EDITSTREAM: A UNIFIED AUTOREGRESSIVE FRAMEWORK FOR INTERACTIVE VIDEO GENERATION AND EDITING

Anonymous authors
Paper under double-blind review

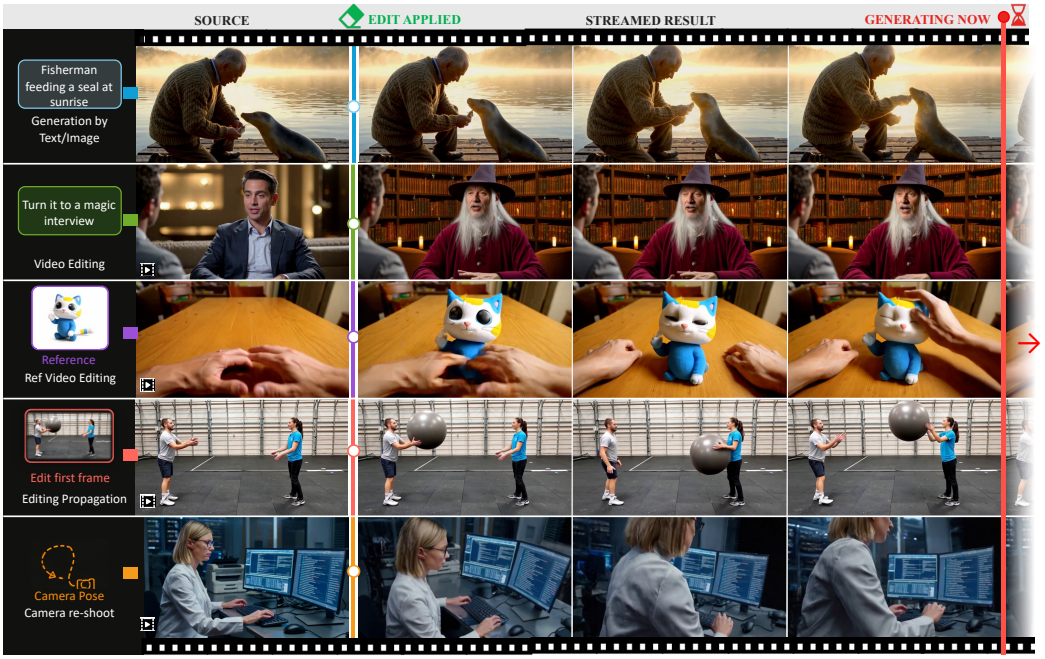


Figure 1: **Interactive video generation and editing with EditStream**. Starting from text, images, source videos, or live camera, EditStream can generate new content (T2V), do edits (V2V), introduce a subject from a reference image (RefV2V), propagate a first-frame image edit (EditProp), or reshoot a scene (ReShoot) with explicit camera control. EditStream progressively produces the edited video through few-step autoregressive inference.

ABSTRACT

Interactive video generation and editing are becoming increasingly important for creative design. In this paper, we introduce EditStream—a unified framework for interactive video generation and editing. EditStream unifies Text-to-Video, Video-to-Video, Editing Propagation, Reference-guided Video Editing, and Camera Pose Change, enabling flexible control over video generation and editing within one system. To make the unified model practical for interactive use, we develop a two-stage distillation approach that combines Velocity Moment Matching (VMM) with autoregressive unrolling. VMM matches conditional velocity moments at student-reached intermediate states to preserve generation quality and motion, while unrolling exposes the student to its own autoregressive predictions to improve temporal stability. They alleviate common challenges in few-step autoregressive video generation, including over-saturation, degraded motion, temporal instability, and complex training. EditStream provides a practical and scalable solution that bridges high-quality multi-task diffusion-based video models with interactive creative workflows.

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

Video diffusion models have improved visual fidelity, diversity, and semantic quality (Polyak et al., 2024; Wan et al., 2025; HaCohen et al., 2026; Yang et al., 2026b; Yuan et al., 2026). Users' creative workflows always need to generate videos, edit existing footage, propagate a first-frame edit, insert a reference subject, or redirect camera motion as content is produced. Supporting these operations jointly requires multiple controls, high visual and temporal quality, and high-efficient interaction experience. Unified models such as EditVerse and VACE (Ju et al., 2025; Jiang et al., 2025) already support diverse tasks, but mostly just use offline bidirectional model and are slow. Concatenating all the context tokens also increases token length and cost, especially when source videos, masks, and geometric renders are mostly pixel-aligned with the target. Autoregressive (AR) diffusion enables streaming generation (Chen et al., 2024; Yin et al., 2025; Huang et al., 2026) and editing (Li et al., 2026c; Wang et al., 2026b; Zhao et al., 2026b), but the procedure to convert a strong unified multi-task bidirectional teacher into a few-step causal student remains challenging.

Existing conversion pipelines combine offline ODE-pair preparation, initialization, teacher causalization, or consistency distillation with rollout training (Yin et al., 2025; Huang et al., 2026; Zhu et al., 2026; Zhao et al., 2026a; Zheng et al., 2026; Yuan et al., 2026). Reproducing these stages across multiple tasks is costly. Distribution-matching distillation (DMD) (Yin et al., 2024b;a) can generate sharp but less diverse outputs, and endpoint or transition regression alone does not supply a distributional correction. This motivates a simpler conversion recipe with an objective aligned to the student's few-step trajectory.

We introduce **EditStream**, a unified AR framework for T2V, V2V, reference-guided editing (RefV2V), first-frame edit propagation (EditProp), and camera pose control (ReShoot). A dual-type conditioning interface concatenates pixel-aligned conditions along channels and non-aligned reference conditions as tokens. We then retain a bidirectional teacher and distill a causal student through *Velocity Moment Matching (VMM)* warm-up followed by autoregressive unrolling. It does not require offline ODE-pair preparation or teacher causalization. Energy Annealing (EA) also mitigates over-saturation inherited from guided teacher predictions.

Our contributions are twofold. **(1) Unified conditioning:** one video DiT supports generation and editing through a shared interface and multi-task training. **(2) A practical bidirectional-to-causal conversion recipe:** VMM corrects student transitions using a teacher–auxiliary velocity residual at student-induced states, with inference-aligned timestep sampling. Warm-up provides a strong causal initialization; VMM-forcing trains on generated history. Experiments show cross-task improvements from VMM initialization and stronger T2V dynamics from VMM-forcing, while conditional editing remains competitive with Self-Forcing under the same initialization.

2 RELATED WORKS

Unified Video Generation and Editing. Generalist models unify generation and editing tasks across images and videos Geyer et al. (2024); Deng et al. (2025); Chen et al. (2025b); Jiang et al. (2025); Ju et al. (2025). Existing methods incorporate control conditions through cross-attention Yang et al. (2026a), ControlNet (Zhang et al., 2023; Zhou et al., 2026; Liu et al., 2025b), channel concatenation (Chen et al., 2026a; Wu et al., 2026), or token concatenation Ju et al. (2025); Liu et al. (2026); Cai et al. (2026). Others combine multimodal understanding with generation to improve semantic control (Team et al., 2025; Chen et al., 2026b; Yang et al., 2026a; Team et al., 2026; Wei et al., 2025). EditStream prioritizes inference efficiency within this unified setting. It uses channel concatenation for pixel-aligned inputs, such as source videos. It utilizes token concatenation for non-aligned inputs, such as reference images. Our dual-type scheme supports diverse tasks while reducing token overhead.

Streaming Autoregressive Video Generation and Editing. Joint denoising provides full spatiotemporal context but becomes expensive for long or high-resolution videos. Rolling Diffusion and FIFO generate videos progressively Ruhe et al. (2024); Kim et al. (2024b), although frames within a window can still interact bidirectionally. Strictly causal models attend only to generated history, enabling key–value caching during streaming inference. Diffusion Forcing introduced per-frame noise levels (Chen et al., 2024), while teacher forcing conditions generation on clean history Williams & Zipser (1989); Huang et al. (2026); Zhu et al. (2026). CausVid combined these

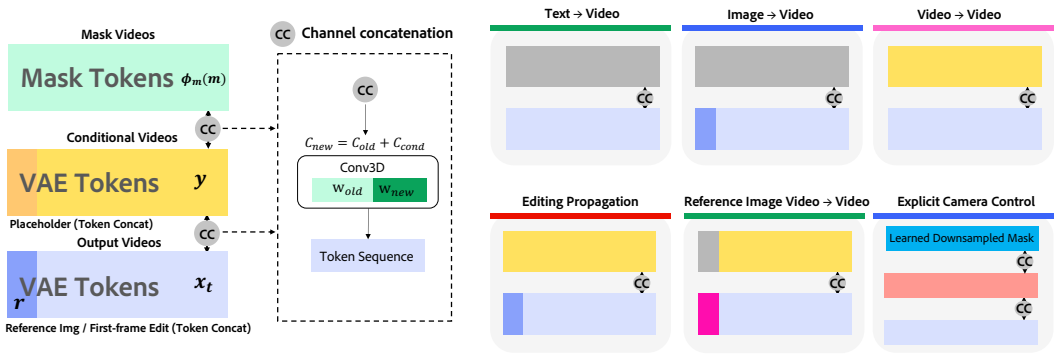


Figure 2: **Dual-type conditioning.** Pixel-aligned source and mask latents use channel concatenation; reference images and edited first frames use token concatenation, shown as the small darker block appended to the left end of the VAE token bar and denoted $\mathbf{r}$.

ideas with distribution-matching distillation (DMD) to convert a bidirectional video model into a causal generator (Yin et al., 2025; 2024b;a).

A key challenge is exposure bias: training often uses clean history, whereas inference relies on the model's own outputs. Self-Forcing addresses this gap through student unrolling and DMD (Huang et al., 2026). Causal Forcing, Causal Forcing++, and Causal-rCM study teacher causality and initialization (Zhu et al., 2026; Zhao et al., 2026a; Zheng et al., 2026); One-Forcing and AAD-1 extend self-rollout distillation to one-step generation (Feng et al., 2026; Li et al., 2026a). For longer horizons, methods including Rolling Forcing, Context Forcing, Long Live, Self-Forcing++, TetherCache, DySink, and Rolling Sink address error accumulation and context management through memory, positional encoding, and cache strategies (Liu et al., 2025a; Chen et al., 2026c; Yang et al., 2026b; Chen et al., 2026d; Hu et al., 2026; Cui et al., 2025; Meng et al., 2026; Ye et al., 2026; Li et al., 2026b).

Streaming editing has progressed from image diffusion models with temporal modules (Xing et al., 2024; Liang et al., 2025; Chen et al., 2025a) toward causal video editors, including RFDM, SANA-Streaming, LiveEdit, EgoEdit, and adaptations of VACE (Salehi et al., 2026; Zhao et al., 2026b; Wang et al., 2026b; Li et al., 2026c; Jiang et al., 2025; Fosdick, 2026). EditStream advances this direction by unifying diverse generation and editing tasks and studying bidirectional-to-causal conversion across them. Its efficient conversion and distillation recipe is also complementary to long-horizon memory and cache-management methods.

3 Method

3.1 Unified Generation and Editing

Let $\mathbf{x}_0 \in \mathbb{R}^{T\times C\times H\times W}$ be a clean VAE video latent, where T is the latent frames, C is the channels, and H, W represent spatial dimensions. Given task conditions $\mathcal{C}$, the flow-matching model v_θ, parameterized by θ, predicts velocity from the noisy latent $\mathbf{x}_t$ and timestep t, $v_\theta(\mathbf{x}_t, t, \mathcal{C})$. Tasks only differs in $\mathcal{C}$, so their data can be mixed during training.

T2V uses text, I2V adds first frame, and V2V inputs a source video. EditProp and refV2V additionally use an edited first frame or single reference image. ReShoot uses a source video reprojected to the target camera pose and a binary validity mask to distinguish missing geometry from dark pixels. To disambiguate tasks, we prepend the prompt with prefixes `[Editing Propagation]`, `[Reference Guidance]`, and `[Camera Control]` to identify EditProp, refV2V, and ReShoot, respectively.

Pixel-aligned conditions share the target's spatiotemporal grid, including source videos and camera-warped inputs. Channel concatenation reduces long context token length to improve efficiency. Non-aligned reference images and edited first frames instead use token concatenation for in-context

learning (Figure2):

$$\widetilde{\mathbf{x}}_t = \mathrm{Concat}_{\mathrm{ch}}[\mathbf{x}_t, \mathbf{y}, \phi_m(\mathbf{m})], \qquad \mathbf{x}_t^* = \mathrm{Concat}_{\mathrm{tk}}[\mathbf{r}, \mathbf{x}_t[:-1]], \tag{1}$$

where $\mathbf{y}$ is the conditional video latent, $\mathbf{m}$ the original-resolution mask, ϕ_m its learned downsampler to latent resolution, and $\mathbf{r}$ the reference/first-frame latent. The token construction prepends $\mathbf{r}$ to the target sequence with its final temporal slot omitted. The expanded patch embedding copies pretrained weights and zero-initializes additional channels; absent channel conditions are also zeroed.

