cache}} \leftarrow \text{KV}_{\text{cache}} \cup \text{Forward}(\hat{z}_b^0, t{=}0, \mathcal{C}) \tag{2}$$

The Clean KV Update places historical states at the clean noise level used by the causal context. It does not remove errors already present in the generated latent.

3.3 Stage 2: Block-wise Self-Forcing Distillation

The Stage-1 generator still requires 50 denoising steps per block, precluding real-time inference. In Stage 2, we reduce the sampling budget to three steps through *block-wise self-forcing distribution matching distillation* (BS-DMD), building on Self Forcing Huang et al. (2026): the student is exposed to autoregressive histories induced by its own predictions rather than clean ground-truth history, reducing the train–inference exposure gap.

Directly backpropagating through an N-block self-forcing rollout would require retaining the activation graph of the entire autoregressive trajectory. We instead use a two-pass block-wise replay procedure. In the first pass, the student generates a complete trajectory without gradient tracking:

$$(\bar{z}_1, \ldots, \bar{z}_N) = \text{Rollout}_\theta(\epsilon_{1:N}, \mathcal{C}) \tag{3}$$

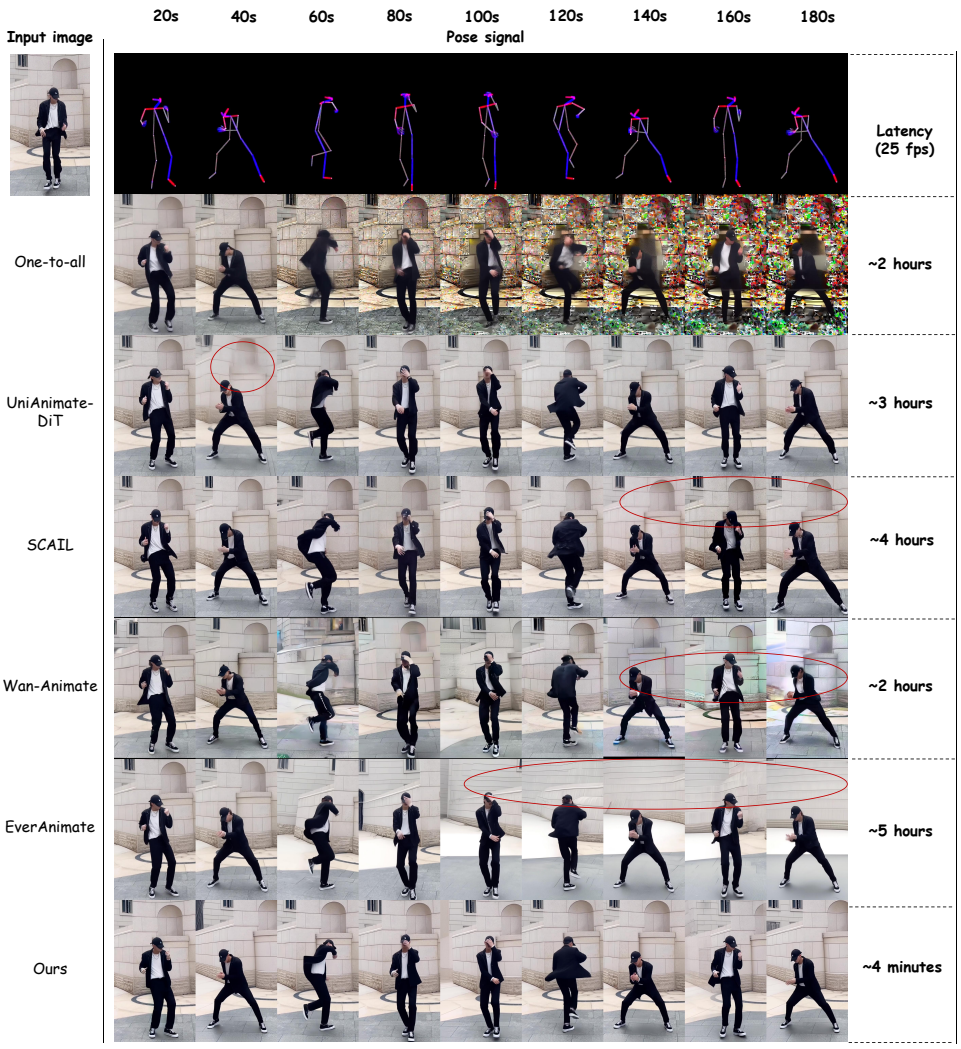

Figure 2: **Qualitative comparison on a full-body sequence.** Frames are sampled every 20 seconds from a three-minute, 25-FPS rollout. The pose signal is shown above the generated frames, and red annotations highlight representative long-horizon artifacts in competing methods. The rightmost column reports total generation time: the baselines require approximately 2–5 hours, whereas LiveAnimate completes the sequence in approximately 4 minutes.

Every block is therefore generated under student-induced history, while the rollout activations are not retained for backward.

In the second pass, we replay one block position T at a time. The contextual latent and KV states reconstructed from the first-pass trajectory are detached, and only the current block is recomputed with gradient tracking:

$$z_T^\theta = G_\theta(\epsilon_T; \mathrm{sg}(\mathrm{KV}(\bar{z}_{<T}))\,, \mathcal{C}_T)\,, \tag{4}$$

where $\mathrm{sg}(\cdot)$ denotes stop-gradient. We insert the recomputed block into the otherwise detached trajectory,

$$\tilde{z}^{(T)} = [\mathrm{sg}(\bar{z}_1), \ldots, z_T^\theta, \ldots, \mathrm{sg}(\bar{z}_N)], \tag{5}$$

and evaluate the distribution-matching objective on the concatenated trajectory:

$$\mathcal{L}_T = \mathcal{L}_{\mathrm{DMD}}\left(\tilde{z}^{(T)}; \mathcal{C}\right), \qquad T = 1, \ldots, N. \tag{6}$$

The teacher and critic thus evaluate a complete student-induced trajectory, whereas gradients reach the student only through the replayed block z_T^θ, whose activation graph is released before the next block is processed. Sliding T from 1 to N gives every block position a direct training signal while

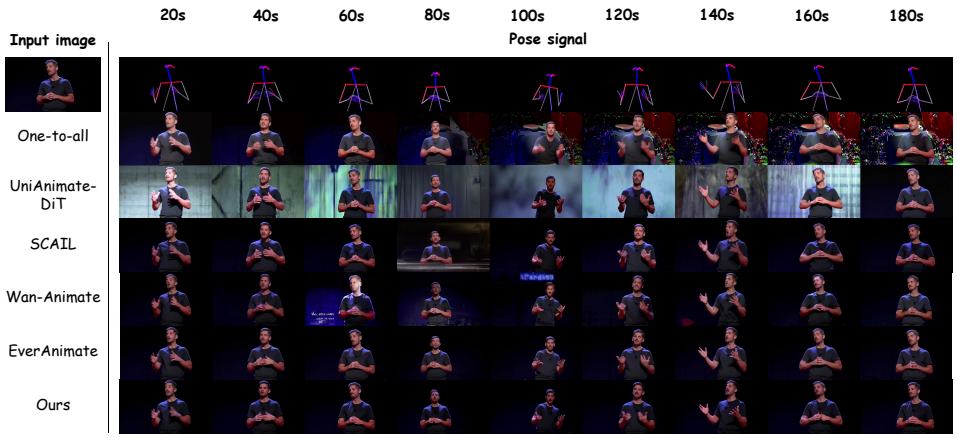

Figure 3: **Qualitative comparison on an upper-body sequence.** Frames are sampled every 20 seconds from a three-minute rollout under the same reference image and driving-pose sequence. LiveAnimate maintains the subject's identity, clothing, and dark background more consistently over time.

retaining only one block's activation graph at a time; together with LoRA, this keeps Stage-2 distillation of the 14B student feasible on a single 8×80GB GPU node (see Appendix C for the gradient and memory analysis).

