EDITALIVE! UNIFIED CHARACTER VIDEO EDITING FOR LIVE STREAMING

Anonymous authors
Paper under double-blind review

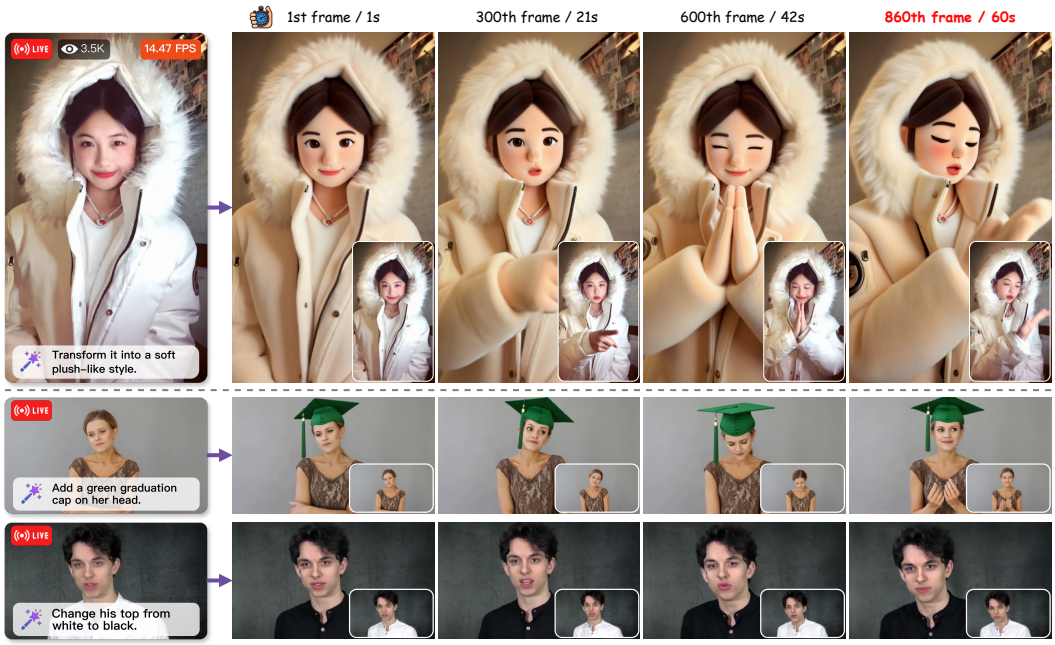


Figure 1: An overview of the editing results and inference efficiency of EditaLive. EditaLive performs diverse local and global appearance edits in real time while faithfully preserving source motion and maintaining a consistent edited appearance over long sequences. Its appearance–motion decoupled formulation further supports cross-character editing.

ABSTRACT

Conventional video editing primarily focuses on scene-level content, whereas live streaming places greater emphasis on the human subject. However, directly applying existing video-editing methods to human-centric live streaming remains challenging, as they may introduce facial-expression inconsistencies and typically depend on multiple offline inference steps, making them unsuitable for real-time interaction. We propose EditaLive, a novel framework for real-time streaming character video editing. In detail, we start from a pretrained image animation model (Wan-Animate (Cheng et al., 2025)), which naturally decouples appearance from motion, and repurpose it as the base model for instruction-based human-centric video editing by reference frame editing and video reconstruction via the collected CharEdit-50K dataset. Besides, we adapt the model from offline bidirectional to causal streaming generation, and design an aligned self-rollout distillation strategy that compresses the model into a two-step sampler, where fixed RoPE and align forcing reduce training–inference discrepancies, and first-frame preserved sparse attention filters redundant historical information to mitigate appearance drift. Extensive experiments demonstrate that EditaLive delivers state-of-the-art editing performance with faithful preservation of facial expressions and low-latency real-time streaming inference. The code will be publicly available.

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.

1 INTRODUCTION

Live streaming has become one of the most popular forms of online entertainment, with streamers altering their on-screen looks to express their personalities and capture viewers' attention. In the era of generative AI, we define and tackle this task as *streaming character video editing*, enabling streamers/viewers to customize live-streaming content using simple text prompts.

However, most of the current video editing methods are designed for the scene-level or object-level content. In detail, early editing approaches (Xing et al., 2024; Chen et al., 2025; Kodaira et al., 2025; Liang et al., 2025) apply image editing models to individual video frames, inevitably leading to temporal flickering. In contrast, recent native video editing algorithms (Qi et al., 2023; Jiang et al., 2025; Team, 2025; Bai et al., 2026) perform joint video-level editing, substantially improving temporal consistency and editing quality, as shown in Fig. 2 (a). Despite their impressive performance, directly extending native video editing algorithms to character editing in live streaming scenarios is far from straightforward. This stems from three primary challenges: *(i) Facial expression mismatch*. Current video editing datasets (Wu et al., 2025b; He et al., 2025; Bai et al., 2026; Huang et al., 2026a) are predominantly synthesized using video generation and editing models (Jiang et al., 2025; Wan et al., 2025), inevitably inheriting facial expression inconsistencies from these models, as shown in Fig. 4 (left). Consequently, models trained on such datasets struggle to faithfully preserve facial expressions. *(ii) Offline inference paradigm*. Existing instruction-based video editing models (Wei et al., 2025; Ju et al., 2025; Team, 2025; Bai et al., 2026) typically require dozens of denoising steps together with classifier-free guidance (CFG) (Ho & Salimans, 2021) to achieve high editing quality, while relying on bidirectional temporal attention over complete video clips to maintain temporal consistency. These offline-oriented designs make existing video editing models unsuitable for live streaming scenarios. *(iii) Long-term drift*. Although recent methods (Wang et al., 2026; Zhao et al., 2026) adopt Self Forcing (Huang et al., 2026b) for real-time streaming video editing, their training and inference procedures remain misaligned in terms of RoPE indexing and KV-cache construction. The resulting errors accumulate during autoregressive inference, causing progressive appearance drift and undermining editing stability over long character video sequences.

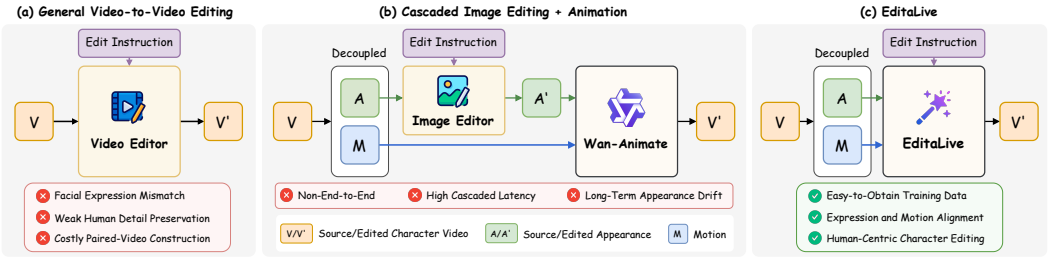


Figure 2: Comparison of character video editing paradigms.

We thus propose EditaLive, a unified framework for real-time streaming character video editing built on three key components. *(i) Appearance–Motion Decoupled Editing*. We build EditaLive on Wan-Animate (Cheng et al., 2025), a pretrained image animation model that naturally decouples character appearance from motion conditions. We repurpose this model for instruction-based character video editing by combining reference-frame editing with video reconstruction, as shown in Fig. 2 (c). For each source video, we edit its reference frame with a forward instruction and construct a corresponding reverse instruction to recover the original appearance. Conditioned on the edited reference, reverse instruction, and source motion signals, EditaLive is trained to reconstruct the source video, thereby learning diverse appearance transformations from paired edited–source appearances. Based on this formulation, we construct CharEdit-50K, a comprehensive dataset covering diverse character appearance editing tasks. *(ii) Causal Streaming Adaptation*. Inspired by causal and autoregressive video generation methods (Chen et al., 2024; Yin et al., 2025), we subsequently adapt the offline editing model to chunk-wise causal streaming generation, preserving bidirectional attention within each chunk while restricting cross-chunk attention to preceding context. *(iii) Aligned Self-Rollout Distillation*. After adapting the model for causal streaming generation, we further employ self-rollout distillation (Huang et al., 2026b) to compress it into a two-step sampler. Although self-rollout mitigates exposure bias (Schmidt, 2019; Ning et al., 2024) by training the model on its own autoregressive predictions, training–inference discrepancies in positional encoding and KV-cache construction can still lead to unstable long-term streaming generation. To address this, we introduce Fixed RoPE for consistent positional encoding and Align Forcing for inference-aligned KV-cache

construction. Furthermore, we propose First-frame Preserved Sparse Attention (FPSA) to filter out redundant historical information while keeping the first-frame features fully visible, effectively mitigating appearance drift during long-term generation. Extensive quantitative and qualitative results show that EditaLive achieves superior editing quality, expression consistency, and long-term stability, while enabling real-time streaming inference. Our contributions can be summarized as:

- We propose EditaLive, a unified framework for real-time streaming character video editing that supports both diverse character appearance editing tasks and motion-driven animation.
- We introduce an appearance–motion decoupled editing paradigm that reframes character video editing as appearance transformation under explicit motion control, thereby preserving facial expressions. We further design a reconstruction-based training strategy built on motion-aligned real-video supervision and construct CharEdit-50K for training.
- We develop an aligned self-rollout distillation strategy equipped with fixed RoPE, align forcing, and first-frame preserved sparse attention, effectively reducing training–inference discrepancies and mitigating appearance drift during long-term generation.

2 Related Work

Instruction-based Image and Video Editing. Recent advances in text-to-image diffusion models (Rombach et al., 2022; Peebles & Xie, 2023) have significantly improved instruction-based image editing, enabling users to modify visual content through natural language instructions (Brooks et al., 2023; Zhang et al., 2023; Sheynin et al., 2024; Zhang et al., 2026b). Some works extend image editing to videos by applying image editing models to individual frames (Xing et al., 2024; Chen et al., 2025; Kodaira et al., 2025; Liang et al., 2025). While efficient, these frame-wise methods lack stable temporal modeling and often lead to flickering and inconsistent edits. Recent instruction-based video editing methods instead perform joint video-level editing with native video diffusion models (Qi et al., 2023; Jiang et al., 2025; Wei et al., 2025; Ju et al., 2025; Team, 2025; Bai et al., 2026), achieving improved temporal consistency and editing quality. However, these methods are mainly designed for offline clip-level editing, relying on multi-step sampling and bidirectional temporal modeling over complete videos. They also commonly depend on synthetic editing datasets (Wu et al., 2025b; Bai et al., 2026), which can introduce facial expression inconsistencies in character-centric scenarios. More recent studies (Wang et al., 2026; Zhao et al., 2026) adopt Self Forcing (Huang et al., 2026b) to enable real-time causal streaming video editing. Nevertheless, they primarily focus on streaming efficiency and do not explicitly address stability over long generation horizons. Unlike these methods, we focus on real-time streaming character video editing with explicit motion preservation and long-term generation capability.

