EDITCAST: ONE-STEP VIDEO EDITING BY BROADCASTING AN EDITED KEYFRAME

Anonymous authors
Paper under double-blind review



Figure 1: **EditCast overview and results.** Top: keyframe edits from FLUX.2 [klein] 4B and Nano Banana 2, with EditCast outputs at frames 16 and 32 (source insets). Bottom: inference speed and DINO similarity to the keyframe.

ABSTRACT

Image editors let users refine a frame's appearance, but carrying revisions through video often requires another costly generation pass. EditCast turns any image editor into a video editor: conditioned on the source clip, an edited keyframe and its temporal slot, and an instruction, our one-step broadcaster predicts the full edited clip in one generator evaluation. We fully fine-tune VACE-1.3B through its native control pathway with only pixel and perceptual losses on sampled temporal windows. At equal training steps, a matched 30-step flow-matching editor fine-tuned from the same checkpoint and interface trails our one-step recipe on every keyframe-following and fidelity measure: by 1.7 dB at the keyframe slot, 2.0 dB at the last frame, and 2.2 dB against the reference clip. On curated KeyEdit-50 (1280×720), with keyframes from four image editors and reference edits, EditCast is **more than 2 dB closer to the keyframe** at the last frame than the strongest propagation baseline. It runs at 6.3 FPS on one H200 (16.8 with TinyVAE), **at least 33 times faster** than all 50-step propagation baselines (87 times with TinyVAE). It accepts one or more keyframes at any slot, follows keyframes from different editors, and preserves 93% of the edit on clips 2.5 times the training length.

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

Image editors (Brooks et al., 2023; Black Forest Labs et al., 2025; Wu et al., 2025; Google DeepMind, 2025; 2026; Black Forest Labs, 2026) let users specify detailed appearance changes on a single frame (Guo & Lin, 2024). Extending these edits to video requires the selected appearance to remain coherent as objects move, turn, or become occluded, and requires broader changes—such as a style transformation or background replacement—to persist across the clip. Today, carrying each revision through a video often requires another time-consuming editing pass (Ku et al., 2024; Geyer et al., 2024). This motivates a practical question: *once an edited keyframe specifies the target appearance, can that edit be broadcast throughout a video in a single generative step?*

We call this objective **keyframe following**: the appearance and scene changes specified by the edited keyframe should persist beyond its temporal position, without reverting to the source, while the source video's motion is retained. Matching the keyframe only at its own position is not enough; nor is changing the source motion to reproduce the desired appearance a faithful video edit.

Prior work already separates the two halves of this problem. Rerender A Video propagates translated keyframes by patch matching; AnyV2V, I2VEdit, and GenProp edit the first frame and regenerate the clip with an image-to-video model; NOVA uses sparse edited keyframes together with the source video; and VACE accepts aligned keyframes as one of several control types (Yang et al., 2023; Ku et al., 2024; Ouyang et al., 2024; Liu et al., 2024; Pan et al., 2026; Jiang et al., 2025). Across these methods, propagation still involves an iterative procedure, such as 50 denoising steps, inversion, or per-clip fine-tuning. Each keyframe revision therefore incurs another full video-generation pass, and most methods accept a keyframe only at the first frame. Few-step editors such as LiveEdit and VDOT reduce sampling steps by distilling a multi-step teacher, typically treating the multi-step model as the quality ceiling and trading some quality for speed (Wang et al., 2026a;b).

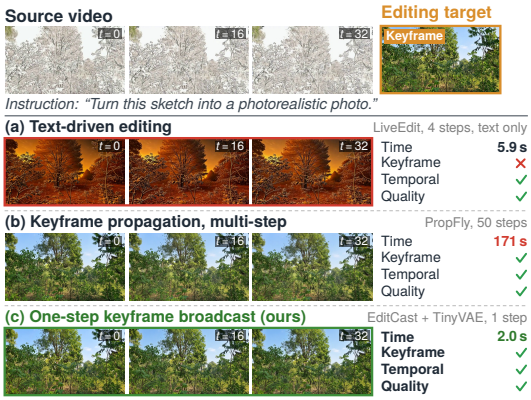

Figure 2: **Three ways to carry an edit through a video.** (a) LiveEdit edits from the text alone and cannot follow the edited keyframe; (b) PropFly follows it with 50 sampling steps; (c) EditCast with TinyVAE follows it in one step. Time: GPU time per 33-frame 1280×720 clip on one H200.

We argue that broadcasting an edit is a different regime from generating a video. Given the source clip and edited keyframe, the target is largely determined: the source supplies motion and the keyframe specifies appearance; the remaining challenge is combining them through occlusion and viewpoint change. This leaves less uncertainty than open-ended video generation and does not call for a sampler designed to draw from a broad distribution. We train a broadcaster directly for one evaluation with pixel supervision. Compared with a multi-step flow-matching editor fine-tuned from the same checkpoint through the same conditioning interface, it better reproduces the edited keyframe at its slot, follows it farther through the clip, realizes more of the edit, and stays closer to the reference clip—at equal training steps and with one-thirtieth as many network evaluations (Section 4.3); moreover, as training continues, the multi-step model increasingly regenerates the edited keyframe instead of trusting it. Thus, for this task, one step is the more accurate choice; its lower latency makes revise-and-preview editing practical without trading away quality. Figure 1 shows the resulting speed and keyframe similarity against the baselines (Figure 2 contrasts the three routes).

We propose EditCast, a one-step video broadcaster that takes a source video, an edited keyframe with its temporal slot, and an edit instruction. We fully fine-tune VACE-1.3B through its native visual-control pathway to predict the edited clip in one generator evaluation, training from the start with pixel and perceptual losses on sampled temporal windows. The image editor can be chosen independently: with keyframes from three editors for the same clip, 88% of the outputs are closest to their own keyframe (Section 4.4). Aligned conditioning supports one or more keyframes at arbitrary slots and transfers image-editing progress to video without retraining the broadcaster. We evaluate control by swapping keyframes while fixing the source and instruction, and measure fol-

lowing separately at the anchor and last frame. On KeyEdit-50 (1280×720), we compare against a matched multi-step editor and few- and many-step baselines, and test four editors, arbitrary and multiple keyframe slots, temporal distance, and clips beyond the training length. At equal training steps, the one-step recipe exceeds the matched multi-step editor by 1.7 dB at the anchor, 2.0 dB at the last frame, and 2.2 dB in reference fidelity, using one-thirtieth as many network evaluations (Section 4.3). Against 50-step propagation, EditCast is over 2 dB closer to the keyframe at the last frame and at least 33× faster (87× with TinyVAE at unchanged last-frame following; Figure 1).

2 Related Work

Video Editing with Temporal Control. A broad line of video-editing work uses diffusion models to make instructed changes while retaining information from the input video. Training-free methods such as FateZero and TokenFlow preserve source structure by reusing internal diffusion features: FateZero fuses attention maps recorded during inversion, whereas TokenFlow propagates features through inter-frame correspondences (Qi et al., 2023; Geyer et al., 2024). More recent video models expose temporal control within a shared generative system. VACE routes video and reference-image conditions through a visual-control pathway, and VideoCoF uses temporal reasoning to ground editing instructions in a clip (Jiang et al., 2025; Yang et al., 2025). These approaches address source preservation and instruction following across time. Our focus is complementary: the user selects the appearance in an edited image, and the video model carries that choice through the clip.

Keyframe-Guided Video Editing. Prior work already separates appearance specification from temporal propagation. Rerender A Video transfers translated keyframes through a clip using temporal-aware patch matching and frame blending (Yang et al., 2023). AnyV2V and I2VEdit edit the first frame and use it to condition image-to-video generation; GenProp likewise starts from an edited frame and uses video generation and feature injection to propagate the change (Ku et al., 2024; Ouyang et al., 2024; Liu et al., 2024). These approaches make the target appearance explicit, but propagation still depends on a generative sampling process or a distinct transfer stage. NOVA conditions video diffusion on sparse edited keyframes together with the source video, while VACE accepts aligned frame-level controls as part of a general video-editing interface (Pan et al., 2026; Jiang et al., 2025). Such interfaces broaden temporal conditioning, though the complete clip is still produced by iterative generation; in first-frame propagation methods, the edited frame is also tied to the start of the clip. EditCast keeps the useful decomposition into image editing and video propagation, while learning the propagation mapping: from the source, edit instruction, and one or more keyframes at their temporal slots, it predicts the edited clip in one generator evaluation.

Few-Step and One-Step Video Editing. A separate line of work reduces the cost of iterative video generation. Consistency and distribution-matching methods provide tools for few-step sampling; CausVid applies them to video generation, while VDOT applies them to video editing (Song et al., 2023; Yin et al., 2024b;a; 2025; Wang et al., 2026b). For video editing, LiveEdit distills an instruction-driven editor into four causal steps, reducing latency while retaining text-based control (Wang et al., 2026a). OSVE demonstrates one-step text-guided video editing, and ChordVideo combines motion-aligned edit fields over multiple evaluations (Lim & Park, 2026; Lao, 2026). These methods establish that editing can be made more efficient, but do not learn the specific mapping studied here: a source clip and a user-edited keyframe at a temporal slot directly to the full edited video in one learned prediction. Section 4.3 tests whether training for this single evaluation is preferable to reducing the steps of a matched multi-step editor.