3.4 Pose Retrieval Sink Attention

Autoregressive streaming faces a trade-off between memory and context. Keeping all historical key–value (KV) states gives full context, but memory and attention cost grow linearly with duration. A bounded cache has constant cost, but it discards appearance evidence as soon as a block leaves the window, which is especially harmful when the subject returns to a previously observed pose. PR-Sink resolves this trade-off with two persistent cache regions. A *Static Sink* permanently stores the first generated block as a time-invariant appearance anchor. A *Dynamic Sink* stores one historical block retrieved from a compact memory bank by pose similarity, restoring pose-specific evidence when a similar pose recurs.

Cache organization. For each block t, attention reads four fixed-capacity regions: the Static Sink (the first generated block), the Dynamic Sink (one pose-retrieved block), a three-slot Rolling Window (clean blocks $t-2$ and $t-1$ plus the current block t), and the global Ref Sink for reference-image tokens. After block t is denoised, the Clean KV Update writes its clean-timestep KV state into the Rolling Window and advances the window. Since all regions have fixed capacity, cache storage and attention cost do not grow with stream duration.

Pose fingerprint. We extract whole-body keypoints (body and hands) with ViTPose and convert them into the animation representation. For each block, we concatenate the keypoint coordinates and confidence scores of its three ordered pose frames and apply ℓ_2 normalization, yielding a single pose fingerprint ϕ_b whose dot products equal cosine similarities. Temporal concatenation preserves short-term motion configuration that would be lost under mean pooling. If pose detection fails for an entire frame, its portion of the fingerprint is zero padded.

Bank update. The memory bank $\mathcal{B} = \{(\phi_i, \mathrm{KV}_i)\}$ has a fixed capacity $M = 5$. Each entry stores the fingerprint and the unrotated keys and values from all transformer layers. Once the bank is full, an incoming block q may replace one existing entry. We test every possible replacement and choose the bank with the lowest average pairwise cosine similarity:

$$i^* = \arg\min_i \ \mathrm{AvgSim}((\mathcal{B} \setminus \{i\}) \cup \{q\}) \,. \tag{7}$$

We perform the replacement only if it lowers the current bank similarity. Thus, writing favors *coverage*: redundant poses are rejected, while novel articulations increase the diversity of the stored context. Bank updates are restricted to the first 20 blocks, after which the selected representatives are reused throughout the stream.

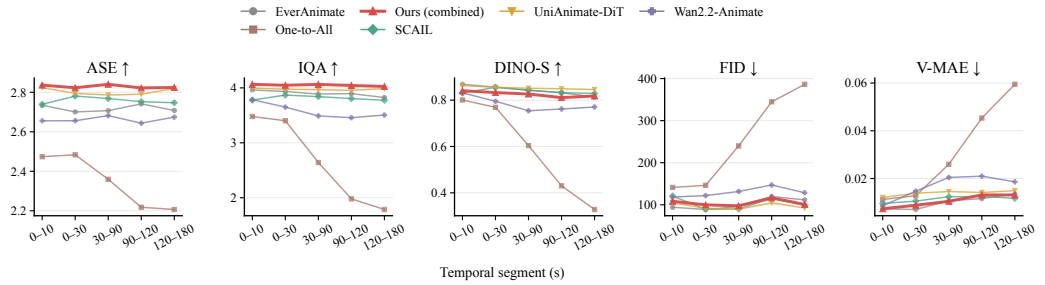

Figure 4: **Quality and identity over a three-minute rollout.** Metrics are computed on a 0–10 s prefix, the 0–30 s initial window, and three subsequent temporal segments. The full LiveAnimate model is highlighted in red. Flat trajectories indicate resistance to accumulated degradation. Higher is better for ASE, IQA, and DINO-S; lower is better for FID and V-MAE.

Bank retrieval. Before generating block b, we compute its fingerprint from the incoming pose signal and retrieve

$$r(b) = \underset{i \in \mathcal{B},\, i \neq b-1}{\arg\max}\ \boldsymbol{\phi}_b^\top \boldsymbol{\phi}_i. \tag{8}$$

We exclude block $b-1$ because it is already present in the Rolling Window; selecting it again would waste the Dynamic Sink on redundant local context. The retrieved KV state is copied into the Dynamic Sink. Bank writing therefore seeks *coverage*, whereas bank reading seeks *relevance* to the current pose. PR-Sink is the union of the pose-conditioned Dynamic Sink and the permanent first-block Static Sink; neither component alone constitutes the full mechanism.

RoPE follow. As the stream advances, the Static Sink, Dynamic Sink, and Ref Sink are retained while new blocks are appended, so the temporal positions assigned to their keys must roll forward together with the current block. Caching keys with their original rotary position embeddings (RoPE) would freeze stale positions and introduce positional mismatch at attention time. We therefore cache K^{raw} before RoPE together with V, and apply RoPE only at attention time, rotating the sink and reference keys at their followed positions in the current cache layout before they are concatenated with the current block. We term this scheme *RoPE follow*: it decouples content selection from the original timestamp, so cached states remain valid at any later position.

4 Experiments

4.1 Experimental Setup

Implementation details. We train LiveAnimate on 40k talking videosEphrat et al. (2018) and 20k human motion videosJafarian & Park (2021); Wang et al. (2024), initialized from the Wan2.2-Animate-14B checkpoint and fine-tuned with LoRA of rank 128 in two stages on 8 NVIDIA H100 GPUs. Inference runs on two H100 80GB GPUs and produces blocks of three latent frames (12 RGB frames). Training hyperparameters are listed in Appendix D.

Evaluation protocol. We construct a three-minute benchmark of 24 reference-image and driving-video pairs: 12 in-the-wild videos covering full-body and upper-body motion, and 12 sequences derived from X-Dance Zhang et al. (2025) by playing each driving pose sequence forward, in reverse, and forward again, so that poses recur at large temporal intervals and directly stress identity consistency. Each method receives the same reference image and driving poses. We report metrics on a 0–10 s prefix, the 0–30 s initial window, and three later segments (30–90, 90–120, and 120–180 s): ASE and IQAWu et al. (2023) measure frame-level visual quality, DINO-SCaron et al. (2021) measures identity consistency with the reference, and FID Heusel et al. (2017) and V-MAE Tong et al. (2022) evaluate distributional quality and temporal representation error. Unless otherwise stated, all latency and throughput numbers measure the DiT generation loop only, which dominates the recurrent per-block cost (Table 2); the measurement protocol, the end-to-end deployment analysis, and a motion-following evaluation under an external pose detector are given in Appendices D, B, and G, respectively.