Motion-driven Character Animation. Motion-driven character animation aims to animate a source character according to motion signals extracted from a driving video. To achieve this, recent methods condition video generation on reference images and diverse motion representations, such as skeleton sequences (Hu, 2024; Zhang et al., 2025a), 3D keypoints (Yan et al., 2026b), motion frames (Yan et al., 2026a), and implicit facial representations (Zhao et al., 2025; Cheng et al., 2025). These works validate the effectiveness of separating appearance and motion for character generation. Building on this, recent efforts such as PersonaLive (Li et al., 2026) further achieve real-time portrait animation through efficient distillation and micro-chunk streaming generation. However, most existing methods focus on animation or reenactment, where the source appearance is assumed to be fixed, rather than instruction-based character appearance editing. In contrast, we apply the idea of appearance–motion decoupling to character video editing, enabling appearance editing under explicit motion control while naturally supporting motion-driven animation by swapping character images.

Efficient and Streaming Video Generation. Efficiency and streaming capability are critical for deploying video generation models in interactive and real-time scenarios. Existing methods commonly accelerate inference through model quantization (Feng et al., 2025b; Xie et al., 2025), sparse attention (Zhang et al., 2026a; Shao et al., 2026), and sampling-step distillation (Yin et al., 2024b;a). To achieve real-time streaming video generation, recent works further introduce autoregressive paradigms (Chen et al., 2024; Ma et al., 2026). CausVid (Yin et al., 2025) adapts diffusion video generation to causal inference. Self Forcing (Huang et al., 2026b) trains the model with its own autoregressive predictions to reduce exposure bias. In this work, we adapt autoregressive streaming generation to character video editing and develop an aligned self-rollout distillation strategy, enabling efficient two-step inference while improving long-term appearance stability.

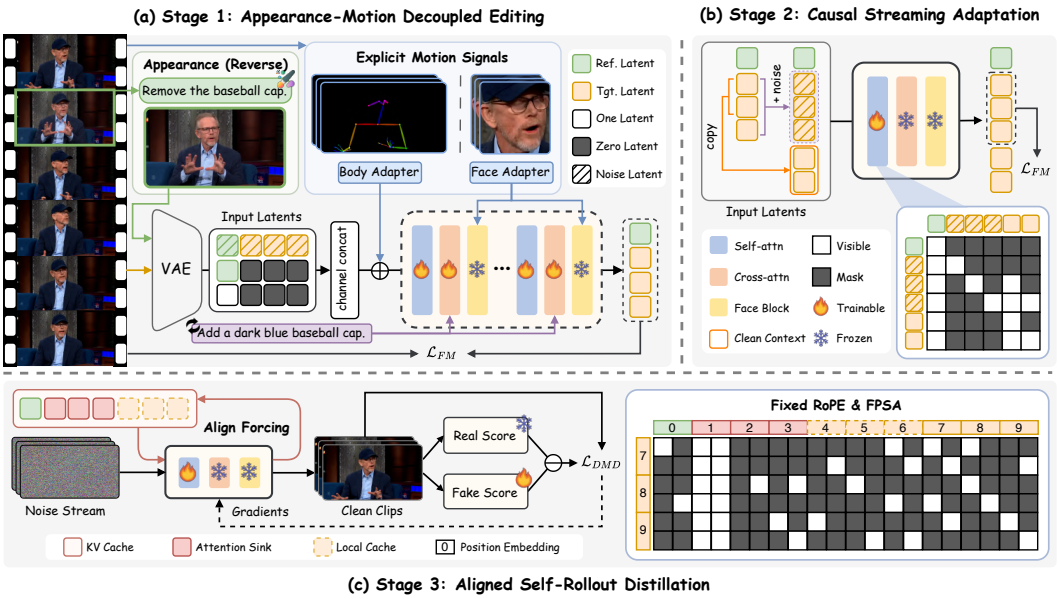


Figure 3: Overview of the three-stage training pipeline of EditaLive. (a) Appearance–Motion Decoupled Editing reformulates character video editing as appearance editing with explicit body and facial motion preservation. (b) Causal Streaming Adaptation converts bidirectional modeling into causal generation using clean historical context and causal attention. (c) Aligned Self-Rollout Distillation compresses the model into a two-step sampler, where Align Forcing and Fixed RoPE align training with streaming inference, and FPSA improves long-term appearance stability.

3 METHODS

In this paper, we define a *character video* as a video centered on a primary human or human-like subject. Streaming character video editing aims to modify the appearance of the primary character in real time according to a user instruction. Formally, given a source video stream $x^{1:N}$ and an editing instruction p, the goal is to generate an edited stream $y^{1:N}$:

$$y^i = \mathcal{F}(x^i, p, y^{<i}), i = 1, 2, \ldots, N, \tag{1}$$

where $\mathcal{F}$ denotes the editing model and $y^{<i}$ represents previously generated outputs. As illustrated in Fig. 3, EditaLive addresses streaming character video editing through three consecutive stages. First, we decompose the source character video into appearance and motion representations, reformulating character video editing as appearance editing conditioned on the explicit motion signals (Sec. 3.1). Second, we adapt the offline bidirectional video model to causal streaming generation, enabling continuous inference based on cached historical context (Sec. 3.2). Finally, we develop an aligned self-rollout distillation strategy that compresses the causal model into a two-step sampler while reducing training–inference discrepancies and improving long-term stability (Sec. 3.3). Together, these components form a unified framework for efficient streaming character editing with explicit motion preservation and stable long-term generation.

3.1 APPEARANCE–MOTION DECOUPLED EDITING

As shown in Fig. 3 (a), we reformulate character video editing by decoupling the appearance to be edited from the motion to be preserved. Given a source character video $x^{1:N}$, we select a reference frame I to represent the character appearance, while extracting a skeleton sequence $s^{1:N}$ and implicit facial representations $f^{1:N}$ to capture the source body motion and facial dynamics, respectively. The editing process is then formulated as

$$y^{1:N} = \mathcal{F}(I, p, s^{1:N}, f^{1:N}), \tag{2}$$

where the reference image I and instruction p jointly determine the edited appearance.

Reconstruction-based Training. Synthetic source–target video pairs often exhibit facial expression mismatches (Fig. 4 (left)), entangling the desired appearance edit with unintended changes in facial

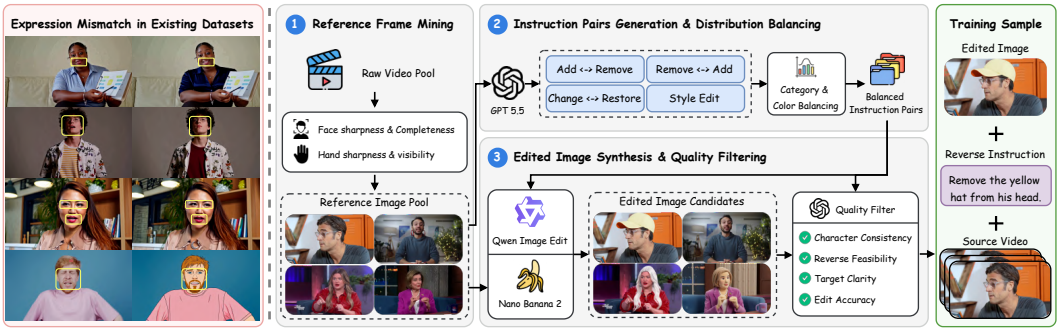


Figure 4: Construction pipeline of CharEdit-50K. (Left) Existing video editing datasets often contain facial expression inconsistencies. (Right) Our three-stage data pipeline.

dynamics. We therefore restrict synthetic editing to the reference image while retaining the real video as the reconstruction target. Specifically, given the reference image I, we first apply a forward instruction p^{fwd} via an off-the-shelf image editing model $\mathcal{E}_{\text{img}}$ to obtain an edited image:

$$\widetilde{I} = \mathcal{E}_{\text{img}}(I, p^{\text{fwd}}). \tag{3}$$

We then construct a reverse instruction p^{rev} to restore the original appearance from $\widetilde{I}$. Next, we encode the edited image $\widetilde{I}$ and the original video $x^{1:N}$ into the VAE latent space, obtaining the reference latent z^{ref} and target video latents z^{tgt}, respectively. We define $Z_1 = (z^{\text{ref}}, z^{\text{tgt}})$ as the clean reference–target sequence and sample $Z_0 \sim \mathcal{N}(0, \mathbf{I})$. At timestep $t \in [0, 1]$, we construct

$$Z_t = tZ_1 + (1-t)Z_0 = (z_t^{\text{ref}}, z_t^{\text{tgt}}), \tag{4}$$

where z_t^{ref} and z_t^{tgt} denote the noised reference and target video latents, respectively. The model input H_t is constructed as

$$H_t = (z_t^{\text{ref}}, z_t^{\text{tgt}}) \,\|\, (z^{\text{ref}}, \mathbf{0}, \ldots, \mathbf{0}) \,\|\, (\mathbf{1}, \mathbf{0}, \ldots, \mathbf{0}), \tag{5}$$

where $(\cdot, \cdot)$ and $\|$ denote temporal and channel-wise concatenation, respectively. The model v_θ predicts the ground-truth flow velocity $V_t = \frac{\mathrm{d}Z_t}{\mathrm{d}t} = Z_1 - Z_0$ and is optimized by

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{Z_1, Z_0, t}\left[\left\| v_\theta\left(H_t, t \mid p^{\text{rev}}, s^{1:N}, f^{1:N}\right) - (Z_1 - Z_0)\right\|_2^2\right]. \tag{6}$$

By treating the original video latent z^{tgt} as the reconstruction target, $\mathcal{L}_{\text{FM}}$ forces the model to follow p^{rev} and recover the original appearance from the edited reference $\widetilde{I}$.

CharEdit-50K. Following the reconstruction-based formulation, we construct CharEdit-50K through the three-stage pipeline shown in Fig. 4 (right). First, we mine high-quality reference frames from a large collection of character videos. Second, GPT-5.5 (Singh et al., 2025) generates candidate bidirectional instruction pairs for each reference image. Third, Qwen-Image-Edit (Wu et al., 2025a) and Nano Banana 2 (Team et al., 2023) synthesize the corresponding edited images, which are subsequently filtered by GPT-5.5. Each retained sample contains an edited image, its reverse instruction, and the corresponding source video. Further details are provided in Appendix B.1.