3.2 Velocity Moment Matching

Unlike DMD (Yin et al., 2024b), which re-noises a complete student output and matches marginal distributions, Moment Matching (MM) (Salimans et al., 2024) matches conditional moments at a noisy state reached by a coarse student transition. VMM applies this principle in velocity space, the native parameterization of our flow-matching model. Under an invertible linear parameterization, velocity- and clean-sample-moment matching are theoretically equivalent, but the latter introduces gradient scaling dependent on both transition times t, s. Direct velocity matching removes this implicit scaling and supervises the local evolution direction; together with inference-aligned sampling, it improves convergence.

VMM objective. We use a frozen full-attention **teacher** T as the reference, a trainable **auxiliary model** A to estimate the velocity field of the student's distribution and stabilize matching targets, and a **student** S. The student shares the teacher architecture for bidirectional inference and uses block-causal attention for streaming. Their velocity predictions are v_T, v_A, and v_S.

Given noisy latent $\mathbf{x}_t$ at timestep t and task conditions $\mathcal{C}$, the student predicts a velocity and takes an ODE step to $s < t$:

$$v_t^S = v_S(\mathbf{x}_t, t, \mathcal{C}), \qquad \mathbf{x}_s = \mathbf{x}_t + (\sigma_s - \sigma_t) v_t^S. \tag{2}$$

Let $\bar{v}_S^{t\to s}$ denote the student velocity associated with this transition; for deterministic ODE updates, $\bar{v}_S^{t\to s} = v_t^S$. Teacher and auxiliary evaluate the same student-generated state $(\mathbf{x}_s, s)$.

The auxiliary model tracks the detached student velocity on the student's current distribution:

$$\mathcal{L}_A = \mathbb{E}_{t,s}\big[\|v_A(\mathbf{x}_s, s, \mathcal{C}) - \mathrm{sg}(\bar{v}_S^{t\to s})\|_2^2\big]. \tag{3}$$

Here sg stops gradients; student states and velocities are fixed when updating A. The student target and loss are

$$\begin{aligned} \widetilde{v} &= \mathrm{sg}\big(\bar{v}_S^{t\to s} + v_T(\mathbf{x}_s, s, \mathcal{C}) - v_A(\mathbf{x}_s, s, \mathcal{C})\big), \\ \mathcal{L}_S &= \mathbb{E}_{t,s}\big[\|\bar{v}_S^{t\to s} - \widetilde{v}\|_2^2\big]. \end{aligned} \tag{4}$$

The residual $v_T - v_A$ acts as a real–fake score difference expressed in velocity space along the student's ODE trajectory. The target is detached, so student gradients pass only through $\bar{v}_S^{t\to s}$.

To align training with few-step inference, we use a J-step schedule $\tau_0 > \cdots > \tau_J = 0$ ($J = 4$): sample i uniformly from $\{0, \ldots, J-1\}$, set $t = \tau_i$, and sample $s \sim \mathcal{U}(\tau_{i+1}, \tau_i)$. Pseudocode is in the Appendix.

Energy Annealing (EA). Strong classifier-free guidance (CFG) can cause teacher over-saturation that transfers to the student. EA operates in clean-sample space, where color statistics are more meaningful. Let c, u denote conditional and unconditional teacher predictions, and w the CFG scale. At query time s, define $\widehat{x}_0^{c,u} = x_s - \sigma_s v_T^{c,u}$ and $\Delta x = \widehat{x}_0^c - \widehat{x}_0^u$. The full-guidance prediction is $g = \widehat{x}_0^c + (w-1)\Delta x$; the reference is $r = \widehat{x}_0^c + \eta_{\mathrm{EA}}(w-1)\Delta x$, where $0 < \eta_{\mathrm{EA}} < 1$ attenuates guidance. We match per-channel means μ and standard deviations Std:

$$\widehat{x}_0^{\mathrm{EA}} = \frac{\mathrm{Std}(r)}{\mathrm{Std}(g) + \epsilon}\big(g - \mu(g)\big) + \mu(r), \qquad v_T^{\mathrm{EA}} = \frac{x_s - \widehat{x}_0^{\mathrm{EA}}}{\sigma_s}. \tag{5}$$

Here $\epsilon > 0$ stabilizes the denominator. Statistics pool over temporal and spatial axes, applying one correction uniformly across each channel's frames and pixels. Using v_T^{EA} as the teacher velocity in Equation 4 retains full-guidance structure while controlling color and saturation against the weaker reference; the same correction applies during teacher inference. We use $\eta_{\mathrm{EA}} \approx 0.7$; smaller values can over-suppress guidance and require model-specific tuning.

3.3 BIDIRECTIONAL-TO-CAUSAL DISTILLATION

For K latent video blocks indexed by $k = 0, \ldots, K-1$, let $\mathbf{x}^k$ denote block k and $\mathcal{C}^k$ its task conditions (text and visual controls from Section 3.1). The model distribution p_θ factorizes as

$$p_\theta(\mathbf{x}^{0:K-1} \mid \mathcal{C}^{0:K-1}) = \prod_{k=0}^{K-1} p_\theta(\mathbf{x}^k \mid \mathbf{x}^{<k}, \mathcal{C}^{\leq k}). \tag{6}$$

Causal attention restricts the student to past and current blocks, supporting key–value (KV) caching and generation beyond a fixed clip. We retain the bidirectional teacher and use the same VMM objective to train a causal student: first with ground-truth context, then with generated history. No offline ODE pairs are required.

Teacher adaptation. To prepare the teacher for block-wise distillation, we augment standard full-sequence diffusion training with first-block cold-start and prefix–suffix teacher-forcing modes. With $B = 3$ latent frames per block, training mixes: (i) first-block cold start with probability p_{first}, truncating every tensor to the first block; (ii) prefix–suffix teacher forcing with probability p_{ps}, noising one uniformly selected block, keeping the others clean at timestep zero, and supervising only the selected block; and (iii) standard full-sequence training with the remaining probability to preserve full-video modeling. Unlike causal-teacher approaches (Zhu et al., 2026; Zhao et al., 2026a; Zheng et al., 2026), this changes noising and loss masks while retaining bidirectional attention. We initialize S and A from the adapted teacher, then freeze T; S uses block-causal attention and A remains bidirectional.

Stage 1 of student distillation: VMM warm-up initialization. We first train the student to denoise blocks in a few steps, using ground-truth context when available. First-block updates retain only block 0 for all three models, teaching generation without video history. Prefix–suffix updates denoise a selected block: the student sees its clean ground-truth prefix, while T/A also see the clean suffix. The suffix supplies additional supervision, not student input.

For either update, let binary loss mask M select the active block. At $\ell \in \{t, s\}$, we construct

$$\mathbf{z}_\ell = (1 - M) \odot \mathbf{x}_0 + M \odot \mathbf{x}_\ell, \qquad \boldsymbol{\tau}_\ell = \ell M. \tag{7}$$

Here $\mathbf{z}_\ell$ is the query video, $\boldsymbol{\tau}_\ell$ its per-block timesteps (zero outside M), and $\odot$ elementwise multiplication. Equation 2 gives the student transition. In Equations 3 and 4, T/A evaluate $(\mathbf{z}_s, \boldsymbol{\tau}_s, \mathcal{C})$, and each loss residual is multiplied by M before squaring. Warm-up stabilizes block generation and reduces boundary jitter, but its ground-truth history leaves a gap to inference.

Stage 2 of student distillation: VMM-forcing unrolling. We now replace ground-truth history with student-generated history and unroll the inference procedure (Figure 3). Each block starts from Gaussian noise, completes all J denoising steps, and supplies its detached clean output to the cache for subsequent blocks. Writing $\mathcal{H}_S^{<k}$ for the KV cache of previously completed blocks,

$$\begin{aligned} \mathbf{x}_{\tau_0}^k &\sim \mathcal{N}(0, I), \quad \mathbf{x}_{\tau_{j+1}}^k = \mathrm{Step}_S(\mathbf{x}_{\tau_j}^k, \tau_j, \tau_{j+1}; \mathcal{H}_S^{<k}), \\ \mathcal{H}_S^{\leq k} &= \mathrm{Append}\big(\mathcal{H}_S^{<k}, \mathrm{KV}_S\big(\mathrm{sg}(\mathbf{x}_{\tau_J}^k), 0\big)\big). \end{aligned} \tag{8}$$

Here $j = 0, \ldots, J-1$; Step_S performs one deployed student sampler step, with task conditions implicit. $\mathrm{KV}_S(\cdot, 0)$ computes student keys and values at timestep zero, and Append adds them to the cache.

For supervision, we sample a local transition along this trajectory and construct $\mathbf{x}_s^k$ using Equation 2, conditioned on $\mathcal{H}_S^{<k}$. Teacher and auxiliary evaluate the same assembled query state to provide the VMM correction. This query does not replace the cached output: the block still runs to $\tau_J = 0$ before being committed. Like Self-Forcing (Huang et al., 2026), we train on generated history; the difference is that its training can cache a clean prediction from a randomly selected early exit, whereas we always complete the deployed schedule.

To preserve cold-start quality while adapting later blocks to generated history, we interleave Stage 1's non-unrolled first-block VMM updates with full-unrolling updates. Block 0 has no autoregressive history and remains directly supervised, making it a relatively reliable shared anchor. During unrolling, we therefore also condition T/A on the student's completed block 0, aligning the initial scene and appearance among the three models. The unrolling loss mask M excludes this conditioning block and selects later blocks. Both updates train the same student using the VMM losses in Equations 3 and 4 on their selected blocks.

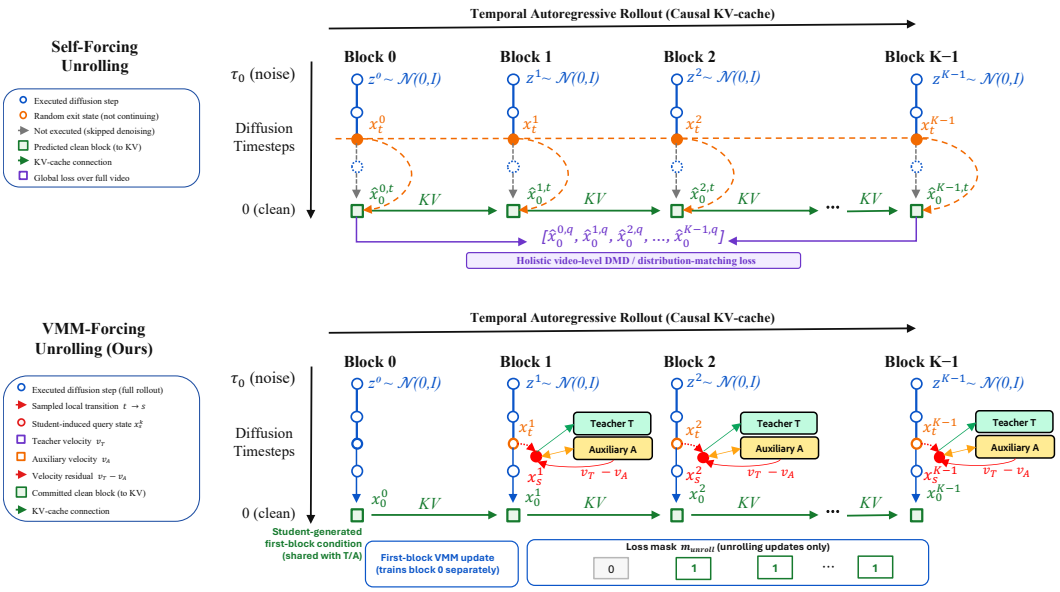


Figure 3: **Self-Forcing (top) versus VMM-Forcing (bottom)**. Self-Forcing commits clean predictions from randomly truncated timestep, so its rollout does not exactly match the deployed inference schedule, whereas VMM-Forcing completes the inference schedule and applies local teacher–auxiliary velocity corrections, exactly matching the deployed rollout. Its student-generated first block conditions T/A and the training alternates first-block supervision and unrolling.

4 Experiments

Training Data Overview EditStream uses a mixture of generation and editing data, with source-level scales and sampling ratios in Appendix A.2 (Table 6). LTX2.3-video, generated with LTX2.3 (HaCohen et al., 2026), and OpenVid-1M (Nan et al., 2025) provide text-/image-conditioned generation samples to preserve the general video prior. Self-collected source–target video pairs support instruction-guided editing and propagation; Pico (Qian et al., 2026) image-editing pairs are converted into short videos through smooth translation augmentation. KiwiEdit (Lin et al., 2026b) supplies source–target–reference triplets for subject or background transfer.