Baselines. We compare with five recent full-body animation systems: EverAnimate Li et al. (2026b), One-to-All Shi et al. (2026), SCAIL Yan et al. (2026), UniAnimate-DiT Wang et al. (2025), and Wan2.2-Animate Wan et al. (2025); Cheng et al. (2025). We additionally ablate the two com-

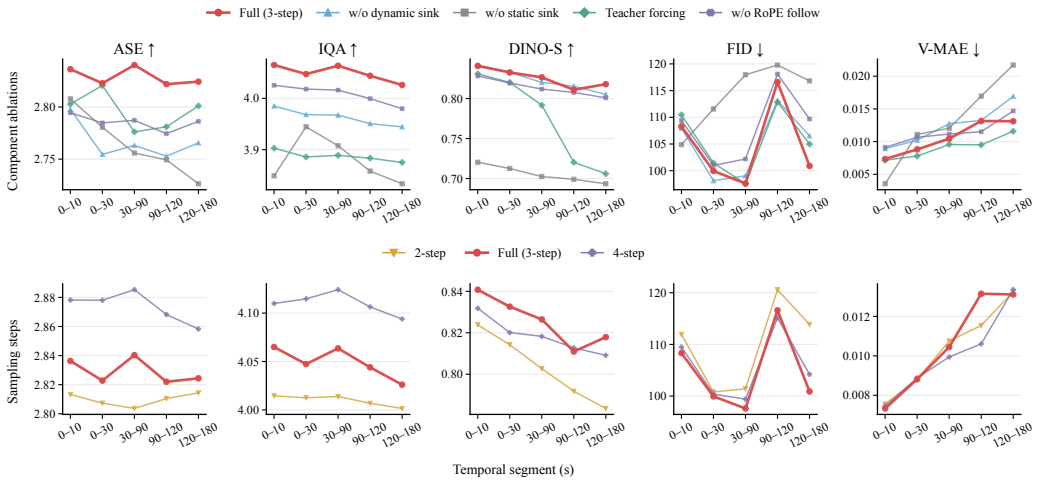

Figure 5: **Quantitative ablation over the three-minute rollout.** Top: component and cache variants, including separate removal of the static and dynamic sinks. The w/o-DMD variant replaces DMD-based self-forcing distillation with conventional teacher forcing. Bottom: the sampling-step trade-off with the full PR-Sink configuration. Higher is better for ASE, IQA, and DINO-S; lower is better for FID and V-MAE.

ponents of PR-Sink (*w/o Dynamic Sink* and *w/o Static Sink*). The long-video baseline protocol is described in Appendix E.

4.2 Quantitative Comparison

Short-horizon quality. Figure 4 reports quality and identity metrics across the temporal segments of the three-minute rollout. On the initial 0–30 s segment, LiveAnimate obtains the best ASE (2.823) and IQA (4.047), showing that three-step distillation preserves strong perceptual quality under a real-time inference budget.

Long-horizon stability. The main advantage of LiveAnimate emerges as the rollout grows: its IQA stays nearly constant (4.047 to 4.026), ASE is unchanged (2.823 to 2.824), and DINO-S decreases by only 0.015. In contrast, One-to-All accumulates severe degradation (IQA 3.402 to 1.786, DINO-S 0.769 to 0.328, FID 146.29 to 386.33), and Wan2.2-Animate also loses identity similarity (DINO-S 0.795 to 0.770). SCAIL and UniAnimate-DiT attain higher DINO-S throughout the rollout, but exhibit inter-frame flickering (Figure 2), and EverAnimate maintains competitive per-frame quality only at the cost of hours of offline computation per sequence. LiveAnimate is thus the only evaluated method that sustains visibly consistent quality over the full rollout while generating in real time. Sequence-parallel scaling results are reported in Appendix F.

Inference-time breakdown. Table 2 reports the steady-state latency breakdown: each 12-frame block takes 0.611 s, a DiT throughput of 19.63 FPS; with a fixed-size KV cache, latency and memory are independent of stream length. Denoising accounts for 75.07% of the runtime, followed by the Clean KV Update at 24.39%, while PR-Sink retrieval and bank maintenance contribute only 0.542%. In deployment, condition preprocessing and VAE coding overlap with DiT generation on separate GPUs, yielding 11.2 FPS end-to-end (Appendix B). All evaluated baselines instead require approximately 2–5 hours to synthesize the same three-minute sequence.

Table 2: **Steady-state profile of the DiT generation loop on the three-minute benchmark.** Steady state denotes inference after compilation and warm-up. Throughput (FPS) is computed from 12 RGB frames per block and covers the DiT loop only; condition processing and VAE encoding/decoding are excluded and accounted for in the end-to-end deployment analysis (Appendix B).

Component	Avg. (s)	Min. (s)	Max. (s)	Share	FPS
Sink read	0.003303	0.003197	0.005136	0.540%	–
Denoising	0.458884	0.451726	0.476683	75.070%	–
Clean KV Update	0.149078	0.146292	0.168945	24.388%	–
Sink write	0.000011	0.000009	0.000023	0.0018%	–
Block total	0.611276	0.601543	0.631650	100.00%	19.63

4.3 Qualitative Comparison

Figure 2 presents a challenging full-body example with large pose changes. The competing methods fail in visibly different ways. One-to-All suffers a severe collapse of image quality as the rollout progresses. UniAnimate-DiT and SCAIL maintain the global structure but exhibit inter-frame flickering. Wan-Animate and EverAnimate remain largely stable but exhibit mild color drift and background drift at later timestamps. In contrast, LiveAnimate follows the driving poses while preserving both the subject's appearance and the scene throughout the three minutes. Competing methods require approximately 2–5 hours to generate the 4500-frame sequence, whereas LiveAnimate completes it in about 4 minutes.

Figure 3 presents an upper-body example under subtler motion. UniAnimate-DiT, SCAIL, and Wan-Animate exhibit background flickering across the sequence, and One-to-All again suffers a severe drop in image quality. EverAnimate remains largely stable with mild drift, whereas LiveAnimate stays consistent; moreover, only LiveAnimate runs in real time.

4.4 Ablation Study

Quantitative ablation. Figure 5 reports the corresponding metric trajectories. Removing the Dynamic Sink moderately degrades long-horizon quality (final-segment DINO-S 0.818 to 0.805, FID 100.90 to 106.52). Removing the Static Sink is substantially worse: final-segment DINO-S collapses to 0.693 despite moderate frame-level scores (ASE 2.726, IQA 3.833), showing that frame-level metrics alone cannot diagnose identity failure. The two sinks are complementary: the Static Sink supplies a stable global appearance anchor, while the Dynamic Sink restores pose-specific evidence that has left the Rolling Window. Disabling RoPE follow also weakens final-segment identity (DINO-S 0.801). Replacing DMD-based distillation with teacher forcing steadily degrades the rollout (IQA 3.849 to 3.438, DINO-S 0.780 to 0.679), indicating that self-forcing limits error accumulation under few-step sampling. Finally, four sampling steps yield the strongest frame-level scores but require an extra denoising pass, while two steps weaken long-horizon identity (final DINO-S 0.783); three steps provide the best efficiency–quality compromise.

Qualitative ablation. Figure 6 visualizes the rollout under each variant. Removing the Dynamic Sink introduces localized appearance errors at recurring poses (face, shoulder). Removing the Static Sink is far worse: the subject and background change completely, showing that pose retrieval alone cannot anchor identity. Replacing DMD-based distillation with teacher forcing causes illumination and appearance to fluctuate, and disabling RoPE follow leads to progressive color and background shifts.