3.2 Causal Streaming Adaptation

While bidirectional self-attention in the previous stage enables temporally consistent editing, it requires complete video clips as input and incurs computational and memory costs that grow rapidly with video length, making it unsuitable for streaming inference. We therefore adapt the model to chunk-wise causal generation, where frames interact bidirectionally within each chunk while different chunks are temporally ordered, as shown in Fig. 3 (b). Let $z^{\text{tgt}} = (z^{(1)}, \ldots, z^{(K)})$ denote the target video latents divided into K chunks. At timestep t, we perturb the target video latents to obtain z_t^{tgt}, while keeping the reference latent z^{ref} clean. The input is constructed as

$$H_t^{\text{causal}} = (z^{\text{ref}}, z_t^{\text{tgt}}, z^{\text{context}}) \,\|\, (z^{\text{ref}}, \mathbf{0}, \ldots, \mathbf{0}) \,\|\, (\mathbf{1}, \mathbf{0}, \ldots, \mathbf{0}), \tag{7}$$

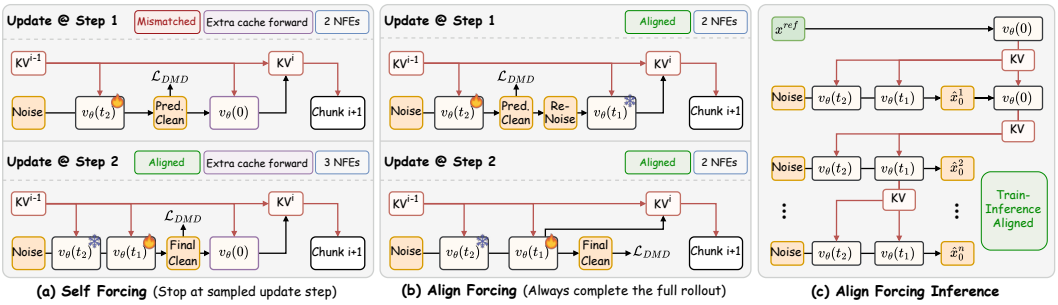


Figure 5: Comparison of Self-Forcing and Align Forcing under two-step distillation.

where $z^{\text{context}} = (z^{(1)}, \ldots, z^{(K-1)})$ denotes the clean context. Unlike stage 1 in Sec. 3.1, the reference latent in the first term is directly used as clean context rather than being perturbed. We apply a chunk-wise causal attention mask such that each noisy target chunk attends to the clean reference latent, its own latents, and the clean context chunks preceding it, but not to other noisy target chunks or future context. This design enables all target chunks to be trained in parallel while preserving the causal dependencies required for streaming inference. The Flow Matching (Lipman et al., 2022) loss is computed over the positions corresponding to z_t^{tgt}, while the reference and clean-context positions are excluded from the loss and used solely as conditioning.

3.3 Aligned Self-Rollout Distillation

Although causal adaptation enables streaming inference, the model still requires multi-step sampling and suffers from error accumulation during long-term generation. Building on Self Forcing (Huang et al., 2026b), we propose an aligned self-rollout distillation strategy that reduces the sampling process to two steps and matches rollout training with streaming inference, as shown in Fig. 3 (c). Specifically, Align Forcing eliminates discrepancies in KV-cache construction, while Fixed RoPE maintains consistent positional encoding between training and inference. We further introduce First-frame Preserved Sparse Attention to filter redundant historical context and mitigate appearance drift.

Align Forcing. As shown in Fig. 5 (a), Self Forcing (Huang et al., 2026b) couples the randomly selected gradient-update step with rollout termination. Once the selected step is reached, the rollout stops and the clean estimate at that step is used to construct the KV cache propagated to subsequent chunks. During streaming inference, however, each chunk completes the entire denoising trajectory, and only the final prediction is used to construct the propagated state. This creates a training–inference mismatch in inter-chunk state construction: training may propagate an intermediate rollout state, whereas inference always propagates the final state. To eliminate this discrepancy, we decouple gradient update from rollout termination. As shown in Fig. 5 (b), gradients are computed only at the selected step, while the remaining denoising steps are completed with stop-gradient. The KV cache is then constructed from the final prediction and propagated to subsequent chunks, always matching the inference stage (Fig. 5 (c)). Under two-step distillation, Self Forcing requires an average of 2.5 NFEs per training iteration, whereas Align Forcing requires 2 NFEs, with only the first chunk incurring one additional t_0 cache forward to initialize the attention sink.

Fixed RoPE. Standard RoPE assigns temporal indices according to the absolute latent positions. Since self-rollout training covers only a finite number of chunks, long-term inference eventually encounters indices outside the training range. Meanwhile, the reference sink remains at position 0, causing its positional offsets from later latents to grow continuously and progressively weakening appearance conditioning. We therefore introduce Fixed RoPE, which applies the same positional layout 0–9 during both rollout training and streaming inference, as shown in Fig. 3 (c). Specifically, the reference sink, attention sink, local-window context, and current chunk are assigned indices 0, 1–3, 4–6, and 7–9, respectively. By assigning positions according to their roles in the streaming context rather than their absolute timestamps, Fixed RoPE avoids positional extrapolation and keeps the appearance reference at a fixed distance from the latents currently being generated.

First-frame Preserved Sparse Attention. Since the local-window context is temporally and visually closest to the current chunk, the current queries tend to over-rely on it while underutilizing the stable appearance cues in the reference and attention sinks. Inspired by VSA (Zhang et al.,

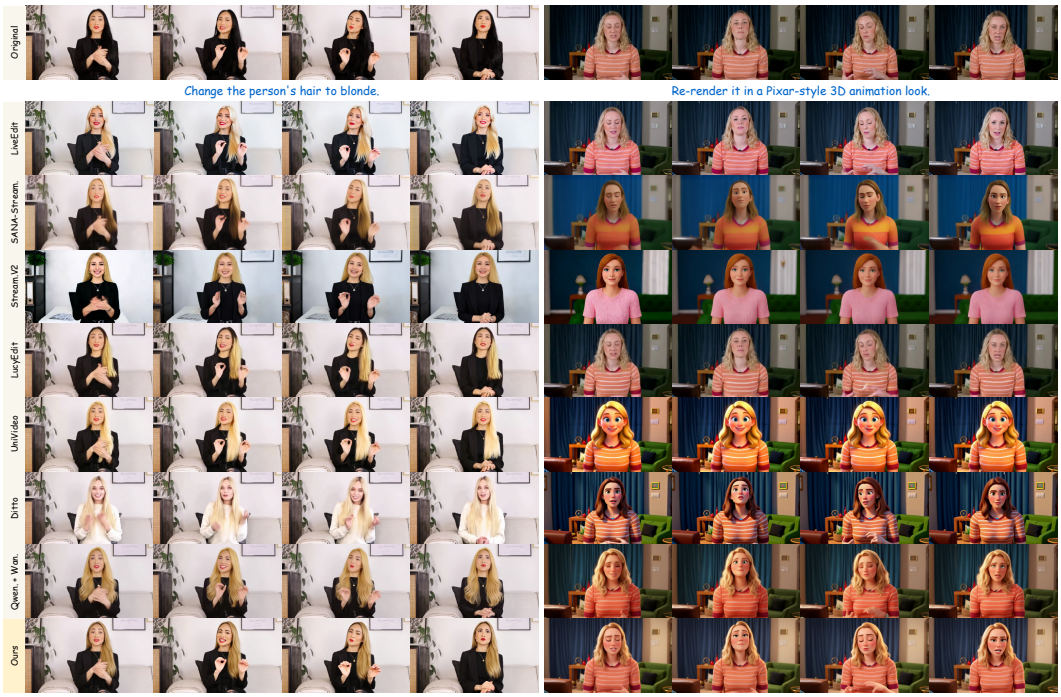


Figure 6: Qualitative comparisons. EditaLive faithfully follows the editing instructions while preserving the source facial expressions, body poses, and scene structure across frames.

2026a), we partition the current query and cached key tokens into spatio-temporal blocks, and compute block-level query–key scores. For each query block, we retain the top-K key blocks, where $K = (1 - \rho)B$, B denotes the total number of candidate key blocks and ρ represents the sparsity ratio. Token-level attention over the original queries, keys, and values is then computed only within the selected block pairs. Since this content-adaptive selection may still favor the local window, we always include all key blocks from the first generated frame in the selected K blocks and choose the remaining blocks according to their scores, as shown in Fig. 3 (c). As a stable realization of the target appearance, the first frame serves as a persistent appearance anchor, thereby mitigating appearance drift during long-term generation.

4 Experiments

4.1 Implementation Details

We build our model on Wan-Animate (Cheng et al., 2025) and train all three stages on CharEdit-50K using LoRA, with both the rank r and scaling factor α set to 128. Experiments are conducted on 8 NVIDIA H100 GPUs. All stages use the AdamW optimizer with a global batch size of 8. Stages 1 and 2 are each trained on 33-frame, 480p video clips for 15K steps with a learning rate of 1×10^{-4}, while Stage 3 is trained on 69-frame, 480p clips for 2K steps with a learning rate of 2×10^{-6}. The Stage 3 student is initialized from the Stage 2 checkpoint, whereas its real-score and fake-score branches are initialized from the Stage 1 checkpoint. For Stages 2 and 3, we set the chunk size to 3. During self-rollout distillation, the first generated chunk is retained as attention sink. We set the sparsity ratio ρ in FPSA to 75%. For the two-step generation, the sampling timesteps are set to $[1000, 250]$. We additionally construct CharEdit-Bench for comprehensive evaluation, comprising a short-video subset (CharEdit-Bench-S) and a long-video subset (CharEdit-Bench-L). Complete implementation details and benchmark specifics are provided in Appendix C.1 and B.2.