For interaction-aware object addition, removal, and propagation, we render HUMOTO (Lu et al., 2025) motions as synchronized object-present/object-absent pairs, inspired by VOID (Motamed et al., 2026). Kubric-style (Greff et al., 2022) synthetic clips provide additional object addition/removal supervision. EgoEdit (Li et al., 2026c) covers egocentric object replacement, and EffectErase (Fu et al., 2026) provides pairs for removing visual effects and their induced artifacts. MultiCamVideo (Bai et al., 2025a) supplies synchronized source–target views for camera control, with target-view reprojections and validity masks constructed from VGGT-Omega (Wang et al., 2026a) reconstructions. Videos are sampled as 81-frame clips at 720p; complete construction, annotation, and filtering procedures are in Appendix A.2.

Setup. We fine-tune Wan2.2-TI2V-5B (Wan et al., 2025) and initialize student/auxiliary from the adapted teacher EMA. Student training uses 200 warm-up and 800 unrolling iterations, a four-step schedule $\{999, 938, 833, 625\}$ and three latent frames per block. Teacher and student training each use 8×8 A100-80GB GPUs. The ODE-initialized Self-Forcing baseline uses the same backbone and tasks, but different initialization and critic-update settings; full hyperparameters are in Appendix A.3.

T2V quality and speed. Table 1 shows the results of T2V task. The teacher finetuning improves total VBench from 83.60 to 84.05 over the original WAN model. VMM initialization followed by Self-Forcing reaches 84.13; VMM-forcing reaches 84.42 with dynamic degree 79.17 versus 66.67 under the same initialization. Color and quality also improve, while semantic and multiple-object

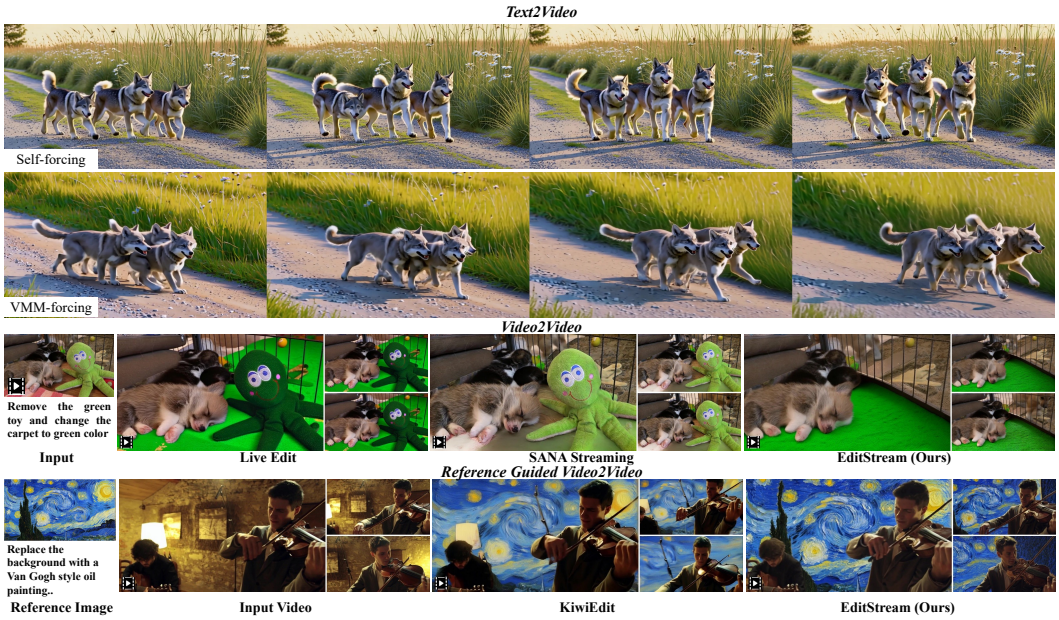


Figure 4: **Representative generation and editing comparisons**. T2V illustrates stronger motion and dynamics; V2V and RefV2V illustrate instruction and reference following and better quality. The complete multi-task galleries, including camera control, further reference editing, egocentric editing and propagation, are in Appendix A.8.

metrics do not improve together showing these trade-offs. Qualitative results show richer motion and scene evolution (Figure 4); the detailed motion/object-detection analysis is in Appendix A.5.

At 720p on one H100, the student has 0.90 s time-to-first-output and 10.56 FPS end-to-end throughput. Parallel VAE decoding and DiT denoising on two GPUs reaches 16.10 FPS. This enables progressive low-latency output but does not yet reach real-time playback at this resolution. Additional decoder distillation, quantization and kernel optimization are orthogonal.

Table 1: **Text-to-Video** Results: VBench text-to-video benchmark. Latency and throughput are both end-to-end performance. Time-to-first-output denotes the wall-clock time until the first decoded RGB block becomes available at batch size 1. Measurements are conducted in one single H100.

Model	#Params	Resolution	Latency: time-to-first-output (s)	Steady-state Throughput (FPS)	Dynamic Degree ↑	Color ↑	Human Action ↑	Multiple Objects ↑	Quality Score ↑	Semantic Score ↑	Total Score ↑
Offline Models											
Wan2.2 5B TI2V Base Wan et al. (2025)	5B	1280×704	72.30	1.12	63.61	**89.52**	97.20	**84.88**	84.04	**81.86**	83.60
EditStream Teacher FT (ema)	5B	1280×704	72.30	1.12	**67.78**	86.15	**97.80**	84.15	**84.73**	81.31	**84.05**
Streaming Models											
EditStream-AR (ODE Init. + Self-Forcing Unrolling Huang et al. (2026))	5B	1280×704	0.90	10.56	48.06	82.72	96.80	**89.54**	84.09	**81.95**	83.67
EditStream-AR (VMM Init. + Self-Forcing Unrolling)	5B	1280×704	0.90	10.56	66.67	84.45	97.40	84.91	85.03	80.52	84.13
EditStream-AR (VMM Init. + VMM-Forcing Unrolling)	5B	1280×704	0.90	10.56	**79.17**	**87.82**	**98.00**	83.26	**85.36**	80.64	**84.42**

V2V and reference-guided editing. On 110 pixel-aligned EditVerse cases (Ju et al., 2025), the teacher provides strong temporal consistency and competitive alignment (Table 2). VMM-forcing has the highest VLM video-quality score among the evaluated streaming variants (7.35), versus 7.01 for ODE+SF and 7.25 for VMM+SF. Commercial and offline models can score higher. On RefVIE-Bench (Table 3a), the teacher scores 3.53 overall, the highest among the evaluated offline open-source methods; streaming VMM+SF and VMM+VMM reach 3.72 and 3.71. Relative to Kiwi-Edit (Lin et al., 2026b), which supplies the same reference-editing training data, examples show faithful subject identity, scale, composition, and texture. These comparisons establish system-level effectiveness, not isolated causality for a specific architectural or distillation choice.

Camera control (ReShoot). Our V2VCameraPose benchmark applies four trajectories (loop, orbit, pan, zoom-in) to 118 clips: 78 DAVIS videos and 20 each from Kling v2.1 and Sora 2, yielding 472 cases. Reprojection provides warped videos and visibility masks; stored camera matrices are

Table 2: **Video-to-Video Editing**: Comparison on EditVerse benchmark for video-to-video editing. Higher values indicate better performance for all metrics. We follow EditVerse to use GPT-4o as the VLM judge model.

Model	CLIP Temp. ↑	DINO Temp. ↑	Frame-Text Align. ↑	Video-Text Align. ↑	Pick Score ↑	Video Quality ↑
Commercial Models						
Runway Aleph	0.9888	0.9849	**28.30**	25.33	20.36	7.64
Kling-O1	**0.9926**	**0.9911**	28.24	**25.72**	**20.59**	**7.93**
Offline Models						
InsV2V Cheng et al. (2024)	0.9720	0.9679	25.84	23.07	19.55	5.07
LucyEditDev DecartAI Team (2025)	0.9852	0.9839	26.30	23.51	19.54	5.87
STDF Yatim et al. (2024)	0.9635	0.9612	26.18	23.59	19.74	4.55
Senorita-2M Zi et al. (2026)	0.9804	0.9810	27.24	24.64	19.75	6.83
TokenFlow Geyer et al. (2024)	0.9867	0.9880	26.63	24.26	20.01	5.34
VACE Jiang et al. (2025)	0.9896	0.9884	27.07	24.40	20.06	6.33
EditVerse Ju et al. (2025)	0.9859	0.9859	**27.89**	**25.66**	**20.12**	**7.67**
EditStream Teacher	**0.9917**	**0.9901**	27.49	25.31	**20.12**	7.52
Streaming Models						
SANA Streaming Zhao et al. (2026b)	0.9889	0.9869	26.91	24.55	19.80	5.95
LiveEdit Wang et al. (2026b)	0.9899	0.9896	25.91	23.28	19.45	6.52
EditStream-AR (ODE Init. + Self-Forcing Unrolling Huang et al. (2026))	0.9903	0.9892	26.81	24.68	19.98	7.01
EditStream-AR (VMM Init. + Self-Forcing Unrolling)	**0.9914**	**0.9897**	27.03	**25.22**	**20.16**	7.25
EditStream-AR (VMM Init. + VMM-Forcing Unrolling)	0.9911	0.9893	**27.05**	24.95	20.01	**7.35**

Table 3: **Reference-guided editing and camera pose control. Models.** Teacher: EditStream Teacher. O+S/V+S/V+V: ODE+Self-Forcing/VMM+Self-Forcing/VMM+VMM-Forcing; Self-Forcing follows Huang et al. (2026). Kiwi-I+R/Kiwi-R: Kiwi-Edit with instruction–reference/reference only. ReCam/ViewCr./TrajCr.: ReCamMaster/ViewCrafter/TrajCrafter. **Metrics.** (a) ID/TC/PC: identity/temporal/physical consistency; RS/MQ/VQ: reference similarity/matting quality/video quality; Avg.: subset average; All: overall. (b) SC/BC/TF/MS: subject/background consistency, temporal flickering score, and motion smoothness (video-only VBench dimensions); SSIM: valid-region SSIM; HC/HTS/GHF: hole completeness, hole temporal stability, and gated hole-fill quality. Holes are disoccluded regions under target-camera reprojection; SSIM measures preservation of observed regions. **Baselines.** UniVideo (Wei et al., 2025), ReCo-Ref (Zhang et al., 2025), ReCamMaster (Bai et al., 2025a), ViewCrafter (Yu et al., 2024), TrajCrafter (Yu et al., 2025), Vista4D (Lin et al., 2026a). Bold/underline mark the best/second-best results within each model group.

(a) **Reference-guided editing.** RefVIE-Bench; Gemini-3-Flash judge. All metrics ↑.

Model	Subject ref.				Background ref.				All
	ID	TC	PC	Avg.	RS	MQ	VQ	Avg.	
Offline									
UniVideo	4.17	3.79	3.59	3.85	3.13	2.55	2.28	2.65	3.44
ReCo-Ref	3.65	2.95	2.70	3.10	3.35	**2.65**	2.38	**2.79**	3.00
Kiwi-I+R	3.51	2.96	2.91	3.13	**3.40**	2.58	**2.40**	**2.79**	2.96
Kiwi-R	**4.28**	3.61	3.55	3.81	3.33	2.35	2.08	2.58	3.40
Teacher	4.13	**3.96**	**3.94**	**4.01**	3.20	2.43	2.10	2.58	**3.53**
Streaming (EditStream-AR)									
O+S	3.49	3.30	3.21	3.33	3.38	**2.78**	2.43	2.86	3.18
V+S	**4.50**	**4.10**	**3.90**	**4.17**	3.35	2.68	**2.48**	2.83	**3.72**
V+V	4.41	4.03	**3.90**	4.11	**3.45**	**2.78**	**2.48**	**2.90**	3.71

(b) **Camera pose control.** V2VCameraPose; 472 samples. All metrics ↑.

Model	Video quality				Reprojection quality			
	SC	BC	TF	MS	SSIM	HC	HTS	GHF
Offline								
ReCam	**89.23**	91.18	**96.44**	**98.65**	0.388	**96.96**	82.07	22.96
ViewCr.	86.86	91.31	93.26	96.12	0.407	96.37	86.43	27.73
TrajCr.	88.19	92.13	93.49	97.00	0.661	94.89	**87.54**	39.06
Vista4D	88.81	92.28	93.22	96.77	**0.751**	96.28	86.26	**51.25**
Teacher	88.52	**92.94**	94.81	98.07	0.665	96.74	84.11	38.84
Streaming (EditStream-AR)								
O+S	88.41	92.87	**94.74**	97.58	**0.657**	89.26	81.41	30.96
V+S	88.23	92.62	94.67	97.50	0.633	93.16	81.68	33.00
V+V	**88.55**	**92.90**	94.70	**97.95**	0.622	**97.48**	**84.03**	**36.27**

not direct inference inputs. Table 3b shows strong content consistency and competitive temporal quality. VMM-forcing improves hole completion and stability among streaming variants, while Self-Forcing retains higher valid-region SSIM. Vista4D and TrajCrafter remain strong offline baselines (Lin et al., 2026a; Yu et al., 2025). Full construction, all metrics, and visual comparisons are retained in Appendices A.4.1 and A.8.