3 Method

3.1 Task Definition

Given a source clip $A = (A_0, \ldots, A_{T-1})$, an edited keyframe K at slot k, and precomputed UMT5 embeddings c, EditCast predicts an edited clip $\widehat{B}$ that follows the keyframe's appearance while retaining the source dynamics. Training uses a paired target clip B and a randomly sampled keyframe. The causal video VAE encodes a 33-frame clip into 9 latent frames: the first frame alone and each following group of four frames.

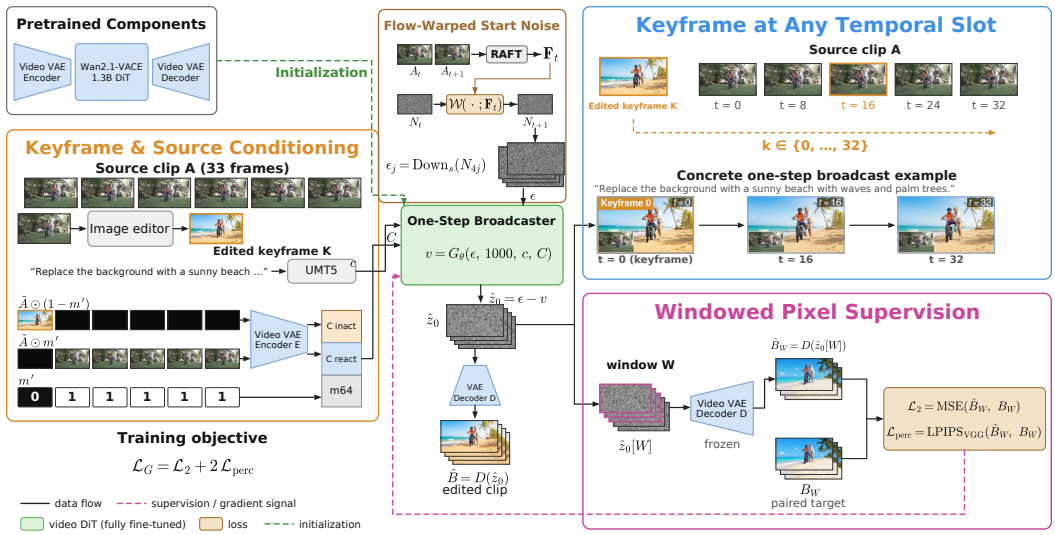

Figure 3: **EditCast overview.** The source video, instruction, and edited keyframe condition a one-step broadcaster. Sampled temporal windows provide pixel and perceptual supervision in training.

Figure 3 summarizes the one-step broadcaster and its windowed pixel supervision.

3.2 Shared Keyframe and Source Conditioning

An image editor supplies the target appearance as a keyframe K, which replaces frame k in source clip A to form $\widetilde{A}$. We define a spatially uniform gate m' that is zero at slot k and one elsewhere. Using the frozen VAE encoder E, we construct two visual conditions:

$$C_{\text{inact}} = E(\widetilde{A} \odot (1 - m')), \tag{1}$$

$$C_{\text{react}} = E(\widetilde{A} \odot m'), \tag{2}$$

$$C = \text{Concat}[C_{\text{inact}}, C_{\text{react}}, m_{64}]. \tag{3}$$

The inactive branch encodes only K (other slots zeroed), while the reactive branch encodes source frames with slot k zeroed (16 channels each). Rearranging the gate as VACE's 8×8 representation m_{64} completes the 96-channel control tensor C. Through VACE's native control pathway (Jiang et al., 2025), `vace_patch_embedding` and `vace_blocks` process C; their outputs are added to designated backbone layers. The noisy-latent input remains 16 channels, and the paired target B is used only for training losses.

3.3 One-Step Broadcast

Initialize G_θ from the original Wan2.1-VACE-1.3B checkpoint and fully fine-tune it, including its VACE conditioning modules. At fixed timestep $t = 1000$, sample Gaussian noise ϵ and predict:

$$\epsilon \sim \mathcal{N}(0, I), \qquad v = G_\theta(\epsilon, 1000, c, C), \qquad z_0 = \epsilon - v. \tag{4}$$

The complete nine-frame latent clip z_0 is decoded by the frozen VAE decoder D into the 33-frame output $\widehat{B} = D(z_0)$. The model predicts the output directly, without latent compositing or copying source content back. It is trained for one evaluation from the start, not distilled; Section 4.3 compares it with a multi-step model using the same backbone and conditioning interface.

3.4 Pixel Supervision on Temporal Windows

At each step, sample a contiguous three-latent-frame window W. Decode it into nine RGB frames $\widehat{B}_W = D(z_0[W])$ and compare them with the corresponding target frames B_W:

$$\mathcal{L}_2 = \text{MSE}(\widehat{B}_W, B_W), \tag{5}$$

$$\mathcal{L}_{\text{perc}} = \text{LPIPS}_{\text{VGG}}(\widehat{B}_W, B_W). \tag{6}$$

Both losses compare full frames. The frozen but differentiable decoder passes gradients to z_0 and G_θ. Resampling W each step supervises every latent frame; only G_θ is trained:

$$\mathcal{L}_G = \mathcal{L}_2 + 2\,\mathcal{L}_{\text{perc}}. \tag{7}$$

3.5 Flow-Warped Start Noise

With i.i.d. start noise, each latent frame of ϵ in Eq. (4) receives an independent noise field, despite the temporal coherence expected of the output. Following Go-with-the-Flow (Burgert et al., 2025) and temporally correlated noise warping (Chang et al., 2024), we transport noise along the source motion. Let $\mathbf{F}_t = \text{RAFT}(A_t, A_{t+1})$ (Teed & Deng, 2020) denote the forward flow between consecutive source frames, estimated at half resolution. We maintain a Gaussian field N_t at s times the latent resolution along each axis ($s = 8$ particles per latent pixel per axis) and advance it as

$$N_0 \sim \mathcal{N}(0, I), \qquad N_{t+1} = \mathcal{W}(N_t; \mathbf{F}_t), \qquad t = 0, \ldots, T-2, \tag{8}$$

where $\mathcal{W}$ moves each particle by its flow vector. Particles arriving at the same cell p are merged using weights ω_i. Empty cells are filled by backward-warping the previous field: a value duplicated n times is scaled by $1/\sqrt{n}$ and receives fresh noise ξ with zero mean within its group of duplicates and variance $1 - 1/n$:

$$N_{t+1}(p) = \begin{cases} \dfrac{\sum_{i \in \mathcal{P}(p)} \omega_i N_t(q_i)}{\sqrt{\sum_{i \in \mathcal{P}(p)} \omega_i^2}}, & \mathcal{P}(p) \neq \emptyset, \\ \dfrac{N_t(p - \mathbf{F}_t(p))}{\sqrt{n}} + \xi(p), & \text{otherwise.} \end{cases} \tag{9}$$

This construction keeps each N_t marginally distributed as $\mathcal{N}(0, I)$ while sharing noise across frames where source motion is coherent. The start noise for latent frame j is the variance-corrected area average at pixel frame $4j$:

$$\epsilon_j = \text{Down}_s(N_{4j}), \qquad j = 0, \ldots, T_{\text{lat}} - 1, \tag{10}$$

replacing the i.i.d. draw in Eq. (4). We use the same warped noise in training and inference; the reported recipe, which uses it, lowers the warping error from 1.89 to 1.81 at equal training steps (Section 5). Appendix A gives implementation details.

3.6 Inference

At inference, one batched VAE encoding constructs C from the source clip, keyframe, and slot (Eq. (3)). We warp the start noise along the source flow (Eqs. (8)–(10)), then evaluate G_θ once at $t = 1000$ and decode its output to obtain $\widehat{B}$ (Eq. (4)), with a fixed seed.

4 Experiments

4.1 Experimental Setup

Experimental setup. We train EditCast on 68,600 source–edited pairs spanning five edit categories, using 33-frame windows cropped to 320×640. The broadcaster is fully fine-tuned for 10,000 steps with flow-warped start noise and evaluated using EMA weights. Evaluation uses KeyEdit-50, a curated benchmark of 50 clips at 1280×720, split into 25 KeyEdit-Img clips with edited keyframes from image editors and 25 KeyEdit-Ref clips with reference edited videos. Each subset contains five clips per edit category drawn from held-out pools; the subsets are mutually disjoint and excluded from training, and the KeyEdit-Img keyframes come from four external image editors rather than from the training pairs. We measure keyframe following on all 50 clips and reference-based quality on KeyEdit-Ref. Data, training details and overlap checks are in Appendix A.