4.2 Comparison with Other Methods

We compare our method against state-of-the-art instruction-based video editing approaches, including the bidirectional baselines LucyEdit (Team, 2025), UniVideo (Wei et al., 2025), and

Table 1: Quantitative comparisons on CharEdit-Bench-S. Numbers in **red** and **blue** indicate the best and the second-best results, respectively. APD is multiplied by 10. TA, EQ, BC and SR denote text alignment, edit quality, background consistency, and success rate, respectively. **LiveEdit and LucyEdit do not support global style editing. We therefore omit their character-consistency metrics, as these scores are not comparable when the requested edit is not successfully performed.*

Method	#Params	Character Consistency			VLM Evaluation				Video Quality	Efficiency	
		ID-SIM ↑	AED ↓	APD ↓	TA ↑	EQ ↑	BC ↑	SR ↑	Pick Score ↑	FPS ↑	Latency ↓
LiveEdit*	1.3B	–	–	–	1.542	1.949	1.640	0.393	19.26	17.25	**0.696**
SANA-Stream.	2B	0.379	0.629	0.381	2.200	1.777	1.582	0.428	19.46	**31.20**	0.772
Stream.V2	14B	0.060	0.885	0.485	1.316	1.869	0.473	0.027	**19.92**	**20.96**	**0.191**
LucyEdit*	5B	–	–	–	1.604	1.797	**2.121**	0.326	18.92	3.083	26.27
UniVideo	13B	**0.592**	0.605	0.447	2.377	2.182	1.866	0.533	19.54	0.119	680.9
Ditto	17B	0.334	0.685	0.355	1.560	1.796	1.127	0.167	19.49	0.294	275.4
Qwen.+Wan.	17B	0.491	**0.541**	**0.132**	**2.792**	**2.544**	1.897	**0.713**	19.55	1.274	63.55
Ours	17B	**0.550**	**0.499**	**0.124**	**2.796**	**2.609**	**2.024**	**0.720**	**19.61**	14.47	0.829

Ditto (Bai et al., 2026), as well as the streaming baselines LiveEdit (Wang et al., 2026), SANA-Streaming (Zhao et al., 2026), and StreamDiffusionV2 (Feng et al., 2025a). We additionally include a cascaded image-editing-and-animation baseline that first edits the reference image using Qwen-Image-Edit (Wu et al., 2025a) and then animates the edited image with Wan-Animate (Cheng et al., 2025). We evaluate the performance in terms of character consistency, editing quality, video quality, and inference efficiency. ID-SIM (Deng et al., 2019) measures identity consistency, while AED and APD (Siarohin et al., 2019) assess the preservation of facial expressions and body poses, respectively. A VLM-based evaluator (Ju et al., 2025) assesses overall editing quality. Pick Score (Kirstain et al., 2023) is used to assess the overall quality of the edited videos. For efficiency, we report end-to-end generation throughput in frames per second (FPS) and average inter-chunk latency. Detailed metric definitions and evaluation protocols are provided in Appendix C.2 and C.3.

Qualitative & Quantitative Comparison. As shown in Fig. 6, EditaLive consistently performs both local appearance edits and global style transformations while faithfully preserving the source facial expressions, body poses, and scene structure. In comparison, existing streaming methods may either fail to fully apply the requested edits or introduce noticeable inconsistencies in facial expression. The bidirectional baselines exhibit similar limitations, despite their substantially higher inference costs. These qualitative observations are consistent with the quantitative results in Table 1, where EditaLive ranks first or second across all eight quality metrics. We omit the character-consistency metrics of LiveEdit and LucyEdit because they do not support global style editing, making these metrics incomparable when the requested transformation is not successfully applied. Additional comparisons on CharEdit-Bench-L are provided in Appendix D.

Efficiency. As reported in Table 1, EditaLive achieves an end-to-end generation throughput of 14.47 FPS with an average inter-chunk latency of 0.829 seconds on a single NVIDIA H100 GPU. Although the dedicated streaming baselines achieve higher raw efficiency, their edit success rates are at most 0.428, substantially below the 0.720 achieved by EditaLive. Compared with LucyEdit, the fastest bidirectional baseline, EditaLive delivers $4.7\times$ the throughput and reduces latency by a factor of 31.7. For the cascaded Qwen-Image-Edit and Wan-Animate baseline, even when excluding the one-time cost of reference-image editing, EditaLive remains $11.4\times$ as fast and reduces latency by a factor of 76.7. These results demonstrate that EditaLive provides a favorable quality–efficiency trade-off. Moreover, replacing the standard Wan-VAE decoder with Flash-VAED (Zhu et al., 2026) further increases the inference speed of EditaLive to 16.4 FPS.

4.3 Ablation Studies

To validate the effectiveness of our key components, we conduct comprehensive ablation studies on CharEdit-Bench-L, covering both causal streaming adaptation and aligned self-rollout distillation.

Causal Streaming Adaptation. We first examine how the reference latent should be used during causal adaptation. As shown in Fig. 7 (left), optimizing both the reference and target positions (*Ref.+Tgt.*) introduces conflicting supervision. Under the same instruction, the reference latent is optimized to reconstruct its unchanged input appearance, whereas the target latents are required to perform the instructed appearance transformation. This conflict weakens the reference conditioning and causes unintended changes to the generated character. We therefore keep the reference latent



Figure 7: Ablation study on causal streaming adaptation and aligned self-rollout distillation.

Table 2: Ablation study on aligned self-rollout distillation.

Method	Character Consistency			VLM Evaluation				Video Quality	Temporal Consistency	
	ID-SIM ↑	AED ↓	APD ↓	TA ↑	EQ ↑	BC ↑	SR ↑	Pick Score ↑	CLIP ↑	DINO ↑
w/o Align Forcing	0.252	0.711	0.338	2.703	2.567	1.306	0.367	**20.03**	**98.62**	**99.07**
w/o Fixed RoPE	0.381	0.618	0.120	2.728	2.639	1.872	0.617	**19.91**	98.50	98.95
w/o FPSA	0.455	0.609	0.116	**2.737**	2.600	2.105	0.800	19.85	98.46	98.88
w/o FP	**0.466**	**0.604**	**0.114**	2.725	**2.622**	**2.139**	**0.833**	19.84	98.42	98.89
Ours	**0.492**	**0.576**	**0.109**	**2.761**	**2.678**	**2.183**	**0.817**	19.78	**98.51**	**98.96**

clean and use it solely as conditioning context, while computing the Flow Matching loss only over the noisy target positions (*Tgt. only*). This target-only objective separates stable appearance conditioning from instructed target generation, leading to more faithful character appearance.

Aligned Self-Rollout Distillation. We evaluate the key components of aligned self-rollout distillation in Table 2 and Fig. 7 (right). Replacing Align Forcing with Self Forcing (*w/o Align Forcing*) introduces a mismatch in KV-cache construction between rollout training and streaming inference, causing the largest performance degradation: ID-SIM decreases from 0.492 to 0.252, and SR decreases from 0.817 to 0.367. Removing Fixed RoPE exposes the model to unseen positional offsets during long-video inference and weakens its reference conditioning, resulting in progressive appearance drift and an ID-SIM of only 0.381. FPSA contributes through both sparse context selection and explicit first-frame preservation. Without FPSA, full attention overemphasizes recent local context while underutilizing stable appearance cues, resulting in an ID-SIM of 0.455. Introducing sparse attention without first-frame preservation (*w/o FP*) improves ID-SIM to 0.466, demonstrating the benefit of filtering redundant historical information. Nevertheless, visible appearance changes still emerge over time, such as the highlighted hair variation at frame 900. Explicitly preserving the first-frame features further increases ID-SIM to 0.492 and reduces AED and APD to 0.576 and 0.109, respectively. Overall, the complete model achieves the best ID-SIM, AED, APD, TA, EQ, and BC, while maintaining comparable video quality and temporal consistency.

5 Conclusion

We introduce EditaLive, a unified framework that brings instruction-guided character appearance editing to live video streams. By decoupling appearance from body motion and facial dynamics, our reconstruction-based training learns diverse appearance transformations from edited reference images and authentic, motion-aligned video targets; we construct CharEdit-50K to support this formulation. We further convert the bidirectional editing model into a chunk-wise causal generator and distill it into a two-step sampler. Align Forcing and Fixed RoPE align rollout training with streaming inference, while FPSA maintains stable appearance cues over long sequences with bounded historical context. Comprehensive experiments demonstrate the advantages of EditaLive in terms of editing quality, motion consistency, long-term stability, and inference efficiency.

AI USE STATEMENT

Generative AI tools were used in this work for two purposes. First, as part of the research pipeline described in the paper, GPT-5.5 was used to generate candidate editing instructions, filter synthesized samples, and perform VLM-based evaluation, while Qwen-Image-Edit and Nano Banana 2 were used to synthesize edited reference images for constructing CharEdit-50K. Second, generative AI assistants were used to support manuscript drafting and language refinement, as well as code implementation and debugging. The core research problem, methodology, experimental design, and interpretation of the results were determined by the authors. All AI-assisted text was manually reviewed and revised by the authors. AI-assisted code was inspected, executed, and tested for correctness against the intended algorithms, and the resulting experimental outputs were verified by the authors. AI-generated instructions and edited images were filtered using the automated quality criteria described in the CharEdit-50K construction pipeline and further inspected through manual spot checks by the authors. The authors take responsibility for the final content of this work, including all text, code, claims, results, and artifacts produced with the assistance of generative AI.

ETHICS STATEMENT

Our work focuses on enabling real-time and long-term character video editing, which raises concerns regarding privacy, consent, copyright, and potential misuse of human-centered visual data.

Data provenance and permitted use. CharEdit-50K is constructed from samples drawn from three publicly released research datasets: SpeakerVid-5M (Zhang et al., 2025b), the Seamless Interaction Dataset (Agrawal et al., 2025), and TikTok-Dance5K (Pang et al., 2025). We accessed and processed these sources solely for non-commercial research and in accordance with the licenses, data-use terms, and access conditions documented by their original providers. We do not host or redistribute the original source videos. Any additional online videos used for qualitative evaluation were included only where permitted by the applicable terms or authorization; they were not used for training and are neither redistributed nor included in any released artifact.

Privacy and responsible use. We do not collect identity labels, biometric identifiers, or sensitive personal attributes, and our method is not designed to identify individuals or recover private identity information. We do not claim ownership of the source media or the identities depicted therein, and our use of these data does not grant downstream users any rights beyond the original data-use terms.

Potential misuse. Realistic character editing may be susceptible to impersonation, deceptive manipulation, or non-consensual use. EditaLive is intended solely for legitimate research, creative production, and interactive applications. Users should obtain consent from depicted individuals, respect applicable rights and licenses, and adopt safeguards such as access control, content disclosure, and watermarking. The authors do not endorse any use that violates consent, privacy, copyright, applicable law, or the original providers' data-use conditions and access restrictions.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the model architecture, training objectives, and inference procedure in the main paper. The appendix further documents the training algorithm, CharEdit-50K construction process, implementation details, baseline configurations, and evaluation protocols to support independent reproduction of our method and experimental results.

REFERENCES

Vasu Agrawal, Akinniyi Akinyemi, Kathryn Alvero, Morteza Behrooz, Julia Buffalini, Fabio Maria Carlucci, Joy Chen, Junming Chen, Zhang Chen, Shiyang Cheng, et al. Seamless interaction: Dyadic audiovisual motion modeling and large-scale dataset. *arXiv preprint arXiv:2506.22554*, 2025.