Egocentric editing and propagation. EgoEditBench examples show editing quality broadly comparable to EgoEdit, with differences in texture sharpness and realism; differing data and distillation makes the comparison less conclusive. For first-frame propagation, examples on the small GenProp benchmark cover difficult background changes and outpainting, despite no dedicated outpainting training (Appendix A.8). These evaluations are qualitative and we use EgoEdit/GenProp official project results for comparison due to unavailable open-sourced implementations. Appendix A.8 shows more details of the comparisons with SANA, AnyV2V and Señorita-2M.

5 Ablation Studies

Table 4: **Ablation study** on the proposed training strategy. Higher is better for all metrics.

	Configuration						Results				
	Warm-up			Unrolling			T2V			V2V	RefV2V
ID	Method	Iter.	W/U Ratio	Method	Alt.	Commit	Quality	Semantic	Overall	VLM	Overall
B1	VMM Init.	200	1:4	VMM-forcing Unrolling	✓	✓	**85.36**	80.64	**84.42**	**7.35**	3.71
B2	ODE Init.	5K	5:1	Self-forcing Unrolling	N/A	N/A	84.09	**81.95**	83.67	7.01	3.18
E1	ODE Init.	5K	1:0	N/A	N/A	N/A	81.72	74.33	80.24	6.98	2.68
E2	VMM Init.	200	1:0	N/A	N/A	N/A	84.17	79.89	83.31	7.13	3.55
E3	No Warmup	0	0	VMM-forcing Unrolling	N/A	N/A	83.80	78.52	82.74	7.22	2.87
E4	VMM Init.	200	1:4	Self-forcing Unrolling	N/A	N/A	<u>85.03</u>	80.52	<u>84.13</u>	<u>7.25</u>	<u>3.72</u>
E5	VMM Init.	500	1:1	VMM-forcing Unrolling	✓	✓	84.71	<u>80.85</u>	83.94	7.24	3.67
E6	VMM Init.	200	1:4	VMM-forcing Unrolling	✓	✗	84.91	80.70	84.07	7.23	3.68
E7	VMM Init.	200	1:4	VMM-forcing Unrolling	✗	✗	84.73	80.43	83.87	7.27	**3.73**

Warm-up initialization. E1/E2 compare initialization without unrolling: 200 VMM iterations yield 83.31/7.13/3.55 on T2V/V2V/RefV2V, versus 80.24/6.98/2.68 after 5K ODE iterations (Table 4). These are iteration counts, not an equal-compute speedup. Holding Self-Forcing fixed (B2/E4), VMM initialization improves all three scores to 84.13/7.25/3.72. Removing warm-up (E3/B1) lowers all three scores and introduces block-boundary discontinuities in the visual comparisons (Appendix A.6). Thus a strong causal initialization is central to the recipe.

Unrolling objective. With VMM initialization fixed (E4/B1), VMM-forcing improves T2V from 84.13 to 84.42, particularly motion dynamics (Table 1). Conditional editing remains comparable: V2V rises from 7.25 to 7.35 while RefV2V changes from 3.72 to 3.71. The evidence supports improved T2V dynamics, not uniform superiority across tasks.

Training schedule and committed context. Under a fixed 1K-iteration budget, extending warm-up from 200 to 500 iterations raises T2V semantics from 80.64 to 80.85 but lowers the total score from 84.42 to 83.94. The original 1:4 allocation better balances initialization and generated-history training. Removing committed-context conditioning together with cold-start masking (E6) reduces T2V to 84.07; additionally removing alternating first-block updates (E7) gives 83.87. Editing changes are smaller, and RefV2V can improve slightly. The clearer benefit of shared context is lower, more stable unrolling loss; alternating first-block updates alone do not visibly recover that behavior. These ablations support optimization stability without implying uniform downstream gains.

Energy Annealing and HOI data. Across 946 VBench generations, no-EA maximizes color richness but also produces the most shadow/highlight clipping and color flicker. EA=0.2 minimizes these artifacts but produces flatter colors; EA=0.7 balances the two. HOI supervision improves correspondence between hands and edited objects: examples include grasping a virtual pot lid and placing a generated hat on a head, including stable object placement after release. Without it, objects can disappear, morph, interpenetrate or remain implausibly attached. Appendix A.6 shows more details.

6 Limitations and Conclusion

EditStream unifies diverse video generation and editing tasks through dual-type conditioning and converts a bidirectional unified DiT into a causal streaming model using simple yet effective Velocity Moment Matching (VMM) distillation. Across text-to-video generation, video editing, editing propagation, and camera pose control, it achieves strong generation quality with efficient inference. VMM warm-up consistently improves upon ODE initialization for Self-Forcing; with the same initialization, VMM-Forcing and Self-Forcing perform comparably on conditional editing, while VMM-Forcing better preserves motion and temporal diversity in text-to-video generation. The current model is primarily designed for short-form videos, without specialized long-memory modeling or mitigation of error accumulation, and its 720p inference is not yet real-time. Future work will address long-form generation and editing alongside more efficient high-resolution inference.

AI USE STATEMENT

We used generative AI tools to assist with refining proofs for our distillation theory, comparing differences between methods, and designing evaluation metrics. The theoretical models, conceptual framework, and initial proofs were developed and written by the authors. The first author checked the mathematical derivations in the AI-assisted proofs. The research framework and experimental design were developed by the authors, with AI assistance in evaluation metric design as noted above. The analysis followed author-developed reasoning and used established benchmark metrics where applicable. The authors conducted the experiments, interpreted the results, and prepared their final presentation. AI models like LTX2.3 are used to generate synthetic video data in the training pipeline. All details of applying generative AI for data generation and cleaning can be found in the Appendix.

For coding, we primarily used Claude Opus to assist with implementation, optimization, and debugging of portions of the code. The core methods and original code were implemented by the authors. All AI-assisted code was manually reviewed, executed, and validated through experiments. For writing, we used GPT-5.6 to polish the English of author-written drafts.

The first author reviewed all AI-assisted code and text in detail, with additional review by coauthors, to verify correctness, accuracy, and consistency with the work performed. We take responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Jianhong Bai, Menghan Xia, Xiao Fu, Xintao Wang, Lianrui Mu, Jinwen Cao, Zuozhu Liu, Haoji Hu, Xiang Bai, Pengfei Wan, et al. Recammaster: Camera-controlled generative rendering from a single video. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14834–14844. IEEE, 2025a.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025b.

BlenderKit. Blenderkit. `https://www.blenderkit.com`, 2026. Accessed: 2026-07-18.

Yuanhao Cai, He Zhang, Xi Chen, Jinbo Xing, Yiwei Hu, Yuqian Zhou, Kai Zhang, Zhifei Zhang, Soo Ye Kim, Tianyu Wang, et al. Omnivcus: Feedforward subject-driven video customization with multimodal control conditions. *Advances in Neural Information Processing Systems*, 38: 115404–115423, 2026.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 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, 2025a.

Guibin Chen, Dixuan Lin, Jiangping Yang, Youqiang Zhang, Zhengcong Fei, Debang Li, Sheng Chen, Chaofeng Ao, Nuo Pang, Yiming Wang, et al. Skyreels-v4: Multi-modal video-audio generation, inpainting and editing model. *arXiv preprint arXiv:2602.21818*, 2026a.

Junyi Chen, Tong He, Zhoujie Fu, Pengfei Wan, Kun Gai, and Weicai Ye. Vino: A unified visual generator with interleaved omnimodal context. *arXiv preprint arXiv:2601.02358*, 2026b.

Shuo Chen, Cong Wei, Sun Sun, Tiancheng Shen, Ping Nie, Kai Zou, Ge Zhang, Ming-Hsuan Yang, and Wenhu Chen. Context forcing: Consistent autoregressive video generation with long context. In *Forty-third International Conference on Machine Learning*, 2026c.

Xi Chen, Zhifei Zhang, He Zhang, Yuqian Zhou, Soo Ye Kim, Qing Liu, Yijun Li, Jianming Zhang, Nanxuan Zhao, Yilin Wang, et al. Unireal: Universal image generation and editing via learning

real-world dynamics. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 12501–12511, 2025b.

Yukang Chen, Luozhou Wang, Wei Huang, Shuai Yang, Bohan Zhang, Yicheng Xiao, Ruihang Chu, Weian Mao, Qixin Hu, Shaoteng Liu, Yuyang Zhao, Huizi Mao, Ying-Cong Chen, Enze Xie, Xiaojuan Qi, and Song Han. Longlive2.0: An nvfp4 parallel infrastructure for long video generation. *arXiv preprint arXiv: 2605.18739*, 2026d.

Jiaxin Cheng, Tianjun Xiao, and Tong He. Consistent video-to-video transfer using synthetic dataset. In *International Conference on Learning Representations*, volume 2024, pp. 16867–16879, 2024.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Justin Cui, Jie Wu, Ming Li, Tao Yang, Xiaojie Li, Rui Wang, Andrew Bai, Yuanhao Ban, and Cho-Jui Hsieh. Self-forcing++: Towards minute-scale high-quality video generation. *arXiv preprint arXiv:2510.02283*, 2025.

DecartAI Team. Lucy edit: Open-weight text-guided video editing. `https://d2drjpuinn46lb.cloudfront.net/Lucy_Edit__High_Fidelity_Text_Guided_Video_Editing.pdf`, 2025.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, et al. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

Epic Games. Fab. `https://www.fab.com`, 2026. Accessed: 2026-07-18.

Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Scandinavian conference on Image analysis*, pp. 363–370. Springer, 2003.

Jiaqi Feng, Justin Cui, Yuanhao Ban, and Cho-Jui Hsieh. One-forcing: Towards stable one-step autoregressive video generation. *arXiv preprint arXiv:2605.23458*, 2026.

Ryan Fosdick. Adapting vace for real-time autoregressive video diffusion. *arXiv preprint arXiv:2602.14381*, 2026.

Yang Fu, Yike Zheng, Ziyun Dai, and Henghui Ding. EffectErase: Joint video object removal and insertion for high-quality effect erasing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *Advances in Neural Information Processing Systems*, 38:75460–75482, 2026.

Michal Geyer, Omer Bar Tal, Shai Bagon, and Tali Dekel. Tokenflow: Consistent diffusion features for consistent video editing. In *International Conference on Learning Representations*, volume 2024, pp. 1608–1620, 2024.

Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, et al. Kubric: A scalable dataset generator. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3749–3761, 2022.

Yoav HaCohen, Benny Brazowski, Nisan Chiprut, Yaki Bitterman, Andrew Kvochko, Avishai Berkowitz, Daniel Shalem, Daphna Lifschitz, Dudu Moshe, Eitan Porat, et al. Ltx-2: Efficient joint audio-visual foundation model. *arXiv preprint arXiv:2601.03233*, 2026.

David Hasler and Sabine E Suesstrunk. Measuring colorfulness in natural images. In *Human vision and electronic imaging VIII*, volume 5007, pp. 87–95. SPIE, 2003.

Qixin Hu, Shuai Yang, Wei Huang, Song Han, and Yukang Chen. Longlive-rag: A general retrieval-augmented framework for long video generation. *arXiv preprint arXiv:2606.02553*, 2026.

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

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21807–21818, 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*, 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.

Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. Musiq: Multi-scale image quality transformer. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5128–5137. IEEE, 2021.

Dongjun Kim, Chieh-Hsin Lai, WeiHsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ode trajectory of diffusion. In *International Conference on Learning Representations*, volume 2024, pp. 44493–44525, 2024a.

Jihwan Kim, Junoh Kang, Jinyoung Choi, and Bohyung Han. Fifo-diffusion: Generating infinite videos from text without training. *Advances in Neural Information Processing Systems*, 37: 89834–89868, 2024b.

Haobo Li, Yanhong Zeng, Yunhong Lu, Jiapeng Zhu, Hao Ouyang, Qiuyu Wang, Ka Leong Cheng, Yujun Shen, and Zhipeng Zhang. Aad-1: Asymmetric adversarial distillation for one-step autoregressive video generation. *arXiv preprint arXiv:2606.03972*, 2026a.

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

Runjia Li, Moayed Haji-Ali, Ashkan Mirzaei, Chaoyang Wang, Arpit Sahni, Ivan Skorokhodov, Aliaksandr Siarohin, Tomas Jakab, Junlin Han, Sergey Tulyakov, et al. Egoedit: Dataset, real-time streaming model, and benchmark for egocentric video editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16042–16053, 2026c.

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.