Baselines and metrics. We compare EditCast with representative propagation methods and few- and many-step samplers using each method's closest native input interface, labeled per row to distinguish interface differences from quality differences. Baselines include pixel-aligned keyframe

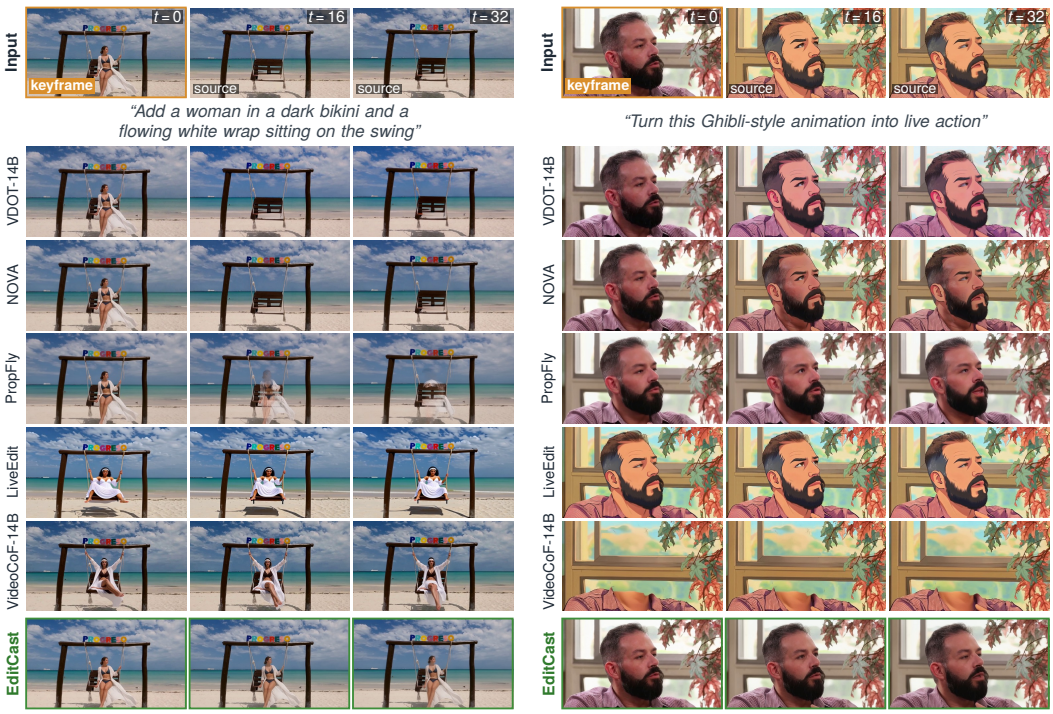

Figure 4: **Qualitative comparison on two benchmark subsets.** Left: KeyEdit-Img; right: KeyEdit-Ref. Each example shows the edited keyframe, source frames, and method outputs.

methods (VDOT-14B (Wang et al., 2026b) at its released four-step setting without classifier-free guidance, and the untrained VACE-1.3B base (Jiang et al., 2025)), first-frame propagation methods (AnyV2V (Ku et al., 2024) and NOVA (Pan et al., 2026)), reference-image propagation with an unaligned keyframe (PropFly (Seo et al., 2026)), and instruction-only editors without keyframe input (LiveEdit (Wang et al., 2026a) and VideoCoF-14B (Yang et al., 2025)). Edit-region annotations are used only for evaluation and never enter the model or training objectives. We report appearance following with respect to the edited keyframe, video quality, temporal behavior, and per-clip GPU inference time under the stated protocol; training variants are compared using per-clip paired differences and paired t statistics. A local vision-language judge (Qwen3-VL-32B) scores instruction compliance, consistency, and visual quality from 1 to 5 on side-by-side source/output frames.

4.2 Main Comparison

Table 1 compares one-step broadcast with the baselines at 1280×720, the native resolution of both test sets; Table 7 in Appendix C repeats the comparison at 832×480, the training resolution of most baselines, with the same ordering of methods. Figure 4 compares EditCast with five baselines covering the three interface types (VDOT-14B, NOVA, PropFly, LiveEdit and VideoCoF-14B) on one clip from each test set; Appendix G compares every baseline on ten more clips covering every edit type (Figures 10–12). The supplementary material has videos of 25 edits, five per edit type.

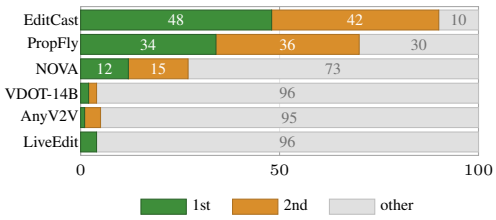

Figure 5: **Blind user study.** Share of 282 trials ranked first, second, or outside the top two.

We report keyframe following as in-mask PSNR to the edited keyframe at the anchor and last frame; video quality as in-region PSNR and LPIPS against the KeyEdit-Ref references; and temporal behavior as warping error under optical flow from the reference clip. Keyframe following is read together with motion: a still clip of the repeated keyframe would maximize it but has zero dynamic degree, whereas EditCast keeps the dynamic degree of the source (0.50 for both; Table 5). The VLM score is the mean (1–5) of Qwen3-VL-32B's instruction-

Method	Steps	Keyframe interface	Keyframe following (KeyEdit-50, dB ↑) anchor	last frame	Video quality (KeyEdit-Ref) PSNR ↑	LPIPS ↓	Temporal (KeyEdit-Ref) EWARP ↓	VLM judge (KeyEdit-50) score ↑	Speed (FPS) ↑ GPU
Copy source	–	–	10.25	10.25	10.78	0.596	2.01	1.39	–
VACE-1.3B, no training♯	50	keyframe	33.58†	10.82	12.20	0.544	3.82	2.92	0.193
VDOT-14B	4	keyframe	32.54†	10.61	12.10	0.544	2.66	2.79	1.22
NOVA♯	50	first frame	33.74†	14.61	17.99	0.320	2.40	3.78	0.166
PropFly-1.3B (Seo et al., 2026)	50	keyframe as reference	25.81	15.45	20.64	0.203	2.38	3.77	0.193
LiveEdit♯	4	instruction	9.87	9.48	10.34	0.626	3.59	2.87	5.63
VideoCoF-14B♯	4	instruction	12.96	12.15	12.93	0.533	2.06	3.21	0.521
fixed native size									
AnyV2V (640 × 320)	50 + 500 inv.	first frame	23.38	15.64	18.26	0.354	4.88	3.63	0.0560
EditCast	1	keyframe in its slot	31.42	17.86	24.28	0.113	1.80	**4.12**	6.30
EditCast + TinyVAE§	1	keyframe in its slot	30.39	17.86	24.05	0.130	2.13	4.09	16.8

Table 1: Main comparison on KeyEdit-50 at 1280×720.

Editor (same checkpoint and conditioning interface)	Sampling	Train. steps	No-reference (KeyEdit-50) TA ↑	BC ↑	MS ↑	DD	AQ ↑	IQ ↑	Keyframe following (KeyEdit-50) anchor ↑	last ↑	edit ratio ↑	Temporal EWARP ↓
Multi-step, flow matching	30 steps	2k	**20.73**	0.959	0.992	0.50	0.588	0.707	30.21	16.10	0.847	2.36
	30 steps	8k	20.69	0.964	0.992	0.52	0.594	**0.714**	28.49	15.88	0.813	2.28
	30 steps	13k	20.63	0.964	0.992	0.52	0.597	0.713	28.46	16.04	0.829	2.23
One-step, development recipe	1 step	2k	20.43	0.965	**0.993**	0.50	**0.599**	0.712	29.93	17.39	0.885	1.98
	1 step	8k	20.37	**0.968**	**0.993**	0.52	0.597	0.711	30.15	**17.90**	**0.913**	1.89
One-step, EditCast (reported)	1 step	10k	20.37	0.967	**0.993**	0.50	0.598	0.709	**31.42**	17.85	0.907	**1.80**

Table 2: One-step broadcast against a multi-step flow-matching editor fine-tuned from the same checkpoint through the same conditioning interface, on KeyEdit-50 at 1280×720 (EMA weights; columns as in Table 4). The multi-step editor trains on the final data index and with EMA 0.999 (Section 4.3); the one-step development recipe shares its effective batch 8 and learning rate; the last row is the reported model (effective batch 16).

compliance, consistency and visual-quality ratings over 50 clips (Appendix B). Speed is 33 frames divided by per-clip GPU time for condition encoding, generation and decoding on one H200; the protocol and per-method settings are in Appendix A. A dagger marks a keyframe copied into frame 0 (compare at the last frame); a sharp marks runs above the model's ≈ 480p training resolution; and a section mark denotes TAEHV TinyVAE (Boer Bohan, 2025) replacing the Wan VAE.

In a blind user study, 14 raters see the instruction, the edited keyframe and the source video, and pick the best and the second-best of six anonymized results in random order (EditCast, PropFly, NOVA, VDOT-14B, AnyV2V and LiveEdit) on 20 random KeyEdit-50 clips each, judging appearance following, temporal coherence and naturalness. Over 282 trials, EditCast is ranked first 48% of the time (95% CI 36–59%, two-way rater and clip bootstrap) and in the top two 90% of the time (82–96%; chance 17% and 33%), against 34% and 70% for PropFly and 12% and 27% for NOVA; every other method is ranked first in at most 4% of trials (Figure 5). On three of the 50 clips the study shows EditCast's 8k development checkpoint.

4.3 Is One Step a Compromise?

Few-step editors are often distilled from a multi-step model, implicitly treating it as the quality ceiling. We test this premise by fine-tuning a 30-step flow-matching editor from the same Wan2.1-VACE-1.3B checkpoint, with the same conditioning interface, learning rate and effective batch as the one-step development recipe; the multi-step editor even trains on the final data index with EMA 0.999, neither of which the development recipe uses. The multi-step editor samples with 30 Euler steps (shift 3.0) without classifier-free guidance. Table 2 reports its checkpoints against the one-step model at equal training steps.

At 8,000 steps, the one-step model is better on every keyframe-following, edit-realization, reference-fidelity and temporal measure: 1.7 dB at the keyframe slot, 2.0 dB at the last frame and 2.2 dB in-region PSNR against the reference clips (24.00 vs. 21.82 dB). Longer multi-step training does not close the gap: the multi-step model increasingly shifts the edited keyframe toward the source appearance, weakening the realized edit. The multi-step editor gains no perceptual quality in exchange: at 8k its VBench consistency, aesthetic and imaging scores stay within 0.004 of the one-step model. Multi-step sampling is therefore not a quality ceiling for keyframe broadcast, and the reported model improves keyframe-slot fidelity and warping error further (31.42 dB, EWARP 1.80).