Qingyan Bai, Qiuyu Wang, Hao Ouyang, Yue Yu, Hanlin Wang, Wen Wang, Ka Leong Cheng, Shuailei Ma, Yanhong Zeng, Zichen Liu, et al. Scaling instruction-based video editing with a high-quality synthetic dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 37971–37981, 2026.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18392–18402, 2023.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 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.

Feng Chen, Bohan Zhuang, and Qi Wu. Streaming video diffusion: Online video editing with diffusion models. In *2025 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1–9. IEEE, 2025.

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.

Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4690–4699, 2019.

Tianrui Feng, Zhi Li, Shuo Yang, Haocheng Xi, Muyang Li, Xiuyu Li, Lvmin Zhang, Keting Yang, Kelly Peng, Song Han, et al. Streamdiffusionv2: A streaming system for dynamic and interactive video generation. *arXiv preprint arXiv:2511.07399*, 2025a.

Weilun Feng, Haotong Qin, Chuanguang Yang, Zhulin An, Libo Huang, Boyu Diao, Fei Wang, Renshuai Tao, Yongjun Xu, and Michele Magno. Mpq-dm: Mixed precision quantization for extremely low bit diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 16595–16603, 2025b.

Yuchao Gu, Guian Fang, Yuxin Jiang, Weijia Mao, Song Han, Han Cai, and Mike Zheng Shou. Anyflow: Any-step video diffusion model with on-policy flow map distillation. *arXiv preprint arXiv:2605.13724*, 2026.

Haoyang He, Jie Wang, Jiangning Zhang, Zhucun Xue, Xingyuan Bu, Qiangpeng Yang, Shilei Wen, and Lei Xie. Openve-3m: A large-scale high-quality dataset for instruction-guided video editing. *arXiv preprint arXiv:2512.07826*, 2025.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

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.

Xijie Huang, Chengming Xu, Donghao Luo, Xiaobin Hu, Peng Tang, Xu Peng, Jiangning Zhang, Chengjie Wang, and Yanwei Fu. Ffp-300k: Scaling first-frame propagation for generalizable video editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23172–23181, 2026a.

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, 2026b.

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.

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

Xuan Ju, Tianyu Wang, Yuqian Zhou, He Zhang, Qing Liu, Nanxuan Zhao, Zhifei Zhang, Yijun Li, Yuanhao Cai, Shaoteng Liu, et al. Editverse: Unifying image and video editing and generation with in-context learning. *arXiv preprint arXiv:2509.20360*, 2025.

Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *Advances in neural information processing systems*, 36:36652–36663, 2023.

Akio Kodaira, Chenfeng Xu, Toshiki Hazama, Takanori Yoshimoto, Kohei Ohno, Shogo Mitsuhori, Soichi Sugano, Hanying Cho, Zhijian Liu, Masayoshi Tomizuka, et al. Streamdiffusion: A pipeline-level solution for real-time interactive generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12371–12380, 2025.

Zhiyuan Li, Chi-Man Pun, Chen Fang, Jue Wang, and Xiaodong Cun. Personalive! expressive portrait image animation for live streaming. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18118–18128, 2026.

Feng Liang, Akio Kodaira, Chenfeng Xu, Masayoshi Tomizuka, Kurt Keutzer, and Diana Marculescu. Looking backward: Streaming video-to-video translation with feature banks. In *International Conference on Learning Representations*, volume 2025, pp. 46425–46445, 2025.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Yue Ma, Jiangming Wang, Yucheng Wang, Xilai Wang, Zhiyuan Li, Xinyu Wang, Hongyu Liu, Ruofan Liang, Songchun Zhang, Yuxuan Xue, et al. Livelight: Real-time streaming video relighting with interactive control. *arXiv preprint arXiv:2608.01771*, 2026.

Mang Ning, Mingxiao Li, Jianlin Su, Albert Ali Salah, and Itir Onal Ertugrul. Elucidating the exposure bias in diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024.

Yatian Pang, Bin Zhu, Bin Lin, Mingzhe Zheng, Francis EH Tay, Ser-Nam Lim, Harry Yang, and Li Yuan. Dreamdance: Animating human images by enriching 3d geometry cues from 2d poses. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14039–14050, 2025.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Chenyang Qi, Xiaodong Cun, Yong Zhang, Chenyang Lei, Xintao Wang, Ying Shan, and Qifeng Chen. Fatezero: Fusing attentions for zero-shot text-based video editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15932–15942, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

George Retsinas, Panagiotis P. Filntisis, Radek Daněček, Victoria F. Abrevaya, Anastasios Roussos, Timo Bolkarr, and Petros Maragos. 3d facial expressions through analysis-by-neural-synthesis. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2490–2501, 2024.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Florian Schmidt. Generalization in generation: A closer look at exposure bias. *EMNLP-IJCNLP 2019*, pp. 157, 2019.

Shitong Shao, Zikai Zhou, Haopeng Li, Yingwei Song, Wenliang Zhong, Lichen Bai, and Zeke Xie. LIVEditor-14b: Lightning unified video editing via in-context sparse attention. In *Forty-third International Conference on Machine Learning*, 2026.

Shelly Sheynin, Adam Polyak, Uriel Singer, Yuval Kirstain, Amit Zohar, Oron Ashual, Devi Parikh, and Yaniv Taigman. Emu edit: Precise image editing via recognition and generation tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8871–8879, 2024.

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.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

DecartAI Team. Lucy edit: Open-weight text-guided video editing, 2025.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

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.

Xinyu Wang, Chongbo Zhao, Fangneng Zhan, and Yue Ma. Liveedit: Towards real-time diffusion-based streaming video editing. *arXiv preprint arXiv:2606.26740*, 2026.

Cong Wei, Quande Liu, Zixuan Ye, Qiulin Wang, Xintao Wang, Pengfei Wan, Kun Gai, and Wenhu Chen. Univideo: Unified understanding, generation, and editing for videos. *arXiv preprint arXiv:2510.08377*, 2025.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025a.

Yuhui Wu, Liyi Chen, Ruibin Li, Shihao Wang, Chenxi Xie, and Lei Zhang. Insvie-1m: Effective instruction-based video editing with elaborate dataset construction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16692–16701, 2025b.

Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, et al. Sana: Efficient high-resolution text-to-image synthesis with linear diffusion transformers. In *The Thirteenth International Conference on Learning Representations*, 2025.

Zhening Xing, Gereon Fox, Yanhong Zeng, Xingang Pan, Mohamed Elgharib, Christian Theobalt, and Kai Chen. Live2diff: Live stream translation via uni-directional attention in video diffusion models. *arXiv preprint arXiv:2407.08701*, 2024.

Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. Vitpose: Simple vision transformer baselines for human pose estimation. *Advances in neural information processing systems*, 35:38571–38584, 2022.

Wenhao Yan, Fengjia Guo, Zhuoyi Yang, and Jie Tang. Scail-2: Unifying controlled character animation with end-to-end in-context conditioning. *arXiv preprint arXiv:2606.10804*, 2026a.

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, 2026b.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and Bill Freeman. Improved distribution matching distillation for fast image synthesis. *Advances in neural information processing systems*, 37:47455–47487, 2024a.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6613–6623, 2024b.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22963–22974, 2025.

Shenghai Yuan, Yuanyang Yin, Zongjian Li, Xinwei Huang, Xiao Yang, and Li Yuan. Helios: Real real-time long video generation model. *arXiv preprint arXiv:2603.04379*, 2026.

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*, 2025a.

Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. *Advances in Neural Information Processing Systems*, 36:31428–31449, 2023.

Peiyuan Zhang, Yongqi Chen, Haofeng Huang, Will Lin, Zhengzhong Liu, Ion Stoica, Eric Xing, and Hao Zhang. Faster video diffusion with trainable sparse attention. *Advances in Neural Information Processing Systems*, 38:152509–152534, 2026a.

Youliang Zhang, Zhaoyang Li, Duomin Wang, Jiahe Zhang, Deyu Zhou, Zixin Yin, Xili Dai, Gang Yu, and Xiu Li. Speakervid-5m: A large-scale high-quality dataset for audio-visual dyadic interactive human generation. *arXiv preprint arXiv:2507.09862*, 2025b.

Zechuan Zhang, Ji Xie, Yu Lu, Zongxin Yang, and Yi Yang. Enabling instructional image editing with in-context generation in large scale diffusion transformer. *Advances in Neural Information Processing Systems*, 38:139195–139227, 2026b.

Xiaochen Zhao, Hongyi Xu, Guoxian Song, You Xie, Chenxu Zhang, Xiu Li, Linjie Luo, Jinli Suo, and Yebin Liu. X-nemo: Expressive neural motion reenactment via disentangled latent attention. In *The Thirteenth International Conference on Learning Representations*, 2025.

Yuyang Zhao, Yicheng Pan, Qiyuan He, Jincheng Yu, Junsong Chen, Tian Ye, Haozhe Liu, Enze Xie, and Song Han. Sana-streaming: Real-time streaming video editing with hybrid diffusion transformer. *arXiv preprint arXiv:2605.30409*, 2026.

Lunjie Zhu, Yushi Huang, Xingtong Ge, Yufei Xue, Zhening Liu, Yumeng Zhang, Zehong Lin, and Jun Zhang. Flash-vaed: Plug-and-play vae decoders for efficient video generation. *arXiv preprint arXiv:2602.19161*, 2026.

A PRELIMINARIES

Video Diffusion Models. Video diffusion models learn a conditional distribution over video sequences through iterative denoising. To reduce the computational cost of processing high-dimensional videos, an input video x is typically encoded into a compact latent representation Z_1 using a spatio-temporal VAE, and generation is performed in the resulting latent space. Under the flow-matching formulation (Lipman et al., 2022), a noisy latent Z_t is constructed by interpolating between a Gaussian sample $Z_0 \sim \mathcal{N}(0, I)$ and the clean video latent:

$$Z_t = tZ_1 + (1-t)Z_0, \qquad t \in [0,1]. \tag{8}$$

The corresponding target velocity $V_t = \frac{\mathrm{d}Z_t}{\mathrm{d}t} = Z_1 - Z_0$ is constant along this path. A conditional velocity model v_θ is trained to estimate V_t from the interpolated latent Z_t, timestep t, and conditioning signals c by minimizing

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}_{Z_1, Z_0, t}\left[\left\| v_\theta(Z_t, t \mid c) - (Z_1 - Z_0)\right\|_2^2\right], \tag{9}$$

where c may include text, image, or other conditions. At inference time, a video latent is generated by numerically integrating the learned velocity field from the Gaussian prior at $t = 0$ toward the data distribution at $t = 1$, after which the VAE decoder maps the resulting latent back to video space.