Kuan Heng Lin, Zhizheng Liu, Pablo Salamanca, Yash Kant, Ryan Burgert, Yuancheng Xu, Koichi Namekata, Yiwei Zhao, Bolei Zhou, Micah Goldblum, et al. Vista4d: Video reshooting with 4d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 32671–32682, 2026a.

Yiqi Lin, Guoqiang Liang, Ziyun Zeng, Zechen Bai, Yanzhe Chen, and Mike Zheng Shou. Kiwi-edit: Versatile video editing via instruction and reference guidance. *arXiv preprint arXiv:2603.02175*, 2026b.

Kunhao Liu, Wenbo Hu, Jiale Xu, Ying Shan, and Shijian Lu. Rolling forcing: Autoregressive long video diffusion in real time. *arXiv preprint arXiv:2509.25161*, 2025a.

Qi Liu, Gang Yue, Mingyu Yin, Lisai Zhang, Yidi Wu, Yaole Wang, Yaohui Wang, Chang Yao, Jingyuan Chen, and Lin Ma. Tide: Task-isolated diffusion for unified video editing and generation. *arXiv preprint arXiv:2606.08260*, 2026.

Shaoteng Liu, Tianyu Wang, Jui-Hsien Wang, Qing Liu, Zhifei Zhang, Joon-Young Lee, Yijun Li, Bei Yu, Zhe Lin, Soo Ye Kim, et al. Generative video propagation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17712–17722, 2025b.

Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models. In *International Conference on Learning Representations*, volume 2025, pp. 50611–50649, 2025.

Jiaxin Lu, Chun-Hao Paul Huang, Uttaran Bhattacharya, Qixing Huang, and Yi Zhou. Humoto: A 4d dataset of mocap human object interactions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10886–10897, 2025.

Yu Meng, Xiangyang Luo, Letian Li, Wenyuan Jiang, Chen Gao, Xinlei Chen, Yong Li, and Xiao-Ping Zhang. Tethercache: Stabilizing autoregressive long-form video generation with gated recall and trusted alignment. *arXiv preprint arXiv:2606.13035*, 2026.

Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik. No-reference image quality assessment in the spatial domain. *IEEE Transactions on image processing*, 21(12):4695–4708, 2012.

Saman Motamed, William Harvey, Benjamin Klein, Luc Van Gool, Zhuoning Yuan, and Ta-Ying Cheng. Void: Video object and interaction deletion. *arXiv preprint arXiv:2604.02296*, 2026.

Kepan Nan, Rui Xie, Penghao Zhou, Tiehan Fan, Zhenheng Yang, Zhijie Chen, Xiang Li, Jian Yang, and Ying Tai. Openvid-1m: A large-scale high-quality dataset for text-to-video generation. In *International Conference on Learning Representations*, volume 2025, pp. 1045–1064, 2025.

Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024.

Yusu Qian, Eli Bocek-Rivele, Liangchen Song, Jialing Tong, Yinfei Yang, Jiasen Lu, Wenze Hu, and Zhe Gan. Pico-banana-400k: A large-scale dataset for text-guided image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 37226–37235, 2026.

ITURBT Recommendation. 709-6: Parameter values for the hdtv standards for production and international programme exchange. *Basic parameter values for the HDTV standard for the studio and for international programme exchange, now ITU-R BT*, 2015.

David Ruhe, Jonathan Heek, Tim Salimans, and Emiel Hoogeboom. Rolling diffusion models. *arXiv preprint arXiv:2402.09470*, 2024.

Mohammadreza Salehi, Mehdi Noroozi, Luca Morreale, Ruchika Chavhan, Malcolm Chadwick, Alberto Gil Ramos, and Abhinav Mehrotra. Rfdm: Residual flow diffusion model for efficient causal video editing. *arXiv preprint arXiv:2602.06871*, 2026.

Tim Salimans, Thomas Mensink, Jonathan Heek, and Emiel Hoogeboom. Multistep distillation of diffusion models via moment matching. *Advances in Neural Information Processing Systems*, 37: 36046–36070, 2024.

Gaurav Sharma, Wencheng Wu, and Edul N Dalal. The ciede2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations. *Color Research & Application: Endorsed by Inter-Society Color Council, The Colour Group (Great Britain), Canadian Society for Color, Color Science Association of Japan, Dutch Society for the Study of Color, The Swedish Colour Centre Foundation, Colour Society of Australia, Centre Français de la Couleur*, 30(1):21–30, 2005.

Sketchfab. Sketchfab. `https://sketchfab.com`, 2026. Accessed: 2026-07-18.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

Bernini Team, Chenchen Liu, Junyi Chen, Lei Li, Lu Chi, Mingzhen Sun, Zhuoying Li, Yi Fu, Ruoyu Guo, Yiheng Wu, et al. Bernini: Latent semantic planning for video diffusion. *arXiv preprint arXiv:2605.22344*, 2026.

Kling Team, Jialu Chen, Yuanzheng Ci, Xiangyu Du, Zipeng Feng, Kun Gai, Sainan Guo, Feng Han, Jingbin He, Kang He, et al. Kling-omni technical report. *arXiv preprint arXiv:2512.16776*, 2025.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. Vggt-ω. *arXiv preprint arXiv:2605.15195*, 2026a.

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

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.

Ronald J. Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1(2):270–280, 1989. doi: 10.1162/neco.1989.1.2.270.

Jianzong Wu, Hao Lian, Jiongfan Yang, Dachao Hao, Ye Tian, Yunhai Tong, Jingyuan Zhu, Biaolong Chen, Qiaosong Qi, Aixi Zhang, et al. Loomvideo: Unifying multimodal inputs into video generation and editing. *arXiv preprint arXiv:2606.06042*, 2026.

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.

Hao Yang, Zhiyu Tan, Jia Gong, Luozheng Qin, Hesen Chen, Xiaomeng Yang, Yuqing Sun, Yuetan Lin, Mengping Yang, and Hao Li. Omni-video 2: Scaling mllm-conditioned diffusion for unified video generation and editing. *arXiv preprint arXiv:2602.08820*, 2026a.

Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, Yingcong Chen, Yao Lu, et al. Longlive: Real-time interactive long video generation. In *ICLR*, 2026b.

Danah Yatim, Rafail Fridman, Omer Bar-Tal, Yoni Kasten, and Tali Dekel. Space-time diffusion features for zero-shot text-driven motion transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8466–8476, 2024.

Bo Ye, Xinyu Cui, Jian Zhao, Tong Wei, and Min-Ling Zhang. Dysink: Dynamic frame sinks for autoregressive long video generation. *arXiv preprint arXiv:2605.21028*, 2026.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T 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.

Mark Yu, Wenbo Hu, Jinbo Xing, and Ying Shan. Trajectorycrafter: Redirecting camera trajectory for monocular videos via diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 100–111, October 2025.

Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. *arXiv preprint arXiv:2409.02048*, 2024.

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.

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

Zhongwei Zhang, Fuchen Long, Wei Li, Zhaofan Qiu, Wu Liu, Ting Yao, and Tao Mei. Region-constraint in-context generation for instructional video editing. *arXiv preprint arXiv:2512.17650*, 2025.

Min Zhao, Hongzhou Zhu, Kaiwen Zheng, Zihan Zhou, Bokai Yan, Xinyuan Li, Xiao Yang, Chongxuan Li, and Jun Zhu. Causal forcing++: Scalable few-step autoregressive diffusion distillation for real-time interactive video generation. *arXiv preprint arXiv:2605.15141*, 2026a.

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*, 2026b.

Kaiwen Zheng, Guande He, Min Zhao, Jintao Zhang, Huayu Chen, Jianfei Chen, Chen-Hsuan Lin, Ming-Yu Liu, Jun Zhu, and Qianli Ma. Causal-rcm: A unified teacher-forcing and self-forcing open recipe for autoregressive diffusion distillation in streaming video generation and interactive world models. *arXiv preprint arXiv:2606.25473*, 2026.

Zhenghong Zhou, Xiaohang Zhan, Zhiqin Chen, Soo Ye Kim, Nanxuan Zhao, Haitian Zheng, Qing Liu, He Zhang, Zhe Lin, Yuqian Zhou, et al. Tri-prompting: Video diffusion with unified control over scene, subject, and motion. *arXiv preprint arXiv:2603.15614*, 2026.

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

Bojia Zi, Penghui Ruan, Marco Chen, Xianbiao Qi, Shaozhe Hao, Shihao Zhao, Youze Huang, Bin Liang, Rong Xiao, and Kam-Fai Wong. Señorita-2m: A high-quality instruction-based dataset for general video editing by video specialists. *Advances in Neural Information Processing Systems*, 38, 2026.

A APPENDIX

A.1 TASK UNIFICATION AND FORMULATION

Table 5 summarizes the EditStream task space. The model optionally receives the text prompt, source video, first frame for propagation, mask, or single reference image and predicts a target video according to the given conditions and special task tokens. This formulation supports both generation and editing. T2V synthesizes new content, whereas I2V, V2V, refV2V, EditProp and ReShoot preserve or transform existing contents. The same model is expected to learn when to preserve, when to modify, and how to balance the input controls without ambiguity.

Table 5: Unified task formulation in EditStream. All tasks share the same video diffusion backbone and generate a video; they differ only in their conditional inputs and special task tokens.

Task	Abbrev.	Conditional Inputs	Special Task Tokens
Text-to-Video	T2V	Text prompt	–
Image-to-Video	I2V	First-frame image; (optional) Text prompt	–
(Egocentric) Video-to-Video	(Ego)V2V	Source (general/egocentric) video; Text prompt	–
Editing Propagation	EditProp	Source video; First-frame edit; (optional) Text prompt	[Editing Propagation]
Reference-guided Video-to-Video	refV2V	Source video; Reference image; Text prompt	[Reference Guidance]
Explicit Camera Pose Control	ReShoot	Re-projected source video; Mask; (optional) Text prompt	[Camera Control]

A.2 DATA CONSTRUCTION

A unified streaming video editing model must support both general video generation and a wide range of editing behaviors, including text-to-video generation, image-to-video, instruction-guided video-to-video editing, editing propagation, reference-guided video-to-video editing, object addition/removal, egocentric manipulation, visual-effect removal, and camera pose control. We therefore train EditStream on a heterogeneous mixture of generative data, paired editing data, and physically grounded video data. All videos are sampled as 81-frame clips and resized into 720p aspect-ratio buckets, mainly 704×1280 and 1280×704. Table 6 summarizes the training mixture. The ratios denote sampling ratios during training.

Table 6: **Training Data Mixture.** Sources, dataset scales, and sampling ratios used to train EditStream.

Category	Source Family	Scale	Sampling Ratio
Video Generation Data			
Text-to-Video	LTX2.3-video, OpenVid-1M Nan et al. (2025)	205K + 1.02M (∼1.23M clips)	21.875%
Text/Image-to-Video	LTX2.3-video, OpenVid-1M Nan et al. (2025)	205K + 1.02M (∼1.23M clips)	21.875%
Video Editing and Control Data			
Image Editing / Propagation	Pico-style image editing Qian et al. (2026)	∼328K image pairs	9.375%
General Video Editing / Propagation	Curated paired video editing	∼37K pairs	6.25%
Egocentric Video Editing / Propagation	EgoEdit Li et al. (2026c)	∼94K pairs	6.25%
Video Effect Removal / Propagation	EffectErase Fu et al. (2026)	∼96K pairs	6.25%
Reference-Guided Video Editing	Kiwi RefVIE Lin et al. (2026b)	∼476K pairs	6.25%
Human–Object Interaction / Propagation	HUMOTO Lu et al. (2025)	∼21K clips	9.375%
Synthetic Object Editing / Propagation	Kubric Greff et al. (2022)	∼50K clips	6.25%
Explicit Camera Pose Control	MultiCamVideo Bai et al. (2025a)	∼57K clips	6.25%

Generation and Image-to-Video Data. We allocate roughly half of the training mixture to general video generation and image-conditioned video generation. Specifically, text-to-video samples are from LTX2.3-video and OpenVid-1M Nan et al. (2025), and text/image-to-video samples are from the same sources. For the LTX2.3-video dataset, we run the inference of LTX2.3 (HaCohen et al., 2026) on user prompts introduced in Self-forcing. This regularization preserves the base model's general video prior and prevents the unified editor from overfitting to paired editing data only.

General Editing and Propagation Data. We use paired video editing data to teach instruction-following transformations. This includes self-collected general source-target video pairs and Pico Qian et al. (2026) image editing pairs converted into short video training examples with smooth translation augmentation. For propagation training, the model receives either an edited first frame or a weak propagation prompt and learns to propagate the edit to the entire video results. These

propagation samples are important for interactive workflows, when users want to edit a single frame, and manipulate the video editing based on the first frame.

Reference-Guided Editing Data. We include KiwiEdit Lin et al. (2026b) reference-guided editing data, containing roughly 476K source-target-reference triplets. These samples provide a source video, an editing instruction, and an additional reference image. They train the model to transfer subject or background from the reference image to the target video while preserving the consistency of the source video.