	DINO similarity $\times 100$		Full-frame PSNR (dB)	
	frame 16	frame 32	frame 16	frame 32
to its own keyframe	94.05	92.69	17.74	16.78
to the other two keyframes	86.95	85.80	15.17	14.68
margin	+7.09	+6.89	+2.57	+2.10
own keyframe is the closest (of 120)	86 %	88 %	75 %	72 %

Table 3: Keyframe swap: 40 source clips, three keyframes from different editors per clip, one output per keyframe with source, instruction and seed fixed. Chance for "closest" is 33 %; the margin is positive on all 40 clips for every column.

4.4 Does the Keyframe Determine the Output?

This is the central control experiment: we fix the source video, edit instruction and random seed, vary only the edited keyframe, and compare each output with its corresponding keyframe and the other candidates. Keyframe-only, text-only and unedited-keyframe conditions test whether following is driven by the text branch.

Keyframe swap. We evaluate 40 clip–instruction pairs, each with three keyframes. The 25 KeyEdit-Img clips are edited by FLUX.2 [klein], Nano Banana Pro and Nano Banana 2; for 15 instructions on 10 additional clips, keyframes are generated by FLUX.2 [klein], FLUX.1 Kontext and Qwen-Image-Edit. For each editor, we select from three seeds a keyframe that differs visibly from the other two. Each output is compared with all three keyframes for its clip (Table 3). At frame 32, 88% of the 120 outputs are closest by DINO similarity to their own keyframe (chance is 33%). The own-keyframe similarity margin is positive for all 40 pairs, although each clip's three keyframes are similar (mean pairwise DINO similarity 89.5). Figure 8 shows representative swaps.

Keyframe versus text. On KeyEdit-50, removing the instruction (using the empty-prompt embedding from training) leaves the output nearly unchanged: last-frame following is 17.71 dB versus 17.86 dB with full conditioning, and the edit ratio is 0.88 in both cases. Removing the keyframe (using the empty reference from training) lowers following to 13.52 dB and the edit ratio to 0.51. With the instruction retained but the unedited source frame used as the keyframe, the model reproduces the source video (30.9 dB full-frame PSNR to the source; edit ratio 0.09, compared with 0 for the source itself). This shows that the keyframe, rather than the text, primarily determines the edit.

4.5 How Long Does Following Persist?

Anchor reconstruction and propagation are reported separately, since strong performance near the keyframe does not by itself show propagation. Following depends weakly on source motion, and EditCast's lead over PropFly is largest on the fastest clips (+3.4 dB at the last frame; Appendix E).

Following over distance. Figure 7 (Appendix D) summarizes the single-keyframe runs from the keyframe-slot ablation (Section 5), grouped by distance from each frame to the keyframe. EditCast reaches 28.8 dB keyframe agreement at the keyframe slot; as distance increases, the edit continues to propagate to more distant frames, retaining 22.6 dB at a distance of 32 frames.

Beyond the training length. EditCast is trained on 33-frame clips but runs directly on longer ones. On the 49-frame KeyEdit-Ref clips its first 33 frames match the 33-frame run (24.16 vs. 24.11 dB in-region PSNR), and on the 14 clips of 81 frames, 2.5 times the training length, it still realizes 93 % of the reference edit (edit ratio 0.93).

5 Ablation Studies

Controlled ablations are run on a shorter development recipe (Appendix A); each changes one factor of it and is read on KeyEdit-50 with the EMA weights. The reported recipe (Section 4.1) is listed last. We report the no-reference measures used by LiveEdit (Wang et al., 2026a) (CLIP text alignment, five VBench (Huang et al., 2024) dimensions), keyframe following and warping error.

		No-reference (KeyEdit-50)						Keyframe following (KeyEdit-50)			Temporal
Variant	Steps	TA ↑	BC ↑	MS ↑	DD	AQ ↑	IQ ↑	anchor ↑	last ↑	edit ratio ↑	EWARP ↓
Development recipe	2k	20.43	0.965	0.993	0.50	0.599	0.712	29.93	17.39	0.885	1.98
EMA off (live weights)	2k	20.42	0.961	0.993	0.50	0.598	0.716	29.39	17.30	0.876	1.93
Development recipe	8k	20.37	0.968	0.993	0.52	0.597	0.711	30.15	17.90	0.913	1.89
live weights (no EMA read-out)	8k	20.38	0.967	0.993	0.52	0.597	0.709	29.01	17.78	0.910	1.90
learning rate 2×10^{-6} (live)	8k	20.82	0.942	0.992	0.52	0.587	0.702	30.90	14.48	0.753	2.27
Reported recipe	2k	20.55	0.965	0.993	0.50	0.597	0.711	30.89	17.66	0.893	1.90
Reported recipe	8k	20.41	0.967	0.993	0.52	0.599	0.709	31.28	17.74	0.899	1.81
Reported recipe (EditCast)	10k	20.37	0.967	0.993	0.50	0.598	0.709	31.42	17.85	0.907	1.80

Table 4: Training ablations on KeyEdit-50 (EMA weights unless stated); each indented row changes one factor of the row above it at the same step count (the learning-rate run is read with live weights, so it is compared with the live weights of the development run), and the last block is the reported recipe (effective batch 16, so its step counts cover twice the samples). TA: CLIP text alignment ($\times 100$); BC, MS, DD, AQ, IQ: VBench background consistency, motion smoothness, dynamic degree, aesthetic and imaging quality. Keyframe following: in-mask PSNR (dB) to the edited keyframe at the anchor and last frame; edit ratio: realized over reference edit area. EWARP on KeyEdit-Ref.

Keyframe slot. Aligned per-slot conditioning lets EditCast take the keyframe at any slot, and at several (Table 8, Appendix D). On KeyEdit-Ref, a keyframe at an interior slot (8, 16 or 24) raises in-region PSNR by 0.46–0.69 dB over the first slot, and keyframes at both ends, a setting never seen in training, add 1.28 ± 0.32 dB on all 25 clips. VDOT, which shares the aligned interface, instead loses 0.6–0.7 dB away from the first slot.

Reported recipe. At equal training steps, the reported recipe (Section 4.1, Appendix A) raises keyframe fidelity at the anchor by 1.1 dB over the development recipe (31.28 vs. 30.15 dB at 8k, paired $t = 8.2$) and lowers the warping error from 1.89 to 1.81 ($t = -5.4$), while keeping last-frame following and the no-reference quality; at the end of training (10k) EditCast reaches 31.42 dB and 1.80. EMA and the learning rate are discussed in Appendix A.

6 Generalization and Limitations

Keyframe editor and resolution. EditCast follows keyframes from different editors: in the keyframe swap of Section 4.4, 88% of the outputs are closest to their own keyframe across three editors, and on 73 camera and creative edits from OpenVE-Bench (He et al., 2025), keyframes from Qwen-Image-Edit (Wu et al., 2025) and FLUX.2 [klein] 4B (Black Forest Labs, 2026), which differ by 11.7 dB, yield the same realized edit ratio (0.46) and close judge scores (Appendix F). The 480×832 tier keeps last-frame following within 0.16 dB at 41 % of the GPU time, so a revise-and-preview cycle with FLUX.2 [klein] takes about 2.5 s on one H200.

Edit categories. EditCast follows the keyframe to the last frame better than NOVA and PropFly in every edit category (Table 9, Appendix E), with the largest leads over PropFly on local removal (+5.2 dB) and addition (+3.4 dB).

Limitation: camera viewpoint changes. A single aligned keyframe does not convey a viewpoint change, left to future work (Figure 9).

7 Conclusion

EditCast separates video editing into an image edit and a one-step broadcast by a fully fine-tuned transformer trained with pixel and perceptual supervision. At equal training steps this single step reproduces the keyframe more faithfully and follows it further than 30-step sampling of the same backbone, so for keyframe broadcast the multi-step sampler is not a quality ceiling. Against propagation baselines, EditCast is more than 2 dB closer to the keyframe at the last frame, receives the highest vision-language judge score and is ranked first in 48% of user-study trials, while running at least 33$\times$ faster than 50-step propagation. The broadcaster accepts the keyframe at any slot and from different editors and keeps most of the edit beyond the training length. With FLUX.2 [klein], a full revise-and-preview cycle takes about 2.5 s of GPU time.

ETHICS STATEMENT

This work involves human raters and the editing of videos that show identifiable people, and we address both. The blind user study (Section 4.2) asked raters only for ranked preferences between anonymized results: apart from a self-chosen name used to group each rater's answers, no personal data, demographics or free-text responses were recorded, the names are not released, and raters could stop at any point. Training and evaluation clips come from OpenVE-3M, Ditto-1M and the WebNature and WebFilm video collections, used under their respective licences; the supplementary videos contain only frames from these sources and the outputs of the compared methods. Like any video editor, EditCast can change the people, objects and scenes of real footage, including a person's apparent age, and could be misused to fabricate misleading videos. Its outputs are synthesized, not evidence: they should be disclosed as AI-edited and never presented as unedited footage.