Distribution Matching Distillation. Distribution Matching Distillation (DMD) (Yin et al., 2024b;a) distills a pretrained diffusion model into a student generator G_θ with substantially fewer denoising steps. Let $p_{\theta,t}$ denote the distribution obtained by perturbing student-generated samples to timestep t, and let $p_{\mathrm{real},t}$ denote the corresponding distribution induced by the real data or pretrained teacher model. DMD minimizes the reverse Kullback–Leibler divergence between these distributions:

$$\mathcal{L}_{\mathrm{DMD}} = \mathbb{E}_t\left[D_{\mathrm{KL}}\left(p_{\theta,t} \,\|\, p_{\mathrm{real},t}\right)\right]. \tag{10}$$

Given a clean prediction $\hat{Z}_1$ and its perturbed version $\hat{Z}_t$, the gradient of this objective can be estimated from the difference between the score functions of the student and target distributions:

$$\nabla_\theta \mathcal{L}_{\mathrm{DMD}} = -\mathbb{E}_{Z_0,t}\left[\left(s_{\mathrm{real}}(\hat{Z}_t, t, c) - s_{\mathrm{fake},\phi}(\hat{Z}_t, t, c)\right)^\top \frac{\partial G_\theta(Z_0, c)}{\partial \theta}\right], \tag{11}$$

where s_{real} estimates the score of the target distribution and $s_{\mathrm{fake},\phi}$ estimates the score of the current student distribution. The real-score model is kept fixed, whereas the fake-score model is trained on samples produced by the evolving student generator. Training alternates between updating $s_{\mathrm{fake},\phi}$ to track the student distribution and updating G_θ using the score difference above.

B ADDITIONAL DATASET INFORMATION

B.1 CHAREDIT-50K

Video Collection and Reference Selection. To construct CharEdit-50K, we assemble a diverse video pool from three publicly available research datasets: SpeakerVid-5M (Zhang et al., 2025b), the Seamless Interaction Dataset (Agrawal et al., 2025), and TikTok-Dance5K (Pang et al., 2025). From each video, we select a single high-quality frame as the appearance reference by examining facial sharpness and completeness, as well as the visibility and sharpness of the hands when present. Frames with severe occlusion, motion blur, truncation, or other visual degradations are excluded from selection, while each selected frame remains paired with its authentic source video.

Instruction Generation. As shown in Fig. 8, conditioned on each reference image, GPT-5.5 (Singh et al., 2025) generates candidate editing instructions covering four categories. For *add–remove*, the forward instruction adds an object or accessory to an unoccupied and physically plausible region, while the reverse instruction removes the added object. For *remove–add*, the forward instruction removes an existing object and reconstructs the occluded content, while the reverse instruction adds the object with its visible attributes and placement. For *change–restore*, the forward instruction changes only the color of a visible target, such as clothing or an accessory, while preserving its shape, material, and texture, and the reverse instruction specifies its original color. The final category is *stylization*, which converts the reference image into a specified target visual style without

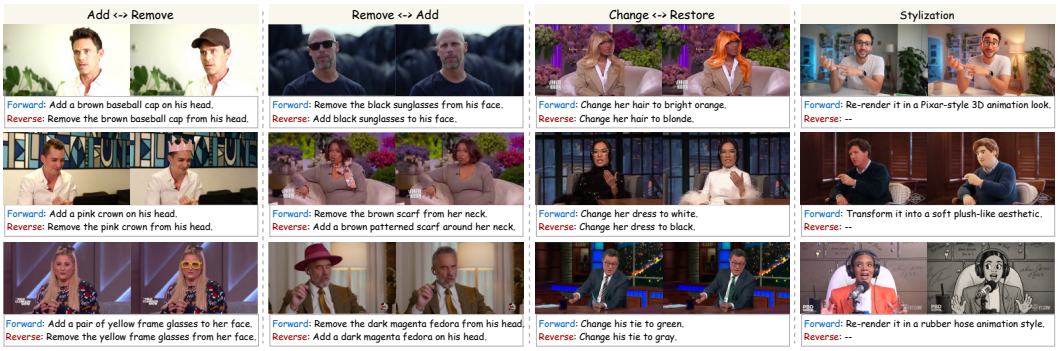


Figure 8: Examples from CharEdit-50K.

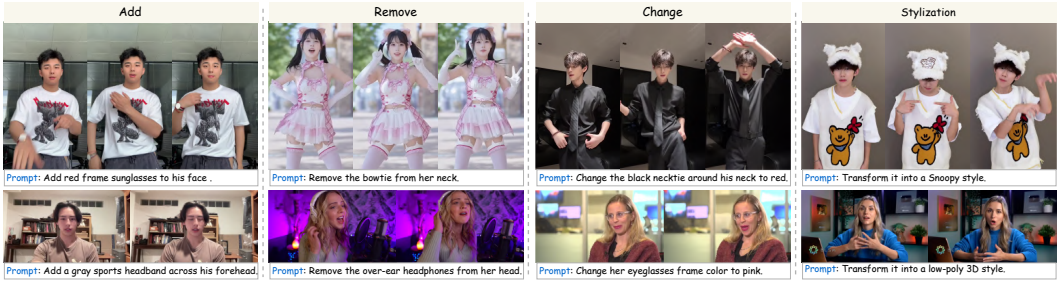


Figure 9: Examples from CharEdit-Bench.

requiring a corresponding sample in the reverse direction. After instruction generation, we balance the distributions of target categories and colors across the complete set of generated instruction pairs.

Image Synthesis and Filtering. Qwen-Image-Edit (Wu et al., 2025a) and Nano Banana 2 (Team et al., 2023) synthesize candidate images according to the generated forward instructions. Given the original reference, an edited candidate, and its associated instruction(s), GPT-5.5 evaluates edit accuracy, target clarity, and character consistency. For bidirectional edits, it additionally evaluates whether the reverse instruction unambiguously specifies the inverse transformation.

B.2 CharEdit-Bench

To comprehensively evaluate character video editing in both conventional and streaming settings, we construct CharEdit-Bench with two subsets: CharEdit-Bench-S and CharEdit-Bench-L. The source videos for CharEdit-Bench are collected from YouTube, TikTok, and Bilibili. CharEdit-Bench-S comprises 150 five-second videos, each of which contains 81 frames at a resolution of 480×832 and is randomly paired with one editing prompt, yielding 150 evaluation cases. CharEdit-Bench-L comprises 30 videos at the same resolution, each longer than one minute at 16 FPS and paired with two editing prompts, resulting in 60 evaluation cases. For every evaluation case, the first frame of the source video is used as the reference image. Examples from CharEdit-Bench are shown in Fig. 9.

C Additional Implementation Details

C.1 Training Details

Our training pipeline proceeds through three stages. In Stage 1, we perform reconstruction-based appearance–motion decoupled training, enabling the model to learn instruction-guided appearance transformations while following independently provided motion conditions. We employ ViTPose (Xu et al., 2022) to extract the skeleton sequence and localize the facial region, subsequently constructing the corresponding facial video. Following this initialization, Stage 2 adapts the bidirectional model to chunk-wise causal generation for streaming inference. We optimize only the LoRA adapters in the self-attention and FFN modules while keeping the cross-attention modules frozen (Gu et al., 2026). We implement the chunk-wise causal attention mask using PyTorch `flex_attention`, allowing bidirectional attention within each chunk while restricting inter-chunk attention to the preceding clean context. After causal adaptation, Stage 3 performs aligned

Algorithm 1 Align Forcing Training with Fixed RoPE

Require: Denoising timesteps $\{t_0, t_1, \ldots, t_T\}$, where $t_0 = 0$
Require: Number of video chunks N
Require: Reference frame x_R
Require: Per-chunk conditions $\{C_i\}_{i=1}^{N}$, including text and motion signals
Require: Cache lengths $(L_{\text{ref}}, L_{\text{sink}}, L_{\text{local}}) = (1, 3, 3)$
Require: Generator G_θ, which also returns pre-RoPE KV features

loop
  Initialize model output $\mathbf{X}_\theta \leftarrow []$
  Initialize $\mathbf{KV}_{\text{sink}} \leftarrow []$, $\mathbf{KV}_{\text{local}} \leftarrow []$
  Sample $s \sim \text{Uniform}\{1, 2, \ldots, T\}$
  Set reference KV cache $\mathbf{KV}_{\text{ref}} \leftarrow sg(G_\theta(x_R; t_0))$
  for $i = 1, \ldots, N$ **do**
    Initialize $x^i_{t_T} \sim \mathcal{N}(0, I)$
    $\mathbf{KV}^i \leftarrow [\mathbf{KV}_{\text{ref}}, \mathbf{KV}_{\text{sink}}, Tail_{L_{\text{local}}}(\mathbf{KV}_{\text{local}})]$ ▷ fixed-length active cache
    Assign Fixed RoPE indices (0:9)
    for $j = T, \ldots, 1$ **do** ▷ complete the full rollout
      if $j = s$ **then**
        Enable gradient computation
        Set $\mathbf{kv}^i, \hat{x}^i_0 \leftarrow G_\theta(x^i_{t_j}; t_j, \mathbf{KV}^i, C_i)$
        $\mathbf{X}_\theta$. `append`$(\hat{x}^i_0)$
        Detach $\mathbf{kv}^i$ from gradient graph
      else
        Disable gradient computation
        Set $\mathbf{kv}^i, \hat{x}^i_0 \leftarrow G_\theta(x^i_{t_j}; t_j, \mathbf{KV}^i, C_i)$
      end if
      if $j > 1$ **then**
        Sample $\epsilon \sim \mathcal{N}(0, I)$
        Set $x^i_{t_{j-1}} \leftarrow \Psi(\hat{x}^i_0, \epsilon, t_{j-1})$
      end if
    end for
    if $i = 1$ **then**
      Disable gradient computation
      Set $\mathbf{kv}^i, _ \leftarrow G_\theta(\hat{x}^i_0; t_0, \mathbf{KV}^i, C_i)$ ▷ one-time cache forward on the first chunk
      $\mathbf{KV}_{\text{sink}}$. `append`$(\mathbf{kv}^i)$
    else
      $\mathbf{KV}_{\text{local}}$. `append`$(\mathbf{kv}^i)$
    end if
  end for
  Update θ via DMD loss
end loop

self-rollout distillation to compress the causal model into a two-step sampler and achieve long-term streaming generation, as shown in Algorithm 1. We use the same trainable modules as in Stage 2. During DMD training, we optimize the student and fake-score branch while keeping the real-score branch frozen, performing five fake-score updates for every student update. To handle the high memory demand of Align Forcing training, we adopt Fully Sharded Data Parallel (FSDP) to reduce memory consumption. For FPSA, the query and key tokens are partitioned into spatio-temporal blocks of shape $(1, 8, 8)$. For style editing, synthesizing a complete stylized video for every training sample would be prohibitively expensive. We instead stylize a single image and repeat it along the temporal dimension to construct a static target video. Since appearance and motion are learned from decoupled conditions, this static target provides sufficient supervision for the target style, which can subsequently be combined with motion signals during inference.