Physically Grounded Editing Data. Inspired by VOID Motamed et al. (2026), to improve physical consistency, we include several rendering data sources. HUMOTO Lu et al. (2025) contains approximately 21K human-object interaction clips paired with corresponding object-present and object-absent videos. We use it for object addition, object removal, and edit propagation.

To improve the diversity of this dataset, we curate indoor and outdoor 3D scenes from Fab Epic Games (2026), Sketchfab Sketchfab (2026), BlenderKit BlenderKit (2026), and other asset sources. For each selected HUMOTO motion, we manually identify valid sub-sequences and target interaction objects. Each rendered sample is specified by a scene, a HUMOTO sub-sequence, a target object, and a camera trajectory. We render two temporally synchronized videos with identical scene layout, camera motion, human motion, lighting, and appearance: one containing the target object and one with the target object removed. To increase viewpoint diversity, we render third-person, first-person, and object-centric views.

We apply Gemini-based Comanici et al. (2025) VLM filtering to reject low-quality pairs showing penetration, floating artifacts, unclear human-object interactions, heavy occlusion, or strange viewpoints. To improve visual diversity, we randomize human and object appearances across samples while keeping the two videos in each pair appearance-consistent.

We generate paired add/remove editing instructions using Qwen3-VL-32B Bai et al. (2025b) conditioned on the rendered videos and the original HUMOTO captions. For object addition, we provide two instruction variants: a concise object-centric instruction without motion details, and an interaction-aware instruction that explicitly describes the human-object interaction. For video-to-video editing training, we utilize the interaction-aware instruction, while for propagation, we only describe the object.

Kubric-style Greff et al. (2022) synthetic object editing contributes around 50K clips and is used for both object removal and object addition. We also use EgoEdit Li et al. (2026c) for egocentric object replacement and EffectErase Fu et al. (2026) for removing visual effects and their induced artifacts.

Camera Pose Control Data. We construct camera-control training data from the MultiCamVideo Bai et al. (2025a) dataset, which provides synchronized multi-view videos with 10 cameras per dynamic scene. Each camera sequence contains 81 frames, and the dataset covers four focal-length/aperture groups: f18-aperture10, f24-aperture5, f35-aperture2.4, and f50-aperture2.4. For each focal group, we process the first 512 scenes by default. Within each scene, we deterministically sample 30 ordered source-target camera pairs from all possible camera pairs.

For every selected scene, we run VGGT-Omega Wang et al. (2026a) once using all 10 camera videos and all 81 frames. VGGT-Omega predicts per-frame camera poses, intrinsics, and depth maps for all views in a shared reconstruction space. For each sampled source-target pair, we unproject the predicted source-view RGB-D sequence into a 4D point cloud and render it into the VGGT-Omega-estimated target camera trajectory. This produces a point-cloud-rendered video that is geometrically aligned with the target view as closely as possible.

Each training sample consists of the source video, the point-cloud render, the target video, the point-cloud alpha mask, and the corresponding camera metadata. The target video serves as the supervision signal, while the point-cloud render and alpha mask provide geometric conditioning for the video model. Since monocular or learned 3D reconstruction can be imperfect, VGGT-Omega may produce inaccurate poses or depths in some scenes. We therefore compute alignment-related quality statistics, including valid-render coverage and pixel-alignment metrics, and use them to filter generated samples before forming the final training set. Captions are generated with Qwen3-VL-32B-Instruct Bai et al. (2025b) from uniformly sampled target-video frames, producing camera-agnostic scene descriptions used as text conditioning.

Overall, the final dataset balances general generation, image-conditioned generation, instruction editing, propagation, reference guidance, physical object editing, egocentric editing, effect removal, and camera control. This balance is essential for a unified streaming editor: the same model must preserve strong generative quality while handling diverse, temporally coherent editing tasks.

A.3 Implementation Details

Teacher training. We fully fine-tune all 5B parameters of `Wan2.2-TI2V-5B` with FSDP (full sharding) on 8 nodes $\times$ 8 A100-80GB GPUs. Training operates on 704×1280 / 1280×704 resolution buckets at 81 frames. We optimize a flow-matching (v-prediction) denoising loss with AdamW (lr $= 2 \times 10^{-5}$, weight decay 0.01, linear warmup followed by linear decay), effective batch is 192, and maintain an EMA of the weights. $p_{first} = 20\%$ and $p_{ps} = 40\%$.

Student distillation. The student is strictly block-causal. The distillation process follows the two stages introduced in the paper: VMM warm-up and VMM-forcing unrolling. The student, auxiliary, and teacher models in the distillation process are all initialized from the same EMA checkpoint of the teacher, where this checkpoint is obtained after teacher-forcing adaptation. The latent frames of the student are divided into chunks of three frames, and chunk-wise unrolling and KV caching are performed during both inference and training. Training adopts a 4-step timestep schedule, $t \in \{999, 938, 833, 625\}$, which is aligned with the timestep schedule of the teacher's 4-step sampling after shifting. During training, the warm-up stage takes 200 iterations, and the unrolling stage takes 800 iterations. The total number of training iterations is 1000. Training for longer does not lead to significant improvements in performance, while the training remains stable throughout. AdamW is likewise used as the optimizer, with an initial learning rate of lr $= 2 \times 10^{-5}$, which is reduced to 4×10^{-6} after entering the unrolling stage, with $\beta = (0.9, 0.95)$ and a weight decay of 0.01. The effective batch size is 128. Training is conducted on 8 nodes, each equipped with 8 A100-80GB GPUs, for a total of 64 GPUs. The update ratio between the auxiliary model and the student is 1:1. In addition, the Energy-Annealing (EA) factor used to mitigate over-saturation is set to 0.7.

Self-forcing baseline. As a baseline we also distill a causal student with the original self-forcing recipe Huang et al. (2026). The generator is warm-started from a causal ODE-regression checkpoint obtained via an offline Multi-task ODE-pair generation-then-regression procedure, rather than initializing directly from the bidirectional teacher's weights as in our recipe. The fake-score critic is updated $5\times$ more frequently than the generator (dfake_gen_update_ratio $= 5$, vs. 1 in our recipe), with separate learning rates for generator and critic (2×10^{-6} / 4×10^{-7}), effective batch size 128, on the same 8×8 A100-80GB setup. More details are listed in Table 7.

Table 7: Implementation summary across the three training stages.

	Teacher	Student (ours)	Self-forcing baseline
Init	Wan2.2-TI2V-5B pretrained	Teacher EMA weights	Causal ODE-regression ckpt
Causal / block size	No	Yes, 3 frames/block	Yes, 3 frames/block
Denoising steps	full schedule	4 ({999,938,833,625})	4 ({999,938,833,625})
Distillation loss	flow matching	VMM	DMD
Block unrolling	–	yes (200-step warmup)	yes
Critic:generator ratio	–	1:1	5:1
Learning rate	2×10^{-5}	$2 \times 10^{-5} \rightarrow 4 \times 10^{-6}$	2×10^{-6} (gen) / 4×10^{-7} (critic)
Effective batch size	192	128	128
GPUs	64×A100-80GB	64×A100-80GB	64×A100-80GB

A.4 New Evaluation Metrics

A.4.1 Camera Pose Control Metrics

Camera pose change is achieved by warping the frames captured from the original camera to a new camera pose through reprojection. Therefore, the existing frame pixels form the *valid* region, while regions that were previously occluded or did not exist in the original frames and thus need to be generated form the black-pixel *hole* region. We provide a binary mask, $M_t \in 0, 1^{H \times W}$, where the mask value is 1 in the hole region. VBench's overall evaluation metrics typically average over both the valid and hole regions, causing the results to be dominated by the valid region while diluting the contribution of the generated region. Therefore, we introduce four new mask-based metrics.

Hole Complete. measures whether the hole region has been completely filled or remains black as before:

$$\text{HoleComplete} = 100 \cdot \frac{1}{|\mathcal{T}_h|} \sum_{t \in \mathcal{T}_h} \frac{\big|\{p : M_t(p) = 1,\ \bar{Y}_t(p) \geq \tau\}\big|}{\big|\{p : M_t(p) = 1\}\big|}. \tag{9}$$

where $\bar{Y}_t \in [0, 255]$ denotes the grayscale intensity, $\tau = 10$ is a near-black threshold, and $\mathcal{T}_h = \{t : |\{p : M_t(p) = 1\}| > 0\}$ is the set of frames that actually contain hole pixels (frames with an all-valid mask are excluded from the temporal average rather than contributing a value). The metric reaches 100% if and only if all hole pixels throughout the video clip are rendered with new content.

Hole Temp. Stab. mainly measures whether the temporal consistency of the generated content is on par with that of real video. For each consecutive frame pair $(t-1, t)$ we run Farneback dense optical flow Farnebäck (2003) on the grayscale frames, forward-warp frame $t-1$ into frame t, and take the per-pixel absolute warp error. e_t^{hole} and e_t^{valid} are this error averaged over the pixels that are hole (resp. valid) in *both* frames $t-1$ and t (no explicit forward–backward flow-consistency check is performed; restricting to pixels whose category agrees across the pair is what keeps the comparison meaningful):

$$\text{HoleTempStab} = 100 \cdot \frac{1}{|\mathcal{T}_s|} \sum_{t \in \mathcal{T}_s} \min\Big(1,\ \frac{e_t^{\text{valid}}}{e_t^{\text{hole}}}\Big). \tag{10}$$

where $\mathcal{T}_s = \{t \in \{1, \ldots, T-1\} : e_t^{\text{hole}} > \epsilon\}$ with $\epsilon = 10^{-6}$; frame pairs with an empty hole/valid intersection, or with degenerate (near-zero) hole-region flow error, are excluded from the average rather than being epsilon-smoothed. A score of 100 indicates that the hole region exhibits no more temporal flickering than the genuinely observed region of the same video.

Valid SSIM mainly measures the preservation quality of the valid region, i.e., whether the appearance of the original video region has been altered. We compute the SSIM between the generated video Y_t and the point-cloud-rendered reference R_t, restricted to the valid region:

$$\text{ValidSSIM} = \frac{1}{T} \sum_t \text{SSIM}\big(Y_t, R_t \,\big|\, M_t = 0\big). \tag{11}$$

Gated Hole Fill Qual. This metric mainly measures whether the generated content in the hole region is sufficiently plausible. The metric is "gated" because we also require the model to preserve the valid region; otherwise, methods that regenerate the entire video may instead achieve high scores, making it difficult to properly evaluate their hole-filling quality. For each frame we take the largest 8-connected hole component (discarding components smaller than 32×32 px or $1024\,\text{px}^2$ in area, since BRISQUE is unreliable on tiny crops), crop its tight bounding box (no padding, no resizing), and require at least 90% of that component's pixels to already be non-black ($\bar{Y}_t \geq \tau$); frames failing this fill gate are assigned a hole-fill score of 0 directly, since BRISQUE does not reliably penalize partially-black crops. Otherwise we apply BRISQUE Mittal et al. (2012), a no-reference quality assessment model based on natural-scene statistics, to the crop and map the raw score s_{BRISQUE} (lower is better) to a 0–100 scale:

$$\text{HoleFillQual}_{\text{BRISQUE}} = 100 \cdot \text{clip}\Big(1 - \frac{s_{\text{BRISQUE}}}{s_{\max}},\ 0,\ 1\Big), \tag{12}$$

with $s_{\max} = 120$ calibrated so that real photographic content in our data lands in $s_{\text{BRISQUE}} \approx [35, 60]$; scores at or above $s_{\max}$ (including degenerate "zero-variance" crops, e.g. still-flat/black patches, for

which BRISQUE is undefined) are floored to 0. Per-frame scores are averaged over frames, then multiplied by Valid SSIM:

$$\text{GatedHoleFillQual} = \text{HoleFillQual}_{\text{BRISQUE}} \times \text{ValidSSIM}. \tag{13}$$

We do not use MUSIQ Ke et al. (2021) here because it cannot effectively penalize regions that remain ungenerated, such as black regions. Our experiments show that when the regions to be generated remain entirely black, this learned metric cannot reliably distinguish between real images and images containing black holes. In contrast, BRISQUE can effectively distinguish between these cases.

A.4.2 Color Quality of Videos

Colorfulness. We use the Hasler–Süsstrunk metric Hasler & Suesstrunk (2003) to measure colorfulness. With opponent channels $rg = R - G$ and $yb = \frac{1}{2}(R + G) - B$,

$$M \;=\; \sqrt{\sigma_{rg}^2 + \sigma_{yb}^2} \;+\; 0.3\sqrt{\mu_{rg}^2 + \mu_{yb}^2}, \tag{14}$$

where $\mu_{(\cdot)}, \sigma_{(\cdot)}$ are the spatial mean and standard deviation of each opponent channel over the frame. Larger M means a more saturated frame and smaller M means a flatter frame.