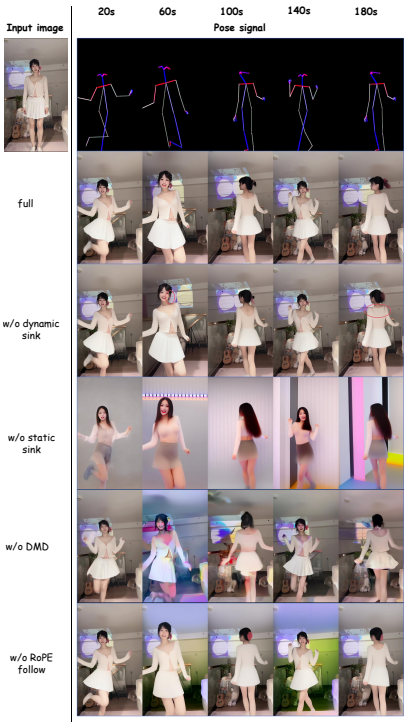

Figure 6: **Qualitative ablation over a three-minute rollout.** Frames are sampled every 40 seconds from 20 to 180 seconds under the same reference image and driving-pose sequence.

5 Conclusion

We presented LiveAnimate, a streaming pose-driven human animation system that combines real-time generation with stable long-form quality. A two-stage pipeline turns a pretrained bidirectional DiT into a block-causal three-step generator, and Pose-Retrieval Sink Attention preserves long-range appearance. On the three-minute benchmark, LiveAnimate sustains nearly constant quality and identity at 19.63 FPS DiT throughput (11.2 FPS end-to-end), while offline baselines degrade or take hours.

REFERENCES

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023a.

Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22563–22575. IEEE, 2023b.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 9630–9640. IEEE, 2021.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024.

Jianchuan Chen, Jingchuan Hu, Gaige Wang, Zhonghua Jiang, Tiansong Zhou, Zhiwen Chen, and Chengfei Lv. Taoavatar: Real-time lifelike full-body talking avatars for augmented reality via 3d gaussian splatting. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10723–10734. IEEE, 2025.

Shuo Chen, Cong Wei, Sun Sun, Ping Nie, Kai Zhou, Ge Zhang, Ming-Hsuan Yang, and Wenhu Chen. Context forcing: Consistent autoregressive video generation with long context. *arXiv preprint arXiv:2602.06028*, 2026.

Gang Cheng, Xin Gao, Li Hu, Siqi Hu, Mingyang Huang, Chaonan Ji, Ju Li, Dechao Meng, Jinwei Qi, Penchong Qiao, et al. Wan-animate: Unified character animation and replacement with holistic replication. *arXiv preprint arXiv:2509.14055*, 2025.

Ariel Ephrat, Inbar Mosseri, Oran Lang, Tali Dekel, Kevin Wilson, Avinatan Hassidim, William T Freeman, and Michael Rubinstein. Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation. *arXiv preprint arXiv:1804.03619*, 2018.

Jiarui Fang and Shangchun Zhao. Usp: A unified sequence parallelism approach for long context generative ai. *arXiv preprint arXiv:2405.07719*, 2024.

Qijun Gan, Yi Ren, Chen Zhang, Zhenhui Ye, Pan Xie, Xiang Yin, Zehuan Yuan, Bingyue Peng, and Jianke Zhu. Humandit: Pose-guided diffusion transformer for long-form human motion video generation. *arXiv preprint arXiv:2502.04847*, 2025.

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. *Advances in neural information processing systems*, 30, 2017.

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

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in neural information processing systems*, 35:8633–8646, 2022.

Li Hu. Animate anyone: Consistent and controllable image-to-video synthesis for character animation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8153–8163, 2024.

Yingcheng Hu, Haowen Gong, Chuanguang Yang, Zhulin An, Yongjun Xu, and Songhua Liu. Multianimate: Pose-guided image animation made extensible. *arXiv preprint arXiv:2602.21581*, 2026.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *Advances in Neural Information Processing Systems*, 38:167283–167308, 2026.

Yubo Huang, Hailong Guo, Fangtai Wu, Weiqiang Wang, Shifeng Zhang, Shijie Huang, Qijun Gan, Lin Liu, Sirui Zhao, Enhong Chen, et al. Live avatar: Streaming real-time audio-driven avatar generation with infinite length. *arXiv preprint arXiv:2512.04677*, 2025.

Yasamin Jafarian and Hyun Soo Park. Learning high fidelity depths of dressed humans by watching social media dance videos. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12753–12762, 2021.

Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. Vace: All-in-one video creation and editing. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17191–17202. IEEE, 2025.

Taekyung Ki, Sangwon Jang, Jaehyeong Jo, Jaehong Yoon, and Sung Ju Hwang. Avatar forcing: Real-time interactive head avatar generation for natural conversation. *arXiv preprint arXiv:2601.00664*, 2026.

Haodong Li, Shaoteng Liu, Zhe Lin, and Manmohan Chandraker. Rolling sink: Bridging limited-horizon training and open-ended testing in autoregressive video diffusion. *arXiv preprint arXiv:2602.07775*, 2026a.

Wuyang Li, Yang Gao, Mariam Hassan, Lan Feng, Wentao Pan, Po-Chien Luan, and Alexandre Alahi. Everanimate: Minute-scale human animation via latent flow restoration. *arXiv preprint arXiv:2605.15042*, 2026b.

Wuyang Li, Wentao Pan, Po-Chien Luan, Yang Gao, and Alexandre Alahi. Stable video infinity: Infinite-length video generation with error recycling. In *International Conference on Learning Representations*, volume 2026, pp. 23406–23432, 2026c.

Kunhao Liu, Wenbo Hu, Jiale Xu, Ying Shan, and Shijian Lu. Rolling forcing: Autoregressive long video diffusion in real time. In *International Conference on Learning Representations*, volume 2026, pp. 91177–91196, 2026.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

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

Le Shen, Qian Qiao, Tan Yu, Ke Zhou, Tianhang Yu, Yu Zhan, Zhenjie Wang, Ming Tao, Shunshun Yin, and Siyuan Liu. Soulx-flashtalk: Real-time infinite streaming of audio-driven avatars via self-correcting bidirectional distillation. *arXiv preprint arXiv:2512.23379*, 2025.

Shijun Shi, Jing Xu, Zhihang Li, Chunli Peng, Xiaoda Yang, Lijing Lu, Kai Hu, and Jiangning Zhang. One-to-all animation: Alignment-free character animation and image pose transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4011–4021, 2026.

Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. *Advances in neural information processing systems*, 32, 2019.

Zhiyao Sun, Ziqiao Peng, Yifeng Ma, Yi Chen, Zhengguang Zhou, Zixiang Zhou, Guozhen Zhang, Youliang Zhang, Yuan Zhou, Qinglin Lu, et al. Streamavatar: Streaming diffusion models for real-time interactive human avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10887–10897, 2026.

Meituan LongCat Team, Xunliang Cai, Meng Cheng, Feng Gao, Zhe Kong, Jiamu Li, Le Li, Weiheng Li, Hongyu Liu, Shuai Tan, et al. Longcat-video-avatar 1.5 technical report. *arXiv preprint arXiv:2605.26486*, 2026.

Linrui Tian, Qi Wang, Bang Zhang, and Liefeng Bo. Emo: Emote portrait alive generating expressive portrait videos with audio2video diffusion model under weak conditions. In *European Conference on Computer Vision*, pp. 244–260. Springer, 2024.

Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in neural information processing systems*, 35: 10078–10093, 2022.

Zhengyan Tong, Chao Li, Zhaokang Chen, Bin Wu, and Wenjiang Zhou. Musepose: a pose-driven image-to-video framework for virtual human generation. *arxiv*, 2024.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. *arXiv preprint arXiv:1808.06601*, 2018.

Xiang Wang, Shiwei Zhang, Changxin Gao, Jiayu Wang, Xiaoqiang Zhou, Yingya Zhang, Luxin Yan, and Nong Sang. Unianimate: Taming unified video diffusion models for consistent human image animation. *Science China Information Sciences*, 2025.

Zhenzhi Wang, Yixuan Li, Yanhong Zeng, Youqing Fang, Yuwei Guo, Wenran Liu, Jing Tan, Kai Chen, Tianfan Xue, Bo Dai, et al. Humanvid: Demystifying training data for camera-controllable human image animation. *Advances in Neural Information Processing Systems*, 37:20111–20131, 2024.

Haoning Wu, Zicheng Zhang, Weixia Zhang, Chaofeng Chen, Liang Liao, Chunyi Li, Yixuan Gao, Annan Wang, Erli Zhang, Wenxiu Sun, et al. Q-align: Teaching lmms for visual scoring via discrete text-defined levels. *arXiv preprint arXiv:2312.17090*, 2023.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, volume 2024, pp. 21875–21895, 2024.

Mingwang Xu, Hui Li, Qingkun Su, Hanlin Shang, Liwei Zhang, Ce Liu, Jingdong Wang, Yao Yao, and Siyu zhu. Hallo: Hierarchical audio-driven visual synthesis for portrait image animation, 2024a.

Sicheng Xu, Guojun Chen, Yu-Xiao Guo, Jiaolong Yang, Chong Li, Zhenyu Zang, Yizhong Zhang, Xin Tong, and Baining Guo. Vasa-1: Lifelike audio-driven talking faces generated in real time. *Advances in Neural Information Processing Systems*, 37:660–684, 2024b.

Zhongcong Xu, Jianfeng Zhang, Jun Hao Liew, Hanshu Yan, Jia-Wei Liu, Chenxu Zhang, Jiashi Feng, and Mike Zheng Shou. Magicanimate: Temporally consistent human image animation using diffusion model. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1481–1490. IEEE, 2024c.

Wenhao Yan, Sheng Ye, Zhuoyi Yang, Jiayan Teng, ZhenHui Dong, Kairui Wen, Xiaotao Gu, Yong-Jin Liu, and Jie Tang. Scail: Towards studio-grade character animation via in-context learning of 3d-consistent pose representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4450–4460, 2026.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6613–6623. IEEE, 2024.

Tianwei Yin, Qiang Zhang, Richard Zhang, William T Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22963–22974. IEEE, 2025.

Jiaming Zhang, Shengming Cao, Rui Li, Xiaotong Zhao, Yutao Cui, Xinglin Hou, Gangshan Wu, Haolan Chen, Yu Xu, Limin Wang, et al. Steadydancer: Harmonized and coherent human image animation with first-frame preservation. *arXiv preprint arXiv:2511.19320*, 2025.

Yue Zhang, Zhizhou Zhong, Minhao Liu, Zhaokang Chen, Bin Wu, Yubin Zeng, Chao Zhan, Yingjie He, Junxin Huang, and Wenjiang Zhou. Musetalk: Real-time high-fidelity video dubbing via spatio-temporal sampling. *arXiv preprint arXiv:2410.10122*, 2024.

Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. *Advances in Neural Information Processing Systems*, 36: 49842–49869, 2023.

Hongzhou Zhu, Min Zhao, Guande He, Hang Su, Chongxuan Li, and Jun Zhu. Causal forcing: Autoregressive diffusion distillation done right for high-quality real-time interactive video generation. *arXiv preprint arXiv:2602.02214*, 2026.

Shenhao Zhu, Junming Leo Chen, Zuozhuo Dai, Zilong Dong, Yinghui Xu, Xun Cao, Yao Yao, Hao Zhu, and Siyu Zhu. Champ: Controllable and consistent human image animation with 3d parametric guidance. In *European Conference on Computer Vision*, pp. 145–162. Springer, 2024.

Appendix

AI USE STATEMENT

Generative AI tools were used in this work only for proofreading and polishing the manuscript text; they played no role in the research itself. The authors have reviewed all AI-assisted work and take responsibility for the entire content of this work, including any text produced with the aid of generative AI.

A SYSTEM OPTIMIZATIONS

To meet the latency requirements of real-time streaming, we employ several infrastructure-level optimizations.

Ulysses sequence parallelism. We distribute computation across N GPUs using the Ulysses all-to-all pattern Fang & Zhao (2024): each GPU computes Q, K, V projections on its local token shard, then all-to-all communication redistributes to head-parallel layout where each GPU computes full-sequence attention for H/N heads, followed by a reverse all-to-all gather. This requires 2 all-to-all operations per layer but each involves only head-dimension-sized tensors. The FFN computation remains embarrassingly parallel on local token shards, and the KV cache is maintained independently per GPU head shard with no additional communication overhead.

torch.compile. All DiT blocks are compiled with `max-autotune-no-cudagraphs` mode. We pre-compile for three resolution buckets (480×480, 672×384, 384×672) with `dynamic=False` for maximum kernel efficiency.

B REAL-TIME DEPLOYMENT

At deployment, the incoming driving stream is divided into consecutive 12-frame chunks and processed by a three-stage pipeline distributed across four NVIDIA H100 GPUs: preprocessing on one GPU, VAE encoding and decoding on a second GPU, and DiT generation on the remaining two GPUs with two-way Ulysses sequence parallelism (Sec. A). To minimize inference overhead, we apply the operator fusion and computation-graph compilation of Sec. A throughout the pipeline. Each chunk is preprocessed, encoded by the causal VAE, denoised in three steps by the DiT using the current condition and the cached history, and immediately decoded without waiting for future frames. After generation, the clean KV states are cached for subsequent chunks: following PR-Sink (Sec. 3.4), the cache retains the first chunk as a static sink, a pose-retrieved dynamic sink, and a rolling window of recent chunks, bounding both memory and per-chunk computation for open-ended streams.

Figure 7 shows the stage-wise runtime at 480×480. Per chunk, preprocessing takes 1074 ms; VAE encoding and decoding take 102 and 166 ms; and on the DiT stage, motion encoding, three-step denoising, and clean KV-cache construction take 201, 447, and 138 ms, respectively. The 585 ms spent on denoising and KV-cache construction is consistent with the 611 ms DiT-loop latency reported in the main paper, and the additional 201 ms of motion encoding corresponds to the condition processing that the main-paper protocol excludes. Since all three stages run concurrently on consecutive chunks, the steady-state throughput is governed by the slowest stage: preprocessing occupies the full 1074 ms chunk launch period, while the VAE and DiT stages are active for only 268 and 786 ms per period and idle otherwise. The deployed system therefore emits one 12-frame chunk every 1074 ms, i.e., 11.2 FPS end-to-end at 480×480, with memory and per-chunk cost independent of the stream duration. Note that real-time streaming requires the generation throughput to match the playback rate of the output stream, not the 25-FPS sampling rate of the training data: at 11.2 FPS end-to-end, the system sustains real-time playback of an 11-FPS output stream, with the incoming driving signal subsampled accordingly.