Table 3: Quantitative comparisons on CharEdit-Bench-L. Numbers in **red** and **blue** indicate the best and the second-best results, respectively. APD is multiplied by 10. **LiveEdit does not support global style editing. We therefore omit its character-consistency metrics, as these scores are not comparable when the requested edit is not successfully performed.*

Method	Character Consistency			VLM Evaluation				Video Quality	Temporal Consistency	
	ID-SIM ↑	AED ↓	APD ↓	TA ↑	EQ ↑	BC ↑	SR ↑	Pick Score ↑	CLIP ↑	DINO ↑
LiveEdit*	–	–	–	1.583	1.700	1.044	0.017	18.91	97.36	98.38
SANA-Stream.	0.233	0.725	0.402	2.272	1.938	1.121	0.233	19.62	97.28	98.25
Stream.V2	0.049	0.876	0.329	1.431	2.064	0.638	0.100	**19.70**	**98.42**	**99.17**
Qwen.+Wan.	**0.398**	**0.659**	**0.126**	**2.733**	**2.304**	**1.776**	**0.750**	19.69	98.30	98.65
Ours	**0.492**	**0.576**	**0.109**	**2.761**	**2.678**	**2.183**	**0.817**	**19.78**	**98.51**	**98.96**

C.2 Baseline Configurations

All baselines are evaluated using their official implementations and released checkpoints. The #Params column in Table 1 reports only the number of parameters in the DiT backbone used by each method. For Ditto (Bai et al., 2026), we use the officially released `ditto_local.safetensors` and `ditto_global_style.safetensors` LoRA checkpoints for local and global style editing, respectively. Since SANA-Streaming (Zhao et al., 2026) was trained at a resolution of 704×1280, we resize the input videos to this resolution for inference and resize the generated videos back to 480×832 before evaluation. For StreamDiffusionV2 (Feng et al., 2025a), we follow its official inference launcher, which uses one-step denoising by default.

C.3 Evaluation Details

We evaluate identity and motion consistency between the source and edited videos using three paired metrics. ID-SIM is computed as the average cosine similarity between ArcFace (Deng et al., 2019) embeddings extracted from temporally corresponding source and edited frames. AED (Siarohin et al., 2019) is calculated as the average ℓ_1 distance between extracted expression parameters of the source and edited frames using SMIRK (Retsinas et al., 2024). APD (Siarohin et al., 2019) is computed as the average ℓ_1 distance between corresponding body keypoints extracted by ViTPose (Xu et al., 2022). Higher ID-SIM and lower AED and APD indicate better preservation of identity, facial expressions, and body poses, respectively. For VLM evaluation, we uniformly sample three temporally aligned frame pairs from each source and edited video and provide GPT-5.5 (Singh et al., 2025) with the source frame, edited frame, and editing instruction. GPT-5.5 assigns a score from 0 to 3 for each of three criteria: Text Alignment, Edit Quality, and Background Consistency. It additionally returns a binary overall-success judgment based on these criteria. Based on these binary judgments, we define edit Success Rate as the proportion of sampled source–edited frame pairs judged successful. We use Pick Score (Kirstain et al., 2023) to assess overall video quality. For inference efficiency, we report FPS and inter-chunk latency on a single NVIDIA H100 GPU using 81-frame clips at a resolution of 384×672 (Yuan et al., 2026; Huang et al., 2025). Runtime is measured over the generation pipeline, from model-ready conditions to decoded RGB frames. For each model, we enable its officially supported acceleration techniques (e.g., FlashAttention, `torch.compile`, and warm-up) to maximize throughput. Specifically, EditaLive is compiled with `torch.compile`, and its reported runtime includes motion-condition encoding, two-step DiT denoising, and VAE decoding. For long-video evaluation, we additionally assess temporal consistency using frame-wise CLIP (Radford et al., 2021) and DINO (Caron et al., 2021) feature similarities.

D More Experimental Results

Evaluation on Long Videos. We further evaluate long-term character video editing in the streaming setting on CharEdit-Bench-L. As shown in Table 3, EditaLive achieves the strongest overall performance among the compared methods, demonstrating that its editing quality and character consistency are maintained over long sequences. The qualitative comparisons in Fig. 10 further show that EditaLive maintains a consistent edited appearance over long video sequences while faithfully following the source motion. These results demonstrate the long-term stability of EditaLive.

Cross-Character Editing. Owing to our appearance–motion decoupled formulation, EditaLive naturally supports cross-character editing by combining the appearance of a reference character

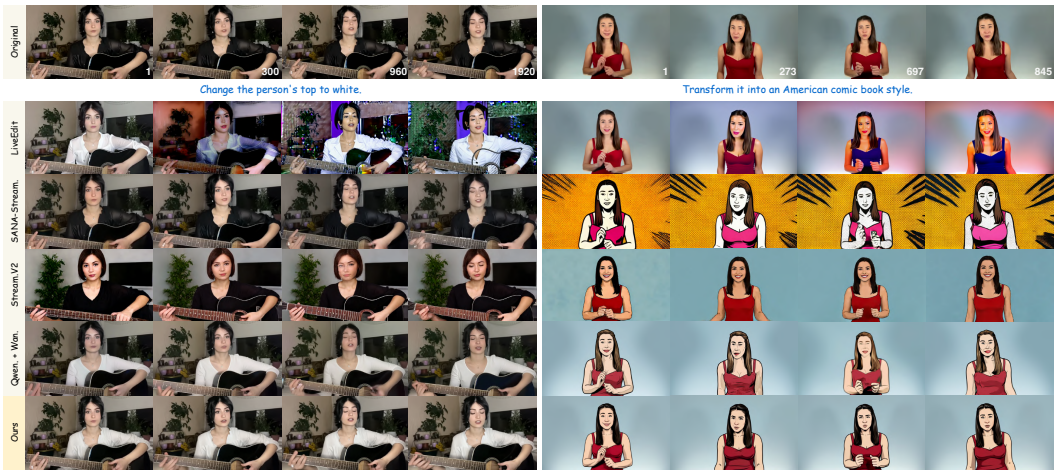


Figure 10: Qualitative comparisons on CharEdit-Bench-L. EditaLive consistently maintains the requested edit and character appearance while following the source motion.

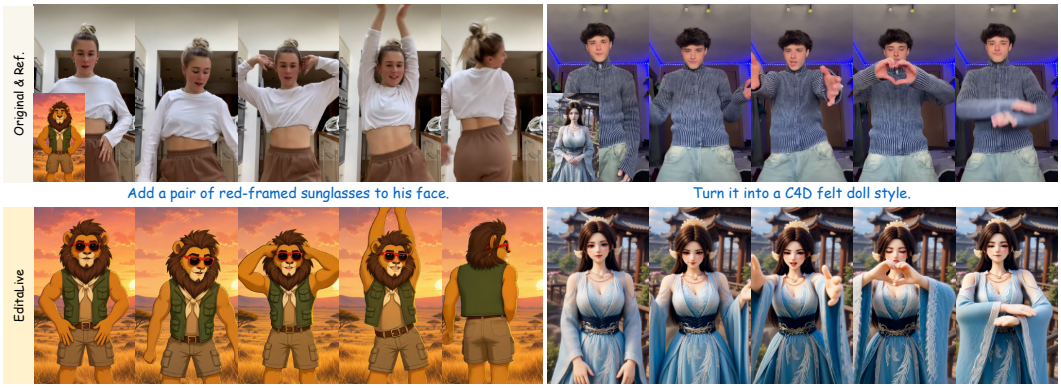


Figure 11: Cross-character editing results.

with motion signals extracted from a different driving video, as shown in Fig. 11. Across diverse reference–motion combinations, EditaLive consistently preserves the reference appearance and requested edits while following the driving motion, demonstrating its flexibility and temporal stability in cross-character editing scenarios involving diverse appearances and motion patterns.

User Study. To further evaluate the qualitative performance of our method against existing baselines, we conducted a user study with 40 participants. The study comprised 60 video sets: 40 short-video sets and 20 long-video sets, with 10 and 5 sets per editing category, respectively. For each set, participants selected the best video according to four criteria: **I**nstruction **A**lignment, **C**haracter **C**onsistency, **T**emporal **C**onsistency, and overall **V**ideo **Q**uality. As shown in Fig. 12, EditaLive receives the highest preference share across all four criteria for both short and long videos, with a particularly pronounced advantage on long videos.

Additional Qualitative Results. Figs. 14, 15, 17, and 19 present additional qualitative comparisons across diverse characters and editing instructions, further demonstrating the robustness and generalization of EditaLive. Figs. 16 and 18 present comparisons over extended video sequences, highlighting the long-term consistency and stability of EditaLive. Compared with existing methods, EditaLive exhibits substantially less appearance drift and consistently maintains the requested edit and character appearance while following the source motion.

E Limitations & Future Work

While EditaLive achieves real-time and temporally coherent long-term character video editing, there remain two directions for further improvement, as shown in Fig. 13. First, the current framework

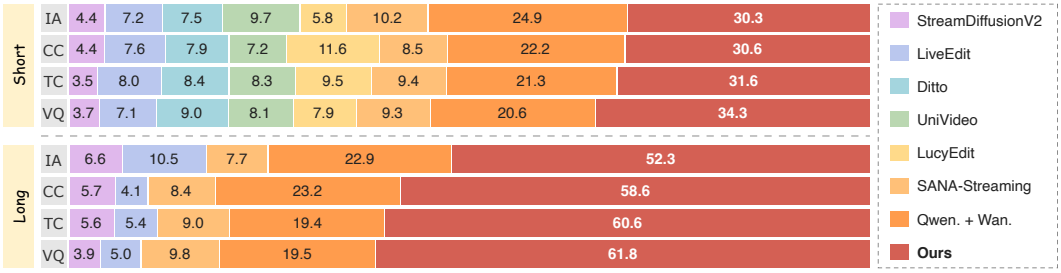


Figure 12: User study results on short and long videos.