CIELAB chroma and HSV saturation. We convert I to CIELAB and compute per-pixel chroma

$$C^*(x) \;=\; \sqrt{a^*(x)^2 + b^*(x)^2}, \qquad \bar{C}^* = \text{mean}_x\, C^*(x), \tag{15}$$

and report the mean HSV saturation $\bar{S} = \text{mean}_x\, S(x)$.

Luminance dynamic range and exposure clipping. Let $Y(x) \in [0, 1]$ denote the Rec. 709 luma Recommendation (2015) of pixel x. We report a percentile-based, exposure-invariant dynamic-range score:

$$\text{DR} = \frac{P_{95}(Y) - P_5(Y)}{P_{95}(Y) + P_5(Y) + \epsilon}. \tag{16}$$

We also measure the fraction of pixels in shadow and highlight clipping regions:

$$\text{ShadowClip} = \frac{1}{N} \sum_x 1\{Y(x) < \tau_{\text{dark}}\}, \tag{17}$$

$$\text{HighlightClip} = \frac{1}{N} \sum_x 1\{Y(x) > \tau_{\text{bright}}\}, \tag{18}$$

where $\tau_{\text{dark}} = 0.02$ and $\tau_{\text{bright}} = 0.98$.

Temporal color flicker. It measures color flickering across frames. We first estimate dense optical flow using the Farneback method Farnebäck (2003). Given two consecutive frames I_t and I_{t+1}, we compute the optical flow $F_{t+1\to t}$ from I_{t+1} to I_t, and then perform backward warping using $F_{t+1\to t}$. We then compute the mean CIEDE2000 color difference Sharma et al. (2005) between the warped frame and I_t in the CIELAB color space:

$$\text{ColorFlicker} = \frac{1}{T-1} \sum_{t=1}^{T-1} \text{mean}_x\, \Delta E_{00}\big(I_t(x), \mathcal{W}(I_{t+1}, F_{t+1\to t})(x)\big), \tag{19}$$

where $\mathcal{W}(I, F)$ denotes backward warping of image I using the estimated optical flow field F, T is the total number of frames, and ΔE_{00} denotes the CIEDE2000 color difference. And the lower of this value, the better.

A.5 Text2Video Results Explanation

To further investigate why the `multiple_objects` score of the Self-forcing baseline is higher than that of VMM-forcing, we hypothesize that this is because the generated videos have stronger motion, causing multiple objects to move out of the frame. To verify this hypothesis, we re-scored the 410 results using the continuous motion score from the RAFT-based optical flow in VBench.

The results show that the `multiple_objects` score of VMM-forcing is negatively correlated with motion: among all results with a score of 0, the average motion is 16.8, while among all videos with a score of 1, the average motion is only 8.5. We then compared the results with Self-forcing on a sample-by-sample basis. Among the 92 prompts where VMM-forcing loses to Self-forcing, 69.6% of the cases have stronger motion, while among the 38 prompts where VMM-forcing wins, this proportion is only 57.9%. This shows that samples with stronger motion are concentrated in the cases where VMM-forcing loses. We manually inspected several examples, and for some prompts, such as "an airplane and a train," the camera moves or zooms, causing objects to leave the frame. Of course, there are also some cases where the detector fails: although we visually observe that the objects are generated completely and remain visible, they still receive a failing score, showing that the detector may have some limitations.

A.6 More Ablation Studies

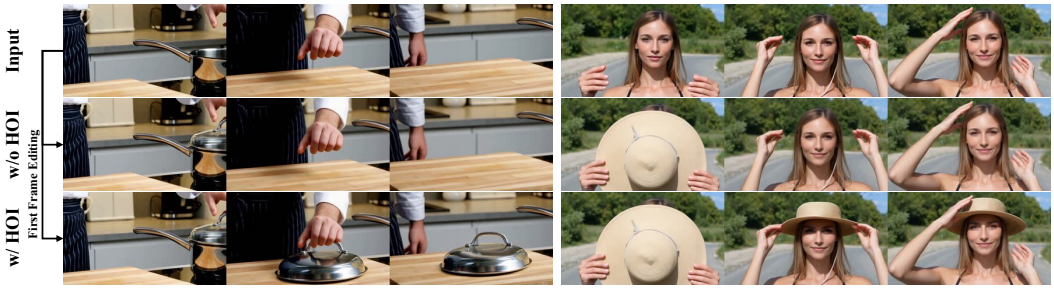


Figure 5: Editing Propagation Results for human-object interaction cases. Without the proposed HOI training data, the model struggles to establish stable physical correspondence between hands and edited objects. As a result, the generated object frequently disappears, fails to follow the intended manipulation, or exhibits severe geometric morphing during interaction. In contrast, incorporating HOI supervision significantly improves the realism and controllability of human–object interactions, making edited virtual objects behave naturally under user manipulation.

Effects of Human-Object Interaction (HOI) Data. To enable realistic interactions between users and newly generated virtual objects, we augment the training data with rendered human–object interaction (HOI) sequences during video-to-video editing and editing propagation training. These synthetic sequences provide explicit supervision for the physical relationship between human hand motions and edited objects, allowing the model to learn plausible object manipulation behaviors.

Effects of Energy Annealing Energy Annealing is essential to improve over-saturation issues caused by Classifier-free Guidance during both teacher inference and distillation. We conducted the ablation study on different energy annealing factors in the proposed EA strategy as shown in Table 8. In the table, we introduced multiple new metrics to better measure the color quality of different models. This is because standard perceptual video metrics (e.g. aesthetic/imaging quality scores from VBench Huang et al. (2024)) use a single scalar which makes it hard to diagnose and differentiate different color issues.

The definition of the metrics can be found in the Appendix. All metrics are computed per frame on the generated RGB clip and averaged over the clip. We report the mean and standard deviation across all vbench videos. Let $I \in [0, 255]^{H \times W \times 3}$ denote a single RGB frame. We utilize the following metrics as shown in Table 8. The detailed definition of these metrics is explained in the appendix.

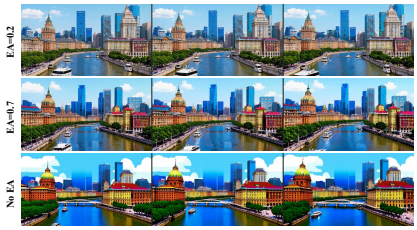
We compute the metrics above over all 946 prompts of the vbench Huang et al. (2024) benchmark for three checkpoints from the EA ablation: the **EA**$= 0.7$ baseline student model, **no-EA** (teacher energy annealing disabled), and **EA**$= 0.2$ (energy-annealing target level lowered from 0.7 to 0.2).

Critically, a model can maximize colorfulness and saturation simply by pushing pixel values toward over-saturation, which we show below is exactly what happens when the teacher's energy annealing (EA) is disabled. Table 8 shows that disabling energy annealing (no-EA) yields the highest colorfulness. In the meanwhile, the variant has the highest shadow/highlight clipping and color flickering. This combination is the quantitative signature of over-saturation. In contrast, EA=0.2 has the lowest

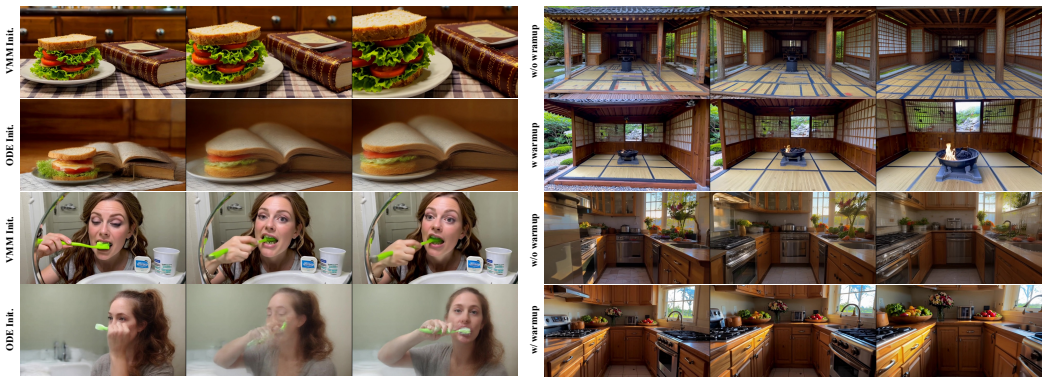
Table 8: **Color Quality**: Comparison of color-related metrics over 946 generations. The first four metrics are descriptive measures of color richness, for which bold marks the highest values; higher richness is not necessarily better. For the final three artifact metrics, bold marks the lowest values.

Metric	EA= 0.7 (ours)	No EA	EA= 0.2
Color richness			
Colorfulness M (Eq. 14)	60.03 ± 27.15	$\mathbf{75.78 \pm 33.11}$	49.58 ± 21.68
Lab chroma $\bar{C}^*$ (Eq. 15)	21.86 ± 10.69	$\mathbf{26.24 \pm 12.36}$	18.77 ± 9.21
HSV saturation $\bar{S}$	0.485 ± 0.174	$\mathbf{0.563 \pm 0.176}$	0.412 ± 0.157
Dynamic range DR (Eq. 16)	0.882 ± 0.129	$\mathbf{0.933 \pm 0.105}$	0.797 ± 0.138
Artifacts / Costs			
Shadow clipping (Eq. 17)	0.073 ± 0.104	0.109 ± 0.109	$\mathbf{0.034 \pm 0.078}$
Highlight clipping (Eq. 18)	0.006 ± 0.025	0.020 ± 0.046	$\mathbf{0.001 \pm 0.004}$
Color flicker ↓ (Eq. 19)	3.16 ± 2.24	3.77 ± 2.43	$\mathbf{2.83 \pm 1.86}$

Figure 6: Visual comparison of sample generations under different EA settings. no-EA yields oversaturation. EA=0.2 yields a flatter and duller look. EA=0.7 balances the colorfulness.



colorfulness range, yielding a flatter and duller look, though having the most temporally stable and least clipping. The baseline EA=0.7 sits between the two and balances the colorfulness and pixel stability.



(a) Quality of different initialization methods. The proposed VMM initialization provides a much stronger starting point.

(b) Importance of Warm-up. Eliminating warm-up introduces noticeable discontinuities and inconsistencies between blocks.

Figure 7: Ablation study visualizations for warm-up existence and initialization methods.

Effect of Warm-up Initialization. The proposed VMM initialization converges within only 200 iterations while already reaching a practically usable performance level. Compared with ODE initialization, VMM initialization significantly reduces the optimization burden of the subsequent unrolling stage by providing a much stronger starting point as shown in Figure 7(a). We also investigate whether the warm-up stage can be removed by directly training with unrolling. Qualitative results in Figure 7(b) further show that directly applying unrolling introduces noticeable discontinuities between autoregressive blocks, making it difficult for the model to learn stable self-convergence. These observations demonstrate that the warm-up stage is an essential prerequisite for efficient and stable autoregressive distillation.

A.7 Distillation Pipeline Comparison

A.7.1 A Unified View of Related Distillation Objectives

VMM (with algorithsm showed in 1) defines a different training objective from related few-step distillation methods, including Distribution Matching Distillation (DMD) Yin et al. (2024b;a), Consistency Distillation (CD) Song et al. (2023); Zhao et al. (2026a); Kim et al. (2024a), and MeanFlow Geng et al. (2026). These methods differ mainly in two aspects: the state on which supervision is applied and the quantity being matched.

Algorithm 1 Velocity Moment Matching (VMM)

Require: Frozen teacher T; trainable auxiliary A and student S
Require: Context $\mathcal{C}$; schedule $\{\tau_i\}_{i=0}^{J}$, $\tau_i > \tau_{i+1}$
for each training iteration **do**
  Sample $i \sim \mathrm{Unif}\{0, \dots, J-1\}$
  Set $t \leftarrow \tau_i$
  Sample $s \sim \mathcal{U}(\tau_{i+1}, \tau_i)$
  Sample noisy latent $\mathbf{x}_t$ from ground truth data sample
  $v_t^S \leftarrow v_S(\mathbf{x}_t, t, \mathcal{C})$
  $\mathbf{x}_s \leftarrow \mathbf{x}_t + (\sigma_s - \sigma_t) v_t^S$
  $\bar{v} \leftarrow v_t^S$
  $v_T^s \leftarrow v_T(\mathbf{x}_s, s, \mathcal{C})$
  $v_A^s \leftarrow v_A(\mathbf{x}_s, s, \mathcal{C})$
  Auxiliary update
  $\mathcal{L}_A \leftarrow \|v_A^s - \mathrm{sg}(\bar{v})\|_2^2$
  $A \leftarrow A - \eta_A \nabla_A \mathcal{L}_A$
  Moment-matching target
  $\tilde{v} \leftarrow \mathrm{sg}(\bar{v} + v_T^s - v_A^s)$
  Student update
  $\mathcal{L}_S \leftarrow \|\bar{v} - \tilde{v}\|_2^2$
  $S \leftarrow S - \eta_S \nabla_S \mathcal{L}_S$
end for

Relation to Distribution Matching Distillation (DMD). The distillation scheme of VMM is closely related to DMD Yin et al. (2024b), sharing a similar teacher–fake residual learning scheme. However, their matching objectives and matching states are different. In DMD, after obtaining an estimated clean endpoint $\hat{\mathbf{x}}_0^S$ from the student, which can be induced from v_t^S, an independent DMD noise level r is sampled and the endpoint is re-noised as

$$\mathbf{y}_r^S = (1 - \sigma_r)\hat{\mathbf{x}}_0^S + \sigma_r \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{20}$$

The teacher model and a trainable fake-score model, which plays a role similar to the auxiliary model in VMM, are evaluated at the same noisy state $(\mathbf{y}_r^S, r)$. Their score difference provides the gradient for matching the diffused endpoint marginals of the student and teacher.