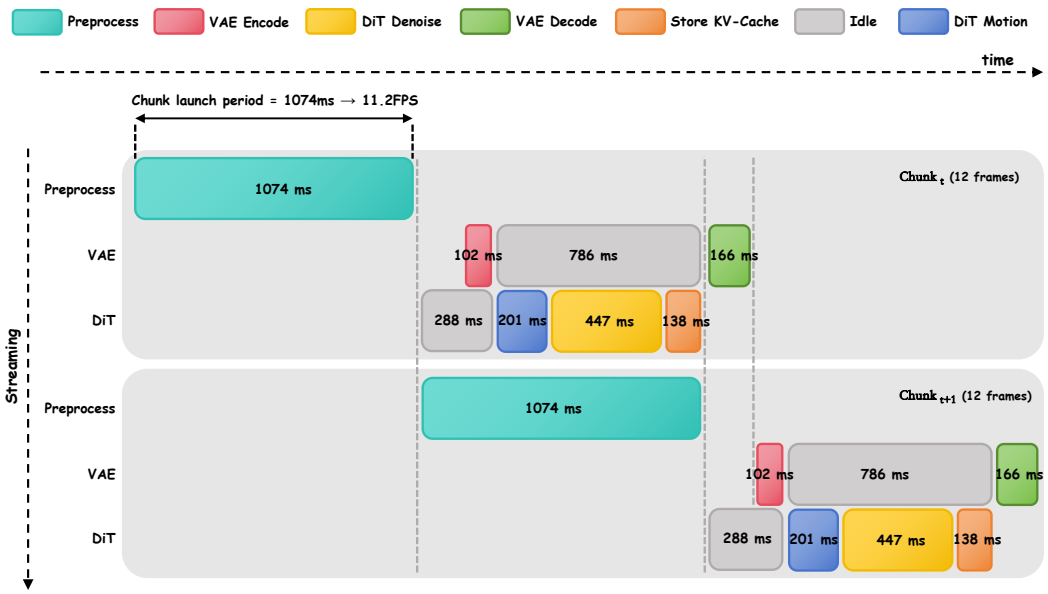

Figure 7: **Stage-wise runtime of the real-time deployment pipeline on four H100 GPUs at** 480×480**.** The incoming stream is processed in 12-frame chunks across three pipelined stages: preprocessing on one GPU, VAE encoding and decoding on one GPU, and DiT generation on two GPUs with Ulysses sequence parallelism. Colored bars denote active computation, and gray bars denote idle time while a stage waits for its input. Preprocessing occupies a full 1074 ms period per chunk and sets the steady-state chunk launch period, yielding 11.2 FPS end-to-end throughput.

C Gradient Analysis of Block-wise Distillation

We analyze why the block-wise replay procedure of Sec. 3.3 yields a well-defined training signal and why it drastically reduces memory. Let $z(\theta) = (z_1, \ldots, z_N)$ denote a self-forcing rollout in which every block is generated under student-induced history, $z_T = G_\theta(\epsilon_T; \mathrm{KV}(z_{<T}(\theta)), \mathcal{C}_T)$. Fully backpropagating through the rollout would differentiate

$$\mathcal{L}(\theta) = \mathcal{L}_{\mathrm{DMD}}(z(\theta); \mathcal{C}), \tag{9}$$

and the chain rule decomposes the gradient into per-block contributions,

$$\nabla_\theta \mathcal{L} = \sum_{T=1}^{N} \frac{\partial \mathcal{L}}{\partial z_T} \Bigg[\underbrace{\frac{\partial G_\theta}{\partial \theta}\Big|_T}_{\text{direct term}} + \underbrace{\sum_{s<T} \frac{\partial G_\theta}{\partial \mathrm{KV}_s} \frac{\partial \mathrm{KV}_s}{\partial z_s} \frac{\partial z_s}{\partial \theta}}_{\text{history term}} \Bigg]. \tag{10}$$

The history term couples block T to the parameters through every earlier block; it is exactly the term that requires retaining the activation graph of the entire trajectory.

The block-wise scheme instead optimizes the surrogate objective

$$\mathcal{L}_{\mathrm{BW}}(\theta) = \sum_{T=1}^{N} \mathcal{L}_T(\theta), \qquad \mathcal{L}_T(\theta) = \mathcal{L}_{\mathrm{DMD}}\left(\tilde{z}^{(T)}; \mathcal{C}\right), \tag{11}$$

with $\tilde{z}^{(T)} = [\mathrm{sg}(\bar{z}_1), \ldots, z_T^\theta, \ldots, \mathrm{sg}(\bar{z}_N)]$ as defined in Sec. 3.3. Since only z_T^θ carries a gradient inside $\mathcal{L}_T$,

$$\nabla_\theta \mathcal{L}_{\mathrm{BW}} = \sum_{T=1}^{N} \frac{\partial \mathcal{L}_T}{\partial z_T^\theta} \frac{\partial G_\theta}{\partial \theta}\Big|_T. \tag{12}$$

Comparing Eq. equation 10 and Eq. equation 12, the sum of the per-block backpropagations performed in the second pass is *exactly* the gradient of Eq. equation 11: it reproduces the direct term

of the full gradient at every block position and drops only the history terms. Because the teacher and critic always evaluate a complete N-block trajectory, the distribution-matching signal itself is unchanged; only the student's gradient path is truncated. The truncation is also what keeps training aligned with inference: every block is always regenerated under student-induced history, so dropping the history gradient does not reintroduce a teacher-forcing exposure gap.

Memory. Let M_{blk} be the activation memory of one denoising block of the 14B student and M_{state} the storage of one block's detached latent and KV state. Full rollout backpropagation keeps all N block graphs simultaneously, requiring $M_{\mathrm{full}} \approx N\, M_{\mathrm{blk}}$. The first pass of the block-wise scheme tracks no gradients and stores only N detached states, and the second pass holds exactly one block graph at a time, giving

$$M_{\mathrm{BW}} \approx M_{\mathrm{blk}} + N\, M_{\mathrm{state}} \;\ll\; M_{\mathrm{full}}. \tag{13}$$

Since a block graph contains all layer activations whereas a state is a handful of latent tensors, $M_{\mathrm{blk}} \gg M_{\mathrm{state}}$ and the peak memory is reduced by roughly a factor of N. Together with LoRA (rank 128), which shrinks gradient and optimizer-state memory, Stage-2 distillation of the 14B student fits on a single 8×80GB GPU node.

D Implementation Details

Training. Both training and inference support three aspect-ratio buckets: 480×480, 384×672, and 672×384 pixels. Training proceeds in two stages on 8 NVIDIA H100 GPUs. In the first stage, we train for 10,000 steps with a learning rate of 5×10^{-5}. In the second stage, we continue training for 20,000 steps with identical LoRA parameters, using learning rates of 1×10^{-5} for both the generator and the critic.

Latency and throughput measurement. Unless otherwise stated, the latency and throughput numbers reported in the main paper measure the DiT generation loop only, rather than end-to-end wall-clock time. Condition processing (e.g., reference and pose encoding) and VAE encoding/decoding are excluded from the measurement, as their costs are either one-time or block-independent, can be overlapped on separate devices, and vary considerably across implementations. Since DiT forward passes dominate the recurrent per-block cost, this protocol isolates the primary bottleneck and provides a fair basis for comparison across systems. End-to-end throughput of the full deployment pipeline is reported separately in Sec. B.