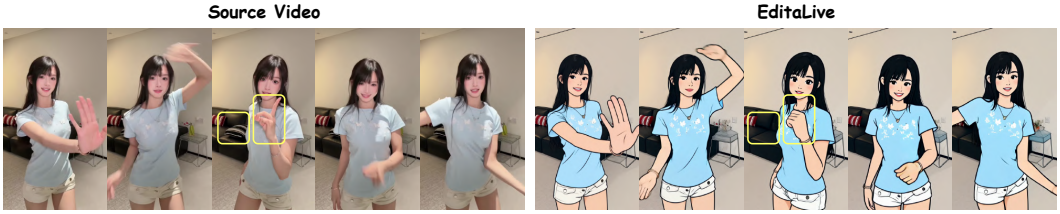


Figure 13: Limitations of EditaLive. The editing prompt is "*Transform it into a Snoopy style*".

relies on skeleton sequences for body-motion control. Although they effectively capture overall body poses, they do not explicitly represent fine-grained finger articulation, and subtle or complex hand gestures may therefore be reproduced less accurately. Incorporating more expressive hand-motion conditions, such as dense hand keypoints or hand-specific representations, could further improve fine-grained motion preservation. Second, character appearance is represented by a single reference image. Although this compact condition is effective in most cases, it provides limited appearance cues for regions occluded in the reference image. When these regions become visible during subsequent motion, their inferred appearance may differ from that in the source video. Future work could leverage multiple reference images to provide more complete appearance cues.



Figure 14: More comparisons between baselines and our EditaLive (1/4).



Figure 15: More comparisons between baselines and our EditaLive (2/4).

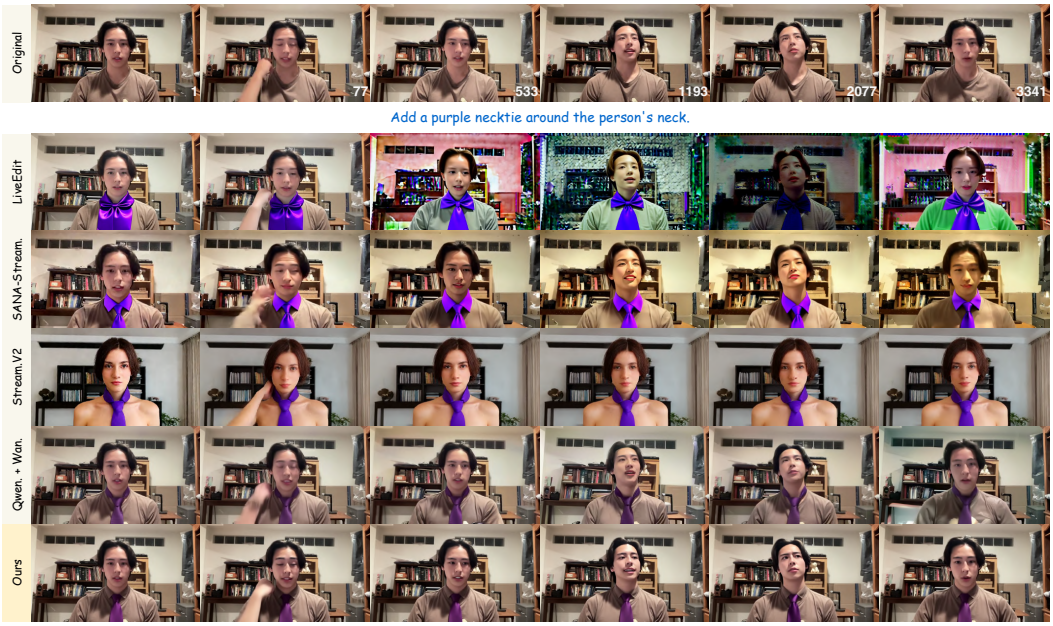


Figure 16: Long-video comparison results (1/2).

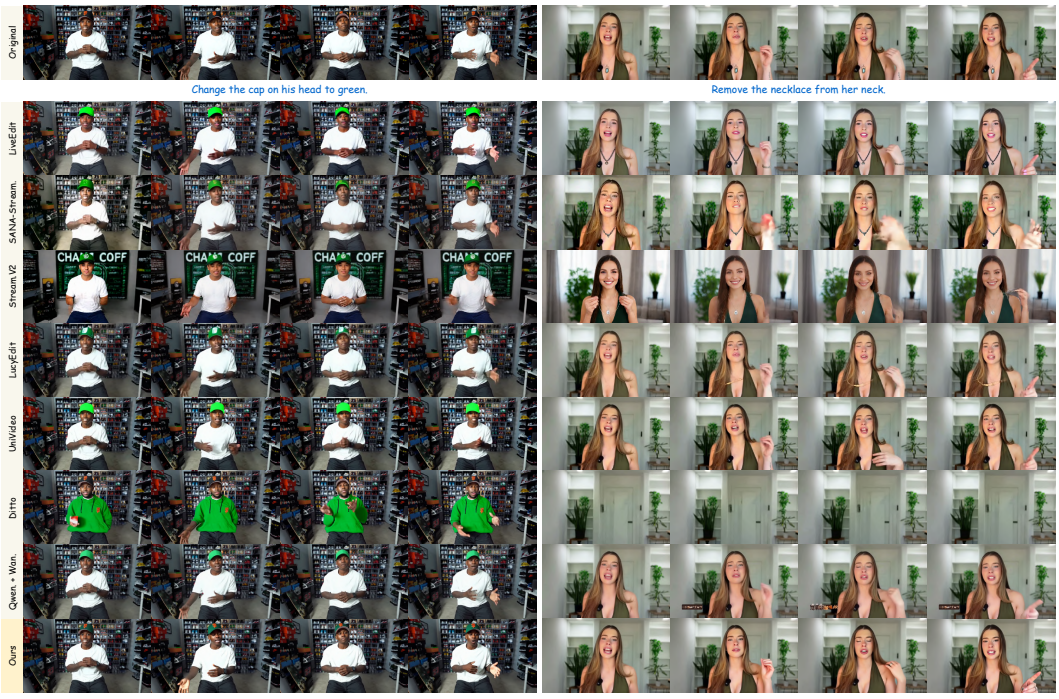


Figure 17: More comparisons between baselines and our EditaLive (3/4).

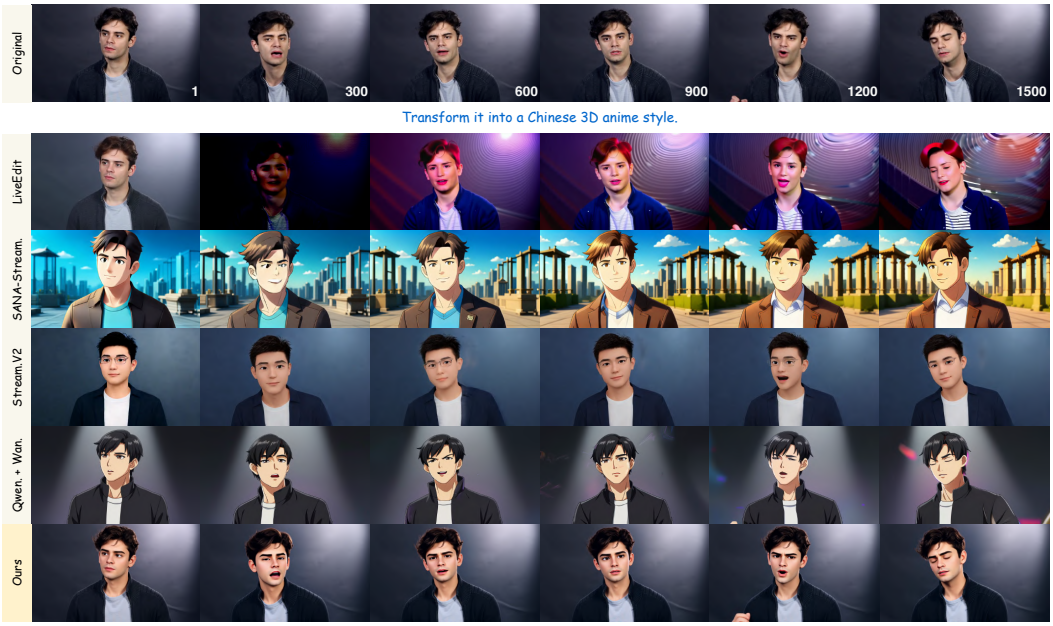


Figure 18: Long-video comparison results (2/2).

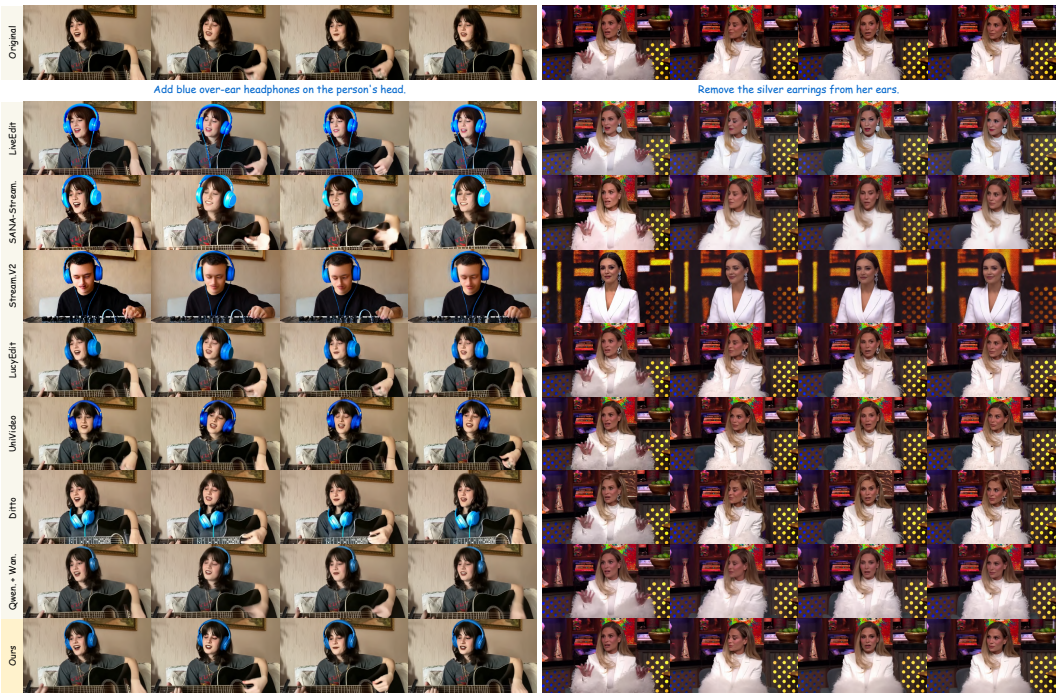


Figure 19: More comparisons between baselines and our EditaLive (4/4).