REPRODUCIBILITY STATEMENT

Section 3 gives the conditioning construction (Eq. 3), the one-step prediction (Eq. 4), the flow-warped start noise (Eqs. 8–10) and the training objective (Eq. 7). The broadcaster is a full fine-tune of the public Wan2.1-VACE-1.3B checkpoint with the public Wan2.1 VAE and UMT5 encoder; Section 4.1 and Appendix A list the training-data pipeline with every filtering rule, and the optimizer settings, batch sizes and learning-rate schedule. The construction of both halves of KeyEdit-50, the metric definitions, the GPU-time protocol and the user-study design are specified in Sections 4.1 and 4.2 and Appendix A, and the VLM-judge protocol in Appendix B. We will release the code, the training configurations, the KeyEdit-50 clip lists, keyframes and scoring masks, the metric and timing scripts, and the model weights.

AI USE STATEMENT

Generative AI enters this work in two roles. As part of the method and its evaluation, image editors produce the keyframes, vision-language models write instructions, filter training pairs and judge outputs, and a video model propagates the H3 training pairs; each use is described where it occurs (Section 4.1, Appendices A and B). As research assistants, LLM-based coding agents were used extensively, under the authors' direction, to write and run training, evaluation, timing and figure scripts, to organize experiment logs, and to draft and edit text, which the authors reviewed and revised. The research direction, the method and all experimental choices were decided by the authors; every reported number was produced by the pipelines described in this paper and checked against the run logs, and the authors take full responsibility for the content of this work.

REFERENCES

Black Forest Labs. FLUX.2 [klein] 4B. https://huggingface.co/black-forest-labs/FLUX.2-klein-4B, 2026.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. FLUX.1 Kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025. URL https://arxiv.org/abs/2506.15742.

Ollin Boer Bohan. TAEHV: Tiny autoencoder for hunyuan video and wan. https://github.com/madebyollin/taehv, 2025.

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*, 2023. URL https://arxiv.org/abs/2211.09800.

Ryan Burgert, Yuancheng Xu, Wenqi Xian, Oliver Pilarski, Pascal Clausen, Mingming He, Li Ma, Yitong Deng, Lingxiao Li, Mohsen Mousavi, Michael Ryoo, Paul Debevec, and Ning Yu. Go-

with-the-flow: Motion-controllable video diffusion models using real-time warped noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Pascal Chang, Jingwei Tang, Markus Gross, and Vinicius C. Azevedo. How i warped your noise: a temporally-correlated noise prior for diffusion models. In *International Conference on Learning Representations (ICLR)*, 2024.

Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. TokenFlow: Consistent diffusion features for consistent video editing. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2307.10373.

Google DeepMind. Gemini 3 Pro Image model card. https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-Pro-Image-Model-Card.pdf, 2025.

Google DeepMind. Gemini 3.1 Flash Image. https://deepmind.google/models/gemini-image/flash/, 2026.

Qin Guo and Tianwei Lin. Focus on your instruction: Fine-grained and multi-instruction image editing by attention modulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6986–6996, 2024.

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.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

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*, 2025. URL https://arxiv.org/abs/2503.07598.

Max Ku, Cong Wei, Weiming Ren, Harry Yang, and Wenhu Chen. AnyV2V: A tuning-free framework for any video-to-video editing tasks. *Transactions on Machine Learning Research*, 2024. URL https://arxiv.org/abs/2403.14468.

Zhiqiang Lao. ChordVideo: One-step, training-free, temporally consistent video editing via low-energy transport. *arXiv preprint arXiv:2608.00769*, 2026. URL https://arxiv.org/abs/2608.00769.

Habin Lim and Gyeong-Moon Park. OSVE: One step video editing with one step diffusion models. *arXiv preprint arXiv:2607.19895*, 2026.

Shaoteng Liu, Tianyu Wang, Jui-Hsien Wang, Qing Liu, Zhifei Zhang, Joon-Young Lee, Yijun Li, Bei Yu, Zhe Lin, Soo Ye Kim, and Jiaya Jia. Generative video propagation. *arXiv preprint arXiv:2412.19761*, 2024.

Wenqi Ouyang, Yi Dong, Lei Yang, Jianlou Si, and Xingang Pan. I2VEdit: First-frame-guided video editing via image-to-video diffusion models. *arXiv preprint arXiv:2405.16537*, 2024.

Tianlin Pan, Jiayi Dai, Chenpu Yuan, Zhengyao Lv, Binxin Yang, Hubery Yin, Chen Li, Jing Lyu, Caifeng Shan, and Chenyang Si. NOVA: Sparse control, dense synthesis for pair-free video editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://arxiv.org/abs/2603.02802.

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*, 2023. URL https://arxiv.org/abs/2303.09535.

Wonyong Seo, Jaeho Moon, Jaehyup Lee, Soo Ye Kim, and Munchurl Kim. PropFly: Learning to propagate via on-the-fly supervision from pre-trained video diffusion models. *arXiv preprint arXiv:2602.20583*, 2026.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning*, 2023. URL https://arxiv.org/abs/2303.01469.

Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision (ECCV)*, 2020.

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

Yutong Wang, Haiyu Zhang, Tianfan Xue, Yu Qiao, Yaohui Wang, Chang Xu, and Xinyuan Chen. VDOT: Efficient unified video creation via optimal transport distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9273–9283, 2026b. URL https://openaccess.thecvf.com/content/CVPR2026/html/Wang_VDOT_Efficient_Unified_Video_Creation_via_Optimal_Transport_Distillation_CVPR_2026_paper.html.

Chenfei Wu et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

Shuai Yang, Yifan Zhou, Ziwei Liu, and Chen Change Loy. Rerender a video: Zero-shot text-guided video-to-video translation. In *ACM SIGGRAPH Asia Conference Proceedings*, 2023. URL https://arxiv.org/abs/2306.07954.

Xiangpeng Yang, Ji Xie, Yiyuan Yang, Yan Huang, Min Xu, and Qiang Wu. Unified video editing with temporal reasoner. *arXiv preprint arXiv:2512.07469*, 2025. URL https://arxiv.org/abs/2512.07469.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Frédo Durand, and William T. Freeman. Improved distribution matching distillation for fast image synthesis. *arXiv preprint arXiv:2405.14867*, 2024a. URL https://arxiv.org/abs/2405.14867.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo 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*, 2024b. URL https://arxiv.org/abs/2311.18828.

Tianwei Yin, Qiang Zhang, Richard Zhang, William T. Freeman, Frédo 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*, 2025. URL https://arxiv.org/abs/2412.07772.

A Training and Evaluation Details

No-reference comparison. Table 5 applies LiveEdit's no-reference protocol (Wang et al., 2026a) (CLIP text alignment and five VBench (Huang et al., 2024) dimensions) to every method of Table 1 over the 50 KeyEdit-50 clips. EditCast has the highest background consistency, the highest motion smoothness (shared with VideoCoF-14B) and near-best aesthetic quality; instruction-only editors, which edit toward the text rather than a keyframe, score higher on text alignment. These measures do not rank keyframe following: the calibration rows, a still keyframe clip and the unchanged source, score close to the best methods.

Method	TA ↑	BC ↑	MS ↑	DD	AQ ↑	IQ ↑
VACE-1.3B, no training	19.37	0.908	0.988	0.52	0.567	0.709
VDOT-14B (4 steps)	18.84	0.915	0.990	0.52	0.587	0.704
NOVA	19.38	0.938	0.992	0.54	0.588	0.706
PropFly-1.3B	20.84	0.964	0.991	0.54	0.597	0.704
LiveEdit	22.56	0.958	0.989	0.60	0.603	0.731
VideoCoF-14B	19.41	0.953	0.993	0.56	0.536	0.721
AnyV2V (640×320)	20.28	0.949	0.983	0.54	0.595	0.688
EditCast	20.37	0.967	0.993	0.50	0.598	0.709
calibration						
Copy source	18.72	0.963	0.992	0.50	0.558	0.705
Keyframe repeated $\times 33$	20.31	1.000	0.998	0.00	0.564	0.718

Table 5: No-reference measures on KeyEdit-50 at 1280×720 (mean over 50 clips). TA: CLIP text alignment ($\times 100$); BC, MS, DD, AQ, IQ: VBench background consistency, motion smoothness, dynamic degree, aesthetic and imaging quality.

Main-table protocol. Table 1 measures keyframe following as the in-mask PSNR of each output frame to the edited keyframe (anchor and last frame), video quality inside the edit region against the KeyEdit-Ref reference clips, and temporal behavior as the whole-frame warping error under optical flow from the reference clip. Every method receives 1280×720 input except AnyV2V, whose released model is fixed to 640×320 (output resized to 1280×720 for scoring, as in Tables 5 and 6; GPU time measured on 1280×720 input). Each baseline maps our inputs onto its closest native interface. Speed is 33 frames divided by the per-clip GPU time for condition encoding (VAE and text), generation and decoding, on one H200 over the same ten clips, with text encoded online and model loading excluded; each method uses its released implementation. [†]The keyframe is written into frame 0 unchanged (NOVA uses it as its image-to-video first-frame condition), so the anchor value measures a copy; compare these rows at the last frame. [♯]Released for about 480p; LiveEdit's fixed latent size and per-frame KV-cache length are set to their 1280×720 values. [§]TAEHV tiny autoencoder (Boer Bohan, 2025) in place of the Wan VAE (Appendix F). In Table 7 every method runs at 832×480 (VDOT, NOVA and PropFly letterboxed, the others resized) except AnyV2V at 640×320; all outputs are scored at 1280×720.