E Long-Video Baseline Protocol

The official implementations of SCAIL and UniAnimate-DiT operate on fixed-length clips using the temporal context available during training and do not provide native long-form generation. For a controlled long-sequence comparison, we extend both methods with the same training-free sliding-window denoising procedure. We initialize a single noise latent spanning the complete target sequence. At every denoising step, we slide each model's native temporal context window across this latent with a fixed stride. Each window is conditioned on the same reference identity image and the temporally aligned local segment of the driving pose sequence. Predictions in overlapping regions are fused using triangular, or tent-shaped, weights that emphasize the temporal center of each window and suppress its boundaries. A single UniPC multistep ODE-solver update Zhao et al. (2023) is then applied to the fused full-sequence prediction, ensuring that the solver state evolves consistently along one global trajectory rather than independently within each window. This extension requires neither retraining nor positional-embedding extrapolation and is used uniformly for SCAIL and UniAnimate-DiT. EverAnimate and One-to-All are instead evaluated using the long-video generation procedures provided by their official implementations.

F Sequence-Parallel Scaling

Table 3 evaluates Ulysses scaling on the same three-minute benchmark. Moving from one to two H100 GPUs increases throughput from 12.41 to 19.63 FPS, yielding a $1.58\times$ speedup at 79.1% parallel efficiency. Expanding to four GPUs improves throughput only to 22.13 FPS: the speedup

saturates at $1.78\times$ and efficiency falls to 44.6%, as the additional all-to-all communication increasingly offsets the reduction in per-GPU attention computation. We therefore adopt two-way sequence parallelism by default, as it offers the best trade-off between throughput and GPU count.

Table 3: **Scaling efficiency of Ulysses sequence parallelism on H100 GPUs at** 480×480**.** Results are measured on the three-minute benchmark and cover the DiT generation loop only. Latency denotes the time required to generate one 12-frame block. Speedup is relative to one GPU, and efficiency is speedup divided by the number of GPUs.

GPUs	Latency (ms)	FPS	Speedup	Efficiency
1	966.96	12.41	1.00×	100.0%
2	611.28	19.63	1.58×	79.1%
4	542.25	22.13	1.78×	44.6%

G MOTION-FOLLOWING EVALUATION

The metrics in the main paper measure visual quality and identity; this section measures whether the generated motion actually follows the driving pose. To avoid self-evaluation, we do not reuse our conditioning preprocessor but run the same external DWPose WholeBody detector **?** on both the driving and the generated videos.

Protocol. Frames are sampled at 2 FPS on an absolute time grid starting at 0 s, and only the time span covered by both sides is compared (no stretching to the total duration); in multi-person frames the subject with the highest area-confidence score is selected, with a detector confidence threshold of 0.3. The body metric uses the 17 COCO-WholeBody body keypoints plus 6 foot keypoints (23 per frame); face keypoints are excluded so that expression and identity differences do not dominate, and the 42 hand keypoints are reported separately. AKD is computed only on keypoints detected on both sides, with coordinates normalized by each image's size and then by the driving subject's bounding-box scale. PCK@τ counts a keypoint as correct within $\tau \in \{5\%, 10\%\}$ of the driving box scale; keypoints visible on the driving side but missed on the generated side count as errors, and MKR reports the fraction of such misses. All numbers are macro averages over the 24 sequences of the three-minute benchmark. This protocol follows the external-detector AKD/MKR convention of FOMM Siarohin et al. (2019) and standard PCK practice.

Table 4: **Motion-following accuracy measured by an external pose detector on the three-minute benchmark.** Body and foot keypoints (23 per frame), macro-averaged over the 24 benchmark sequences. AKD is normalized by the driving subject's bounding-box scale; PCK thresholds are 5%/10% of that scale; MKR is the rate of driving-visible keypoints missed in the generated video.

Method	AKD ↓	PCK@0.05 (%) ↑	PCK@0.10 (%) ↑	MKR (%) ↓
EverAnimate	0.3142	22.29	38.24	3.43
One-to-All	0.2876	33.86	45.36	27.60
SCAIL	0.2887	19.37	38.12	2.99
UniAnimate-DiT	0.2926	23.00	38.04	2.55
Wan2.2-Animate	0.1966	**45.39**	**61.06**	5.67
LiveAnimate	0.2923	20.64	41.15	**2.30**

Findings. Three observations emerge from Table 4. First, LiveAnimate attains the lowest MKR (2.30%): its generated bodies remain detectable and complete even in the final minute, consistent with the stability results in the main paper, whereas One-to-All loses 27.60% of driving-visible keypoints, mirroring its visible quality collapse. Second, on PCK@0.10 LiveAnimate (41.15%) matches or exceeds EverAnimate, SCAIL, and UniAnimate-DiT (38.0–38.2%) while being the only system that runs in real time; Wan2.2-Animate retains higher motion precision (61.06%) but requires hours of offline computation per sequence and loses identity over long rollouts (Figure 4). Third, hand keypoints remain difficult for all methods: LiveAnimate obtains 18.21% hand PCK@0.10 versus 45.35% for Wan2.2-Animate, though with the lowest hand MKR (5.04%).

Breaking LiveAnimate down by temporal segment, body PCK@0.10 decreases from 51.08% in the first 10 s to 33–35% in later segments, while MKR stays flat (2.1–2.6%): poses remain detectable throughout the rollout, and the moderate precision drop partly reflects changing motion difficulty across segments rather than accumulated drift alone. Shifting the generated stream by ± 5 frames (± 0.2 s) changes PCK@0.10 by at most 0.4 points (35.40/36.13 versus 35.76), so these numbers are not an artifact of sub-second timing misalignment. Finally, in the ablation study, removing DMD distillation or the Dynamic Sink lowers PCK@0.10 by about 2 points (33.88% and 33.95% versus 35.96% for the full model), while the remaining variants stay within 0.5 points, indicating that both the distillation objective and the retrieved pose context contribute to motion following.

H Additional Qualitative Results

We provide seven additional three-minute examples to complement the qualitative comparisons in the Experiments section of the main paper. Figures 8 and 9 cover full-body and upper-body animation under diverse identities, clothing, backgrounds, camera framing, and motion patterns. For each example, we show the reference image together with the driving pose signal and generated frame at 20-second intervals from 20 to 180 seconds. These uniformly sampled results expose temporal changes throughout the rollout rather than selecting only adjacent or early frames.

The first two examples in Figure 8 contain large changes in limb configuration and body orientation, whereas the last two emphasize upper-body motion, facial variation, and gestures near the face. Despite these differences, the generated subjects remain recognizable and their clothing and backgrounds do not exhibit progressive changes over the sampled timestamps. Figure 9 further tests appearance preservation across a dim indoor scene, a bright outdoor scene, and a close-up portrait. The outputs track both repeated and newly observed poses while maintaining scene-specific appearance. Together with the quantitative three-minute evaluation in the main paper, these examples provide additional qualitative evidence that LiveAnimate supports stable long-form streaming across varied motion and visual conditions.

I Limitations and Future Work

LiveAnimate currently generates at 480×480 with three denoising steps, which limits visual fidelity and resolution, and it does not support multi-person scenes or large camera motion. Extending LiveAnimate toward higher-resolution streaming generation, multi-person animation, and camera-dynamic scenes is left to future work.

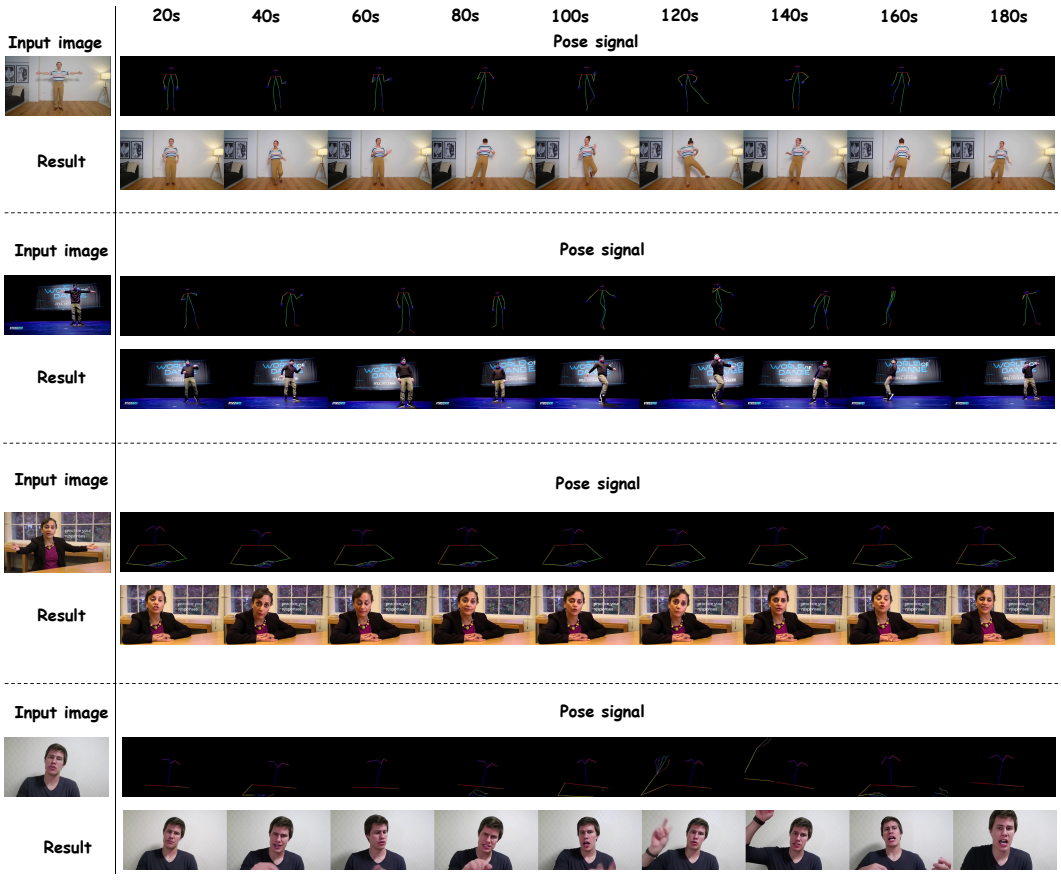

Figure 8: **Additional qualitative results on four three-minute sequences.** For each example, the left column shows the reference image, while the rows on the right show the driving pose signals and corresponding outputs sampled every 20 seconds from 20 to 180 seconds. The examples include full-body dance sequences with large pose changes and upper-body sequences dominated by facial expressions and hand motion. Across the rollout, LiveAnimate follows the input poses while preserving the reference identity, clothing, and scene appearance.

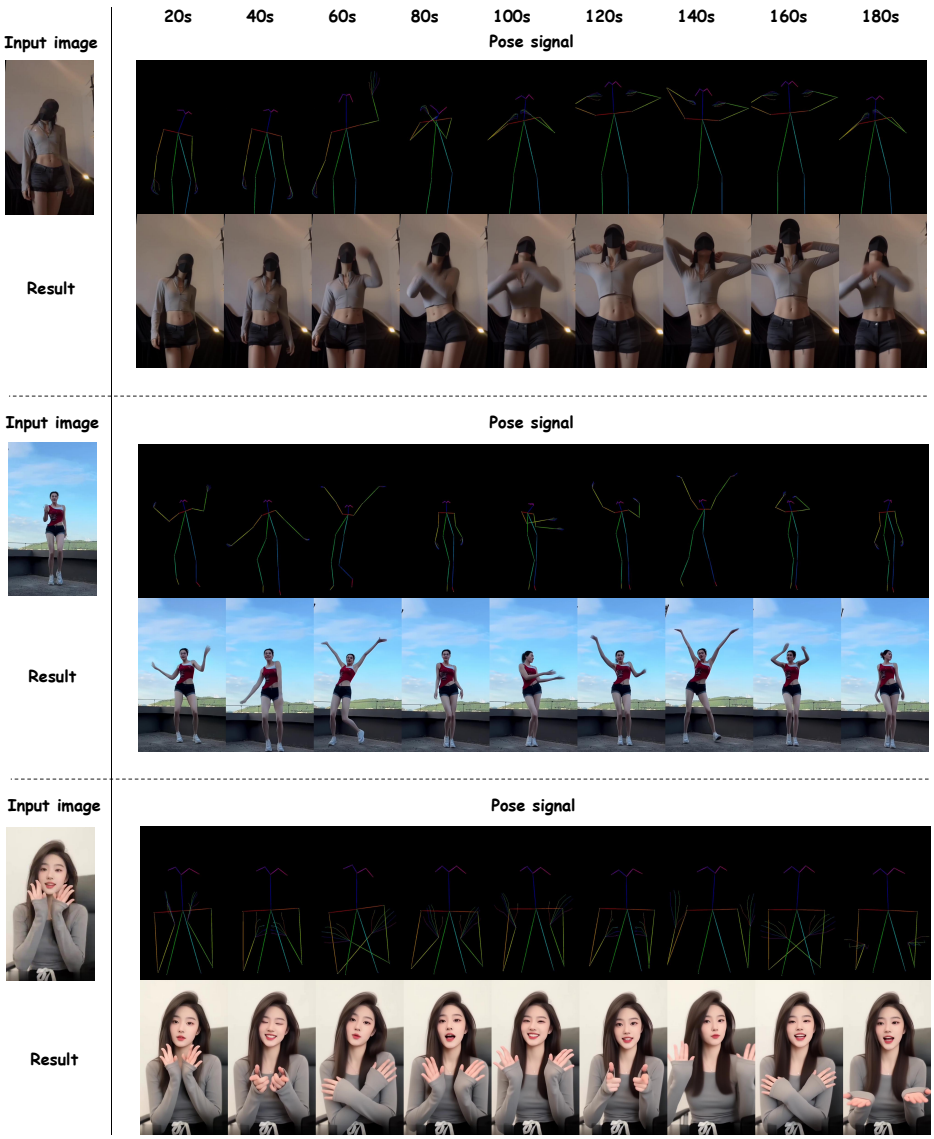

Figure 9: **Additional qualitative results on three three-minute sequences.** We show the reference image, driving pose signal, and generated outputs at 20-second intervals. The sequences cover indoor full-body motion, an outdoor dance sequence, and an upper-body subject performing frequent hand gestures. LiveAnimate maintains the subject's appearance and background while responding to substantial changes in body configuration over the three-minute rollout.