Training data. Figure 6 summarizes the pipeline. *Public pairs.* OpenVE-3M contributes a local-change pool (27,881 candidates) and 18,000 candidates in each of its other four categories; Ditto-1M contributes its local subset (48,969 candidates). Automatic gates keep pairs whose edited area covers 1–35% of the frame (background and style exempt from the upper bound), with a bitrate of at least 1 Mbps, no flicker, an EditScore above 6 and a VLM-confirmed edit; Qwen2.5-VL-7B then confirms the edit on the first source and edited frames. *H3 pairs.* From 8,285 WebNature and WebFilm clips of at least 124 frames without scene cuts, Qwen3-VL-8B writes an instruction, FLUX.2 [klein] 9B edits frame 0 (for style also the middle and last frames), and MiniMax-H3 propagates the edited keyframe from pure noise; hand review keeps 2,075 pairs at 1024×1024. *Assembly.* After deduplication and removal of every evaluation clip and its candidate pool, categories are rebalanced toward the proportions of Señorita-2M (each pair repeated at most twice, style subsampled, H3 pairs repeated twice), giving 68,600 unique pairs and 90,844 samples per epoch: local change 34.5%, addition 25.7%, removal 19.8%, background 15.7%, style 4.3%.

Training. Each sample is a random 33-frame window with a random 320×640 crop at native scale that contains enough of the edit. Training uses bf16, AdamW ($\beta = (0.9, 0.95)$, weight decay 0.01), gradient clipping at 1.0 and a batch of 16 (eight devices, two samples each). The learning rate is 2×10^{-5} for the first 70 % of 10,000 steps, followed by a half cosine to $0.1\times$; we evaluate

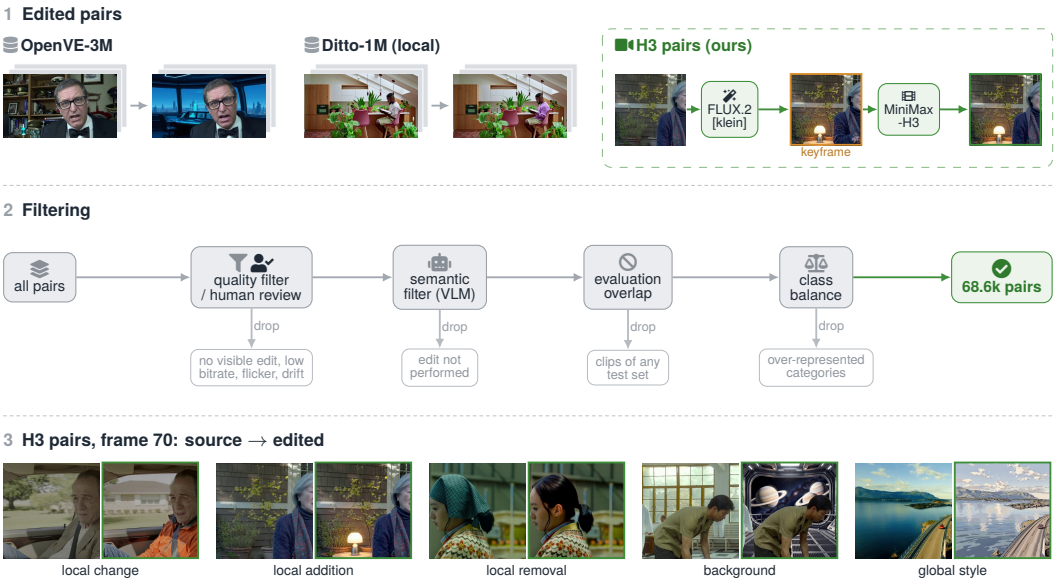

Figure 6: **Training data and filtering overview.** Top: sources of edited pairs; middle: filtering to 68,600 pairs; bottom: representative H3 examples from five edit categories.

EMA weights (decay 0.999). The VAE is frozen and UMT5 embeddings are precomputed. The prompt is replaced by the empty-string embedding with probability 0.3 and the keyframe reference is dropped with probability 0.1. One keyframe per clip is placed at a uniformly random slot and, with probability 0.5, photometrically perturbed (brightness, contrast, saturation, blur or sharpening, JPEG compression, noise); with probability 0.1 each it is taken from a jittered slot or from outside the window. Multiple keyframes are never seen in training. With probability 0.18 (Ditto-1M local pairs) or 0.25 (background changes) a pair is reversed, the edited clip becoming the source and the prompt being dropped, unless its reverse is already in the index or it is an H3 pair. OpenVE-Bench is used only for the 73 camera and creative clips of Section 6; none of their source clips occurs in the training index (checked by path, content hash and perceptual near-duplicate search).

Disabled components. The reported model is trained with Eq. (7) alone: no distribution matching (teacher, score or fake network), no discriminator and no latent compositing.

Benchmark construction and overlap checks. Neither subset overlaps the other or the training index, as checked by file path, content hash and a perceptual near-duplicate search over first and middle frames of all 112,055 training clip files.

Metrics. Keyframe following is calibrated against copying the source unchanged and against repeating the keyframe as a still clip. Training variants are compared by per-clip paired differences and paired t statistics. The reported model is stable across seeds: over three seeds on KeyEdit-50 every metric moves by at most 0.07 dB (0.01 in warping error), and outputs of different seeds agree at 38 dB PSNR on average.

Development recipe. The ablations of Section 5 and the one-step rows of Table 2 use a development recipe: effective batch 8, constant learning rate 2×10^{-5}, EMA 0.99, i.i.d. start noise, the training corpus without the H3 pairs and without direction reversal.

EMA and learning rate. EMA weights follow the anchor 0.54 dB more closely than the live weights of an otherwise identical run and raise background consistency (paired $t = 2.8$ and 4.0). The learning rate of 2×10^{-5} matters: at the 2×10^{-6} of prior streaming editors the broadcaster stays close to the stock weights (relative distance 2.0×10^{-3} after 8,000 steps, against 1.4×10^{-2}), and against the live weights of the development run its last-frame following is 3.3 dB lower ($t = -4.9$) and its realized edit ratio 0.75 instead of 0.91.

Flow-warped start noise. Implementation of Section 3.5: RAFT (Teed & Deng, 2020) flow (12 iterations) between consecutive source frames at half resolution drives a variance-preserving particle

warper (8×8 particles per latent pixel) that keeps every frame marginally $\mathcal{N}(0, I)$; latent frame k takes the warped noise of source frame $4k$. The same noise is used in training and inference.

B VISION-LANGUAGE JUDGE

Qwen3-VL-32B-Instruct judges each output against its source on eight side-by-side frame pairs with the edit instruction, and scores instruction compliance (IC), consistency (CF) and visual quality (VQ) from 1 to 5; the score is their mean. Over the 50 KeyEdit-50 clips EditCast scores highest, ahead of every baseline with $p \leq 0.006$ (paired over clips; Table 6), and the tiny autoencoder keeps the score ($+0.03 \pm 0.12$). An unedited copy of the source scores 1.08 on instruction compliance.

Method	Score ↑	IC ↑	CF ↑	VQ ↑	EditCast − method
EditCast	4.12	4.60	3.88	3.88	–
EditCast + TinyVAE	4.09	4.60	3.82	3.86	$+0.03 \pm 0.12$
NOVA	3.78	4.20	3.52	3.62	$+0.34 \pm 0.23$
PropFly-1.3B	3.77	4.14	3.60	3.58	$+0.35 \pm 0.24$
AnyV2V	3.63	4.34	3.44	3.10	$+0.49 \pm 0.22$
VideoCoF-14B	3.21	3.50	2.78	3.34	$+0.91 \pm 0.41$
VACE-1.3B, no training	2.92	2.84	3.12	2.80	$+1.20 \pm 0.31$
LiveEdit	2.87	3.12	2.64	2.84	$+1.25 \pm 0.42$
VDOT-14B (4 steps)	2.79	2.70	2.76	2.90	$+1.33 \pm 0.34$
Copy source	1.39	1.08	1.20	1.90	$+2.73 \pm 0.32$

Table 6: Qwen3-VL-32B judge on KeyEdit-50 at 1280×720 (mean over 50 clips). Last column: paired difference in score with 95 % interval.

C COMPARISON AT 832×480

Table 7 repeats Table 1 at 832×480, the training resolution of most baselines and EditCast's fast tier, with the same metrics, symbols and GPU-time protocol.

D KEYFRAME SLOT

Table 8 lists the keyframe-slot sweep (Section 5); Figure 7 pools its runs by keyframe distance.

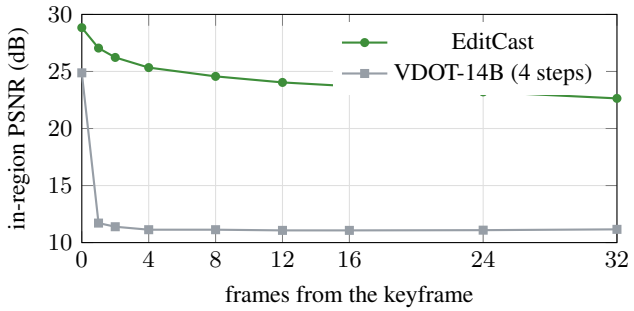

Figure 7: **Keyframe agreement persists across temporal distance.** In-region PSNR to the keyframe, grouped by frame-to-keyframe distance and pooled over five keyframe slots.

E FOLLOWING BY EDIT CATEGORY AND SOURCE MOTION

Source motion. Splitting KeyEdit-50 into terciles of source motion (cut-offs 0.42 and 1.03 px per frame of mean RAFT flow at 640×360), EditCast keeps keyframe agreement within 2.7 dB between frames 8 and 32 in every tercile (last frame 17.85 / 19.02 / 16.76 dB for low / middle / high motion; Spearman -0.15 per clip), and its last-frame lead over PropFly is largest on the fastest clips (+3.4 dB, against +2.5 and +1.4 dB).

Method	Steps	Keyframe interface	Keyframe following (KeyEdit-50, dB ↑) anchor	last frame	Video quality (KeyEdit-Ref) PSNR ↑	LPIPS ↓	Temporal (KeyEdit-Ref) EWARP ↓	Speed (FPS) ↑ GPU
Copy source	–	–	10.25	10.25	10.78	0.596	2.01	–
VACE-1.3B, no training	50	keyframe	30.62†	10.64	11.82	0.581	3.70	0.656
VDOT-14B	4	keyframe	29.52†	10.81	12.40	0.545	2.59	3.65
NOVA	50	first frame	30.82†	14.57	17.63	0.347	2.54	0.692
PropFly-1.3B (Seo et al., 2026)	50	keyframe as reference	24.27	15.84	20.15	0.227	2.41	0.664
LiveEdit	4	instruction	9.91	9.63	10.44	0.643	3.02	14.4
VideoCoF-14B	4	instruction	13.00	12.25	12.94	0.540	2.06	1.94
fixed native size								
AnyV2V (640 × 320)	50 + 500 inv.	first frame	23.38	15.64	18.26	0.354	4.88	0.140
EditCast	1	keyframe in its slot	29.30	17.70	23.64	0.151	1.93	15.5
EditCast + TinyVAE§	1	keyframe in its slot	28.30	17.69	23.36	0.178	2.31	44.6

Table 7: The same comparison at 832×480, the training resolution of most baselines and EditCast's fast tier; metrics, symbols and speed as in Table 1, per-method settings in Appendix A.

	EditCast				VDOT-14B (4 steps)	
Keyframe slot	PSNR ↑	LPIPS ↓	PSNR_{out} ↑	EWARP ↓	PSNR ↑	EWARP ↓
0 (first)	24.28	0.113	31.24	1.80	12.10	2.66
8	24.85	0.109	31.16	1.89	11.51	3.35
16 (middle)	24.96	0.107	31.38	1.87	11.40	3.53
24	24.74	0.110	31.43	1.86	11.51	3.42
32 (last)	23.97	0.125	31.33	1.80	11.40	2.76
0 and 32	25.56	0.093	31.91	1.89	–	–

Table 8: Keyframe slot on KeyEdit-Ref (Section 5). The keyframe is the reference frame at the given slot; PSNR and LPIPS are measured inside the edit region, PSNR_{out} outside it. EditCast: reported model; VDOT: mask-free, 1280×720.

F KEYFRAME EDITOR, RESOLUTION TIER AND INTERACTION COST

Tiny autoencoder. The TinyVAE rows of Tables 1 and 7 replace both VAE passes by TAEHV. At 1280×720 this keeps last-frame following (17.86 dB for both) and the judge score (4.09 vs. 4.12), is within 0.23 dB in-region PSNR (24.05 vs. 24.28) and cuts GPU time $2.7\times$ (1.96 vs. 5.24 s).

Keyframe editor. EditCast follows keyframes from different editors. On 73 camera and creative edits from OpenVE-Bench (He et al., 2025), keyframes from Qwen-Image-Edit (Wu et al., 2025) (40 steps, 17.3 s) and from FLUX.2 [klein] 4B (Black Forest Labs, 2026) (4 steps, 0.91 s at 1280×720) differ strongly (11.7 dB PSNR, LPIPS 0.55), yet yield the same realized edit ratio (0.46) and close judge scores (3.38 vs. 3.51). We therefore use the fast FLUX.2 [klein] in the pipeline.

Keyframe as the control interface. Editing one frame gives pixel-level control over the appearance, position and extent of an edit, which an instruction specifies only through the video model's interpretation, and it makes iteration cheap: a revision re-runs the image editor and one broadcast pass. With FLUX.2 [klein] a revise-and-preview cycle takes about 2.5 s of GPU time at 480×832 on one H200 (0.40 s keyframe, 2.13 s broadcast; 6.2 s at 1280×720), and a hand-retouched keyframe costs only the broadcast. The single-keyframe interface suits appearance, local and creative edits; viewpoint changes lie outside a single aligned keyframe (Figure 9).

Resolution. Broadcasting at 480×832 and upscaling to 1280×720 keeps last-frame keyframe following within 0.16 dB (17.70 vs. 17.86) and in-region PSNR within 0.64 dB (23.64 vs. 24.28 dB) at 41% of the GPU time (2.13 vs. 5.24 s), which makes the lower tier well suited to previews.

G ADDITIONAL FIGURES

Figures 10–12 compare every baseline of Table 1 on ten more clips in the format of Figure 4, grouped by edit type; Figure 8 shows keyframe swaps from Section 4.4, and Figure 9 illustrates the limitation discussed in Section 6.

last-frame agreement (dB)	local change	local addition	local removal	background change	global style	all
NOVA	16.79	10.70	19.16	14.59	11.82	14.61
PropFly-1.3B	18.57	14.13	15.20	15.62	13.72	15.45
EditCast	**18.90**	**17.56**	**20.39**	**18.16**	**14.27**	**17.86**

Table 9: Keyframe following at the last frame per edit category on KeyEdit-50 (10 clips per category), all methods at 1280×720.

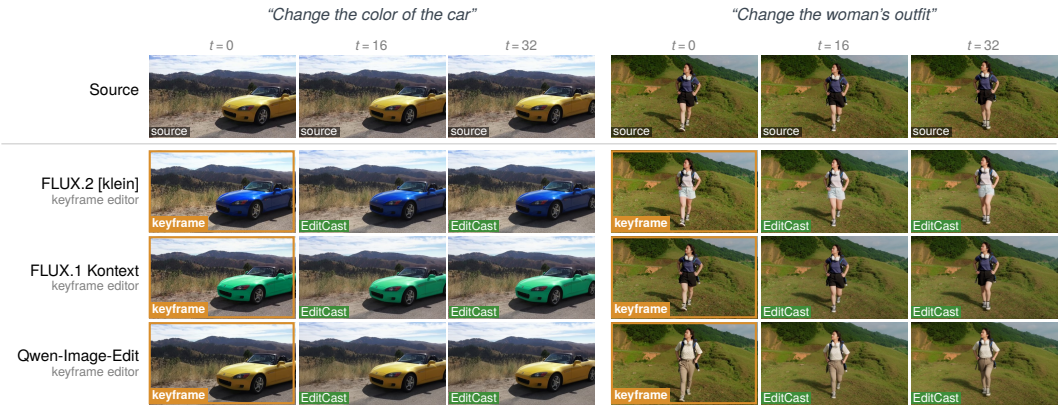

Figure 8: **The edited keyframe determines the output.** Each column fixes the source and instruction while varying the keyframe; orange frames mark the edits.

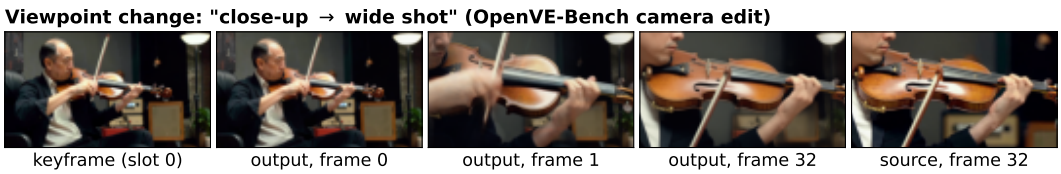

Figure 9: **A failure under viewpoint change.** The output matches the keyframe at its slot but retains the source framing afterward.

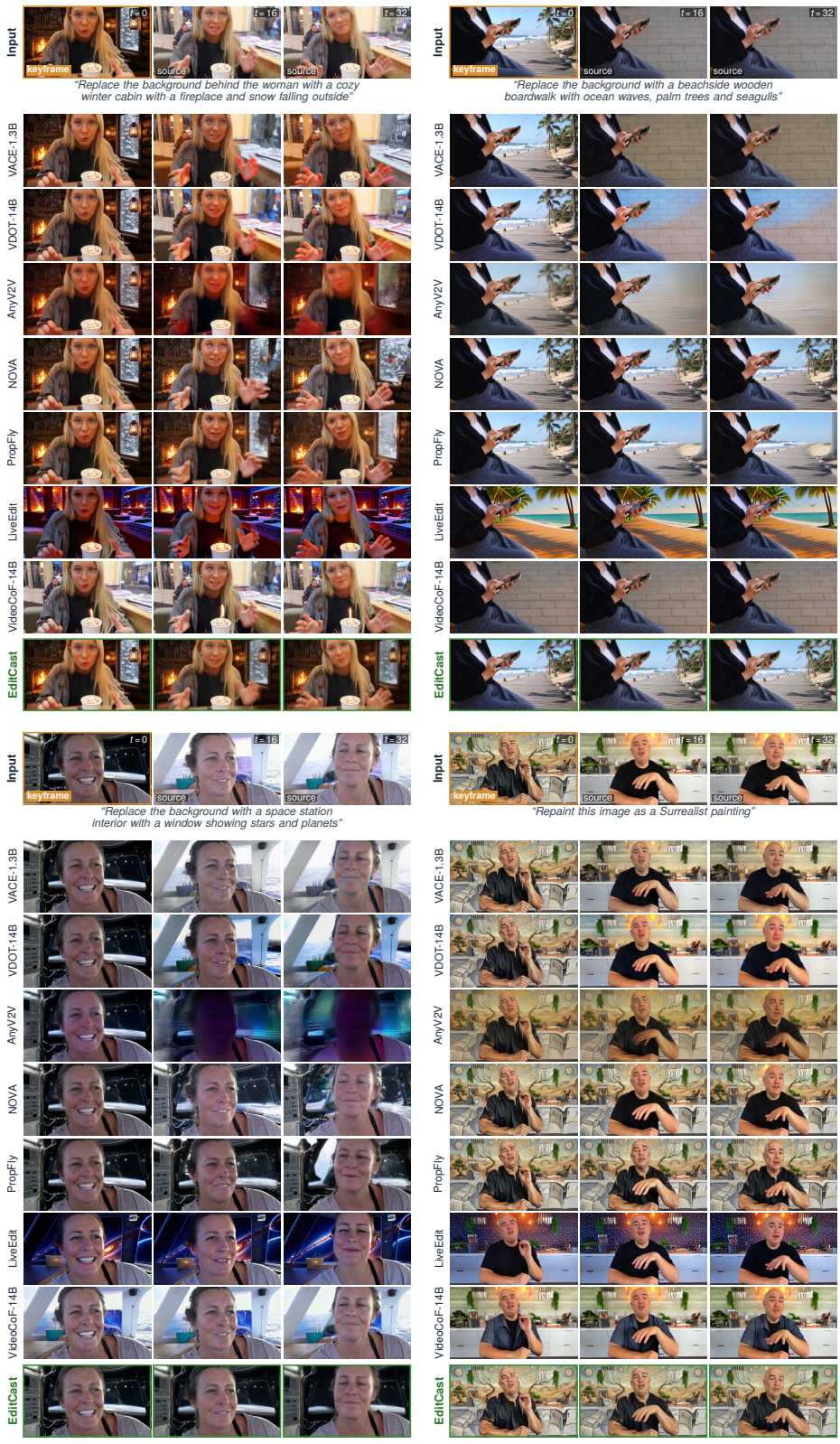

Figure 10: **Additional KeyEdit-Img results.** Examples of background replacement and style edits.

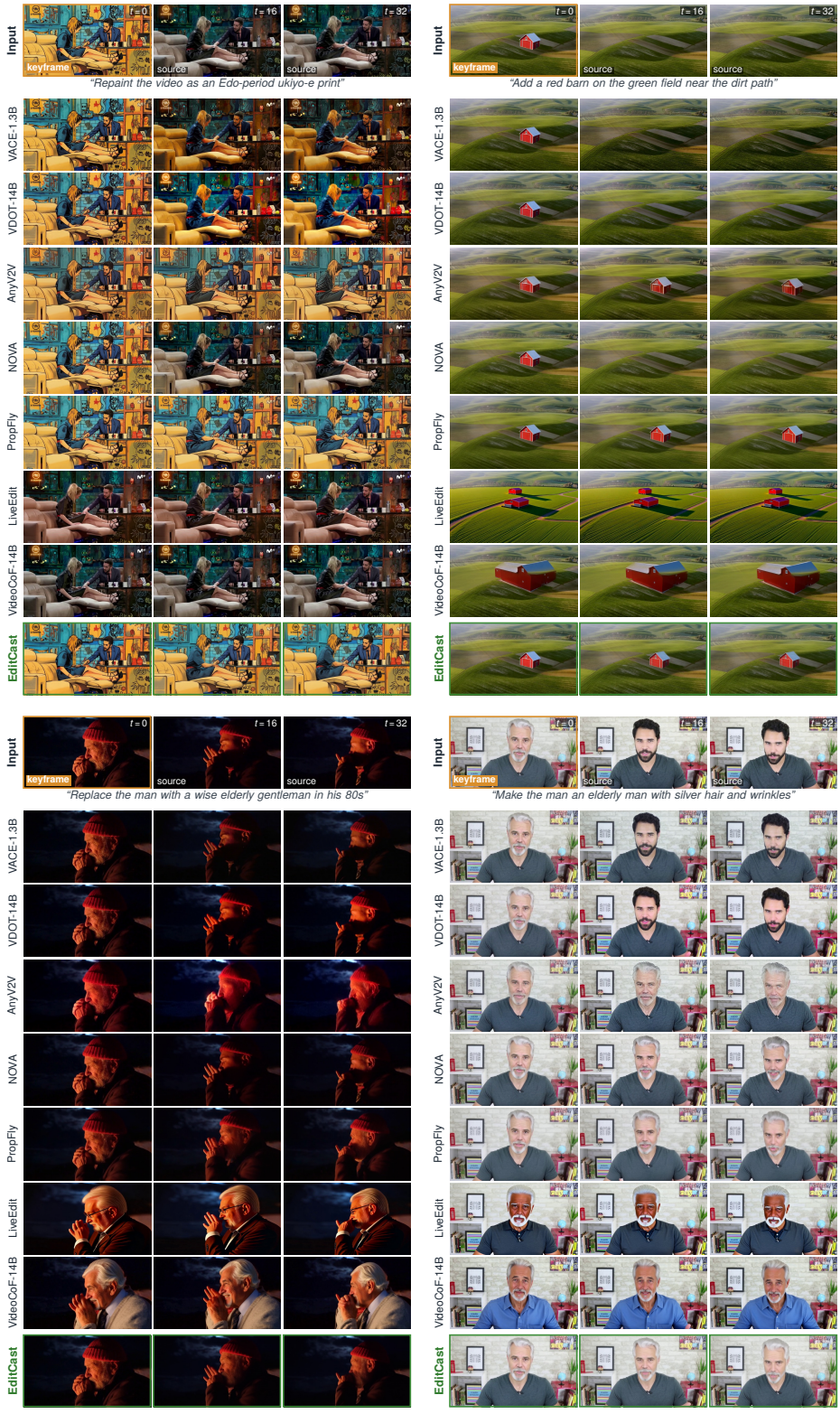

Figure 11: **Additional KeyEdit-Ref results.** Examples of style, object addition, and subject or attribute replacement; keyframes are the reference clips' first frames.

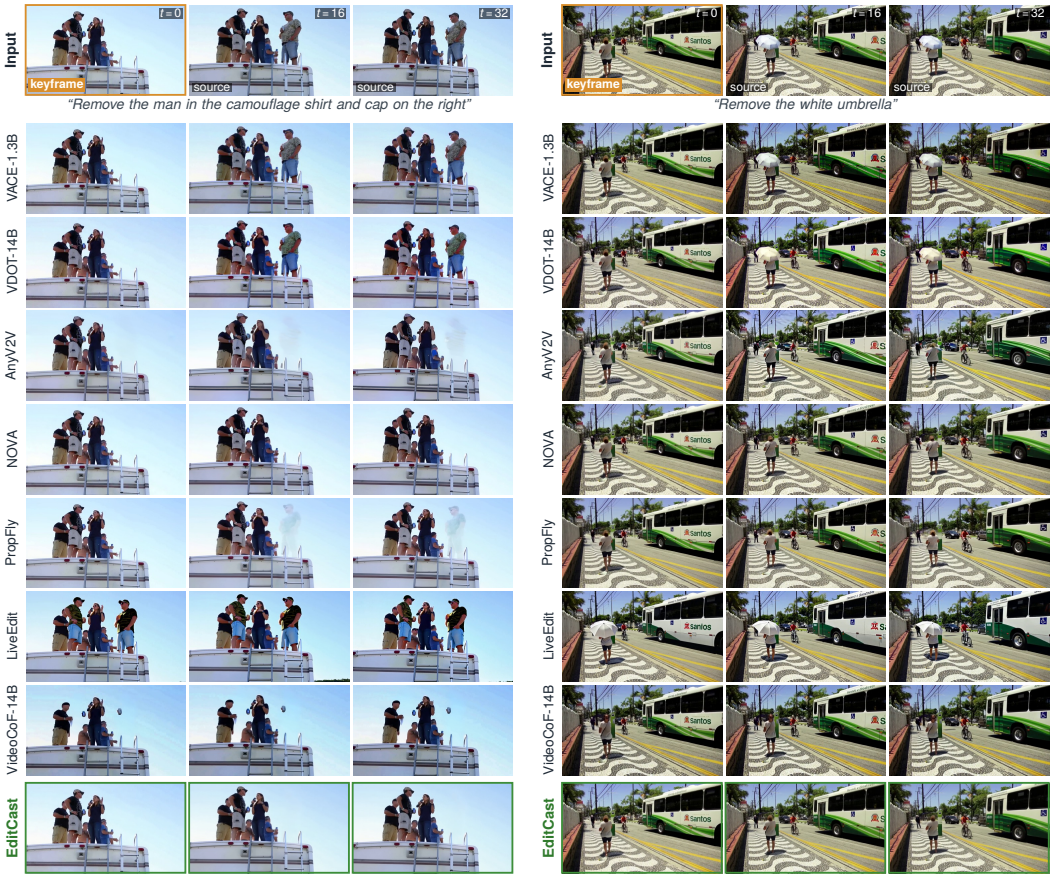

Figure 12: **Additional object-removal results.** Qualitative outputs from the compared methods.