Here, r is a randomly sampled noise scale used to examine the endpoint distribution. It is not the destination of a transition from the source timestep t. DMD therefore provides a multi-scale distributional correction to the student's endpoint distribution, but does not explicitly constrain the intermediate states or ODE trajectory used to produce that endpoint.

VMM retains a teacher–auxiliary residual structure, but evaluates the teacher and auxiliary velocities at an intermediate state produced by the few-step student sampler. Specifically, $s < t$ is sampled within the current inference interval, and both models are evaluated at the same student-induced state $(\mathbf{x}_s^S, s)$. VMM therefore applies a velocity-moment correction at states reached by the student. This is intended to encourage teacher-consistent local dynamics during few-step inference.

Relation to Consistency Distillation, CTM, and MeanFlow. Consistency Distillation Song et al. (2023); Lu & Song (2025) learns a clean-endpoint predictor by enforcing consistent predictions between states on the same teacher ODE trajectory. In discrete CD, the lower-time state is obtained from the higher-time state using a teacher ODE solver, whereas continuous-time CD takes the infinitesimal limit of this consistency relation and evaluates the resulting trajectory derivative using a Jacobian–vector product (JVP). The learned consistency function represents a global mapping from an arbitrary timestep t to the clean endpoint.

Consistency Trajectory Models (CTM) Kim et al. (2024a) generalize this fixed-endpoint parameterization by directly learning an anytime-to-anytime flow map$G_\theta(\mathbf{x}_t, t, s) \approx \mathbf{x}_s$ for $s < t$. MeanFlow Geng et al. (2026) instead represents arbitrary finite-time transport through an average-velocity field. In the teacher-distillation adaptation considered here, MeanFlow predicts the average teacher velocity over the finite interval from t to s, which equivalently determines the teacher state reached

at timestep s.[1] Thus, CD learns a $t \to 0$ endpoint map, whereas CTM and teacher-distilled MeanFlow parameterize arbitrary $t \to s$ finite-time transport through a state map and an average-velocity field, respectively.

VMM also uses a pair of coupled timesteps (t, s), but the role of the destination state is different. CD, CTM, and teacher-distilled MeanFlow construct supervision from a teacher or reference trajectory. In contrast, VMM first produces $\mathbf{x}_s^S$ using the student's own transition and then evaluates the teacher–auxiliary velocity residual at this state. Therefore, in CTM and teacher-distilled MeanFlow, the reference finite-time transition determines the supervision target; in VMM, the student destination $\mathbf{x}_s^S$ instead serves as the query state at which the velocity correction is evaluated.

Unified Fixed-Point View. Let

$$\mathbf{x}_s^T = \Phi_T^{t\to s}(\mathbf{x}_t), \qquad \mathbf{x}_s^S = \Psi_S^{t\to s}(\mathbf{x}_t) \tag{21}$$

denote the destinations produced by the teacher and student respectively. The teacher's average velocity within the interval is

$$u_T^{t\to s}(\mathbf{x}_t) = \frac{\Phi_T^{t\to s}(\mathbf{x}_t) - \mathbf{x}_t}{\sigma_s - \sigma_t}. \tag{22}$$

Equivalently,

$$\mathbf{x}_s^T = \Phi_T^{t\to s}(\mathbf{x}_t) = \mathbf{x}_t + (\sigma_s - \sigma_t) u_T^{t\to s}(\mathbf{x}_t). \tag{23}$$

For DMD, let $\mathbf{x}_0^M \sim p_M$, where $M \in \{S, T\}$, denote a student or teacher endpoint sample. We define its diffused marginal at noise level r as

$$p_{M,r} := \mathrm{Law}\left((1-\sigma_r)\mathbf{x}_0^M + \sigma_r \boldsymbol{\epsilon} \middle| \mathcal{C}\right), \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{24}$$

For VMM, define the conditional first moment of the student transition velocity as

$$m_S(\mathbf{x}, s, \mathcal{C}) = \mathbb{E}\left[\bar{v}_S^{t\to s} \mid \mathbf{x}_s^S = \mathbf{x}, s, \mathcal{C}\right]. \tag{25}$$

The ideal fixed points of the four objectives can then be summarized as

$$\begin{aligned} \text{CD}: &\quad f_S(\mathbf{x}_t, t, \mathcal{C}) \approx \Phi_T^{t\to 0}(\mathbf{x}_t, \mathcal{C}), \\ \text{MeanFlow}: &\quad u_S(\mathbf{x}_t, t, s, \mathcal{C}) \approx u_T^{t\to s}(\mathbf{x}_t), \\ \text{DMD}: &\quad p_{S,r} \approx p_{T,r}, \\ \textbf{VMM (ours)}: &\quad m_S(\mathbf{x}_s^S, s, \mathcal{C}) \approx v_T(\mathbf{x}_s^S, s, \mathcal{C}). \end{aligned} \tag{26}$$

The equations above characterize the ideal solutions of the respective objectives, while their practical training and optimization procedures are often more involved to ensure stable training. Broadly, CD and MeanFlow perform sample-wise flow-map matching, using teacher trajectories to specify target states. DMD performs distribution-wise endpoint matching without requiring sample-wise regression. VMM, in contrast, matches a conditional velocity moment at an intermediate state generated by the student.

Because CD and MeanFlow provide paired supervision from teacher trajectories, they generally offer stable training and help preserve the modes and diversity of the teacher distribution. However, since their objectives regress either a clean endpoint or a finite-time teacher flow map, few-step distillation or limited student capacity may lead to smoother outputs.

In contrast, DMD directly optimizes the distribution of student-generated outputs through a teacher–fake score correction. Since the objective tends to concentrate updates in high-density regions of the teacher distribution, it often produces sharper and perceptually higher-quality samples, while also carrying a higher risk of mode dropping or mode collapse Yin et al. (2024b;a). In autoregressive video generation, this behavior may also lead to low-motion solutions.

VMM falls somewhere in between. Similar to DMD, it applies a teacher–auxiliary residual correction at student-generated states, which can improve local generation quality and recover fine details.

[1] The original MeanFlow is teacher-free and is trained from scratch. Here, we refer specifically to a teacher-distillation adaptation in which the finite-interval average-velocity target is obtained from a pretrained teacher trajectory.

At the same time, because the matching state is associated with a specific t→s student inference transition, VMM also constrains the local sampling dynamics. VMM is therefore expected to strike a balance between perceptual quality and teacher-consistent few-step trajectories. Nevertheless, its final generation quality still depends on factors such as timestep scheduling during optimization and the stability of the auxiliary model.

A.7.2 VMM INITIALIZATION VERSUS OTHERS

Relation to Existing Causal Few-Step Initializations. Existing autoregressive diffusion distillation methods commonly construct a causal few-step student before self-forcing refinement. These initialization methods differ mainly in how the denoising states are constructed and whether the student is trained to regress a fixed teacher flow map or to correct its own transition.

ODE-pair initialization. CausVid and Self-Forcing Yin et al. (2025); Huang et al. (2026) initialize the causal student using ODE pairs generated by a bidirectional teacher. Intermediate states are precomputed from the teacher trajectory, and the student is trained to regress the corresponding clean endpoint. Different blocks can be selected at different stored ODE timesteps similar to Diffusion Forcing. Direct endpoint regression can be underdetermined for the causal student. Causal Forcing Zhu et al. (2026) addresses the issues by first training an autoregressive diffusion teacher with teacher forcing, and then generating causal ODE pairs under the same clean-prefix context used by the student. This produces a better-aligned flow-map target, but requires a separate causal-teacher training stage and offline generation.

Causal consistency initialization. Causal Forcing++ Zhao et al. (2026a) replaces full ODE-pair regression with causal Consistency Distillation. Causal ODE regression and causal CD learn the same autoregressive teacher flow map, but approach it differently. ODE initialization regresses from an intermediate state directly to the clean endpoint, whereas causal CD propagates the endpoint map through local consistency between adjacent timesteps. The latter avoids precomputing complete teacher trajectories and reduces the optimization gap in each training pair.

Causal-rCM Zheng et al. (2026) follows the same general teacher-forcing consistency route. It first converts the bidirectional model into a multi-step causal diffusion teacher, then initializes the few-step student using TF-dCM or continuous-time variants such as TF-sCM and TF-MeanFlow. It requires additional JVP-compatible training infrastructure.

VMM warm-up. The proposed VMM warm-up does not precompute teacher's ODE pairs, train a separate causal teacher, regress the student to a paired teacher destination, or use JVP-based consistency training. It indeed requires a teacher-forcing adaptation but the teacher itself can be retained bidirectional and the adaptation is efficient. Also, the same VMM objective can be retained when moving from warm-up to unrolling.

Note that the bidirectional teacher and auxiliary model can use the clean suffix, while the causal student cannot. Consequently, VMM avoids direct regression to a future-dependent bidirectional endpoint, so the frame-level injectivity requirement of ODE-pair regression Zhu et al. (2026) does not directly apply. However, the teacher–auxiliary residual can still contain future information, and the causal student effectively learns its projection onto the information available from the causal prefix. But to further address the teacher-forcing exposure gap, a subsequent unrolling stage is still necessary and preferred.

A.7.3 VMM-FORCING UNROLLING VERSUS SELF-FORCING UNROLLING

Our unrolling procedure is inspired by Self-Forcing Huang et al. (2026), and both methods address exposure bias by generating the autoregressive history with the student itself and maintaining that history through a causal KV cache. However, they differ in both the within-block rollout strategy and the distillation objective.

Original Self-Forcing randomly samples a denoising exit index $q \in \{1, \dots, J\}$. For every temporal block, it runs the diffusion trajectory only from the initial noise level τ_J to τ_q, predicts a clean endpoint from that exit state, and commits this prediction to the KV cache:

$$\mathbf{x}^k_{\tau_J} \longrightarrow \cdots \longrightarrow \mathbf{x}^k_{\tau_q} \longrightarrow \widehat{\mathbf{x}}^{k,q}_0 \longrightarrow \mathrm{KV}\,. \tag{27}$$

The temporal autoregressive rollout covers all blocks, but the within-block denoising depth is truncated. Sampling different exit indices across iterations allows different denoising steps to receive gradients while reducing training cost. A holistic distribution-matching objective, such as DMD, SiD, or a GAN loss, is then applied to the generated video.

In contrast, our VMM unrolling completes the deployed few-step trajectory for every block before committing its output:

$$\mathbf{x}^k_{\tau_J} \longrightarrow \mathbf{x}^k_{\tau_{J-1}} \longrightarrow \cdots \longrightarrow \mathbf{x}^k_0 \longrightarrow \mathrm{KV}, \tag{28}$$

while VMM is evaluated on a selected local transition $t \to s$ along the trajectory. Consequently, the history used by every later block is produced by the same complete sampler used at inference, rather than by an endpoint prediction obtained from a randomly truncated denoising trajectory.

The two approaches also optimize different quantities according to the difference of DMD and VMM. Self-Forcing with DMD directly matches the distribution of complete student-generated videos and is free to learn a trajectory-independent generation shortcut. VMM unrolling instead applies a teacher–auxiliary velocity residual at intermediate states produced by the student's current few-step trajectory, which makes the training more stable.

Importantly, once a sufficiently strong causal few-step initialization is available, both self-forcing unrolling and VMM unrolling are valid post-training candidates. DMD provides a direct endpoint-distribution correction and may be preferable when the main goal is perceptual quality or sharpness, whereas VMM provides a trajectory-conditioned velocity correction that is more closely aligned with a fixed deterministic ODE schedule. We use VMM in both warm-up and unrolling so that the distillation objective remains unchanged across the two stages.

A.8 More Multi-task Visual Results

We show more qualitative results of T2V (Figure 8), V2V (Figure 9), RefV2V (Figure 10), EgoCentric V2V (Figure 13), and ReShoot (Figure 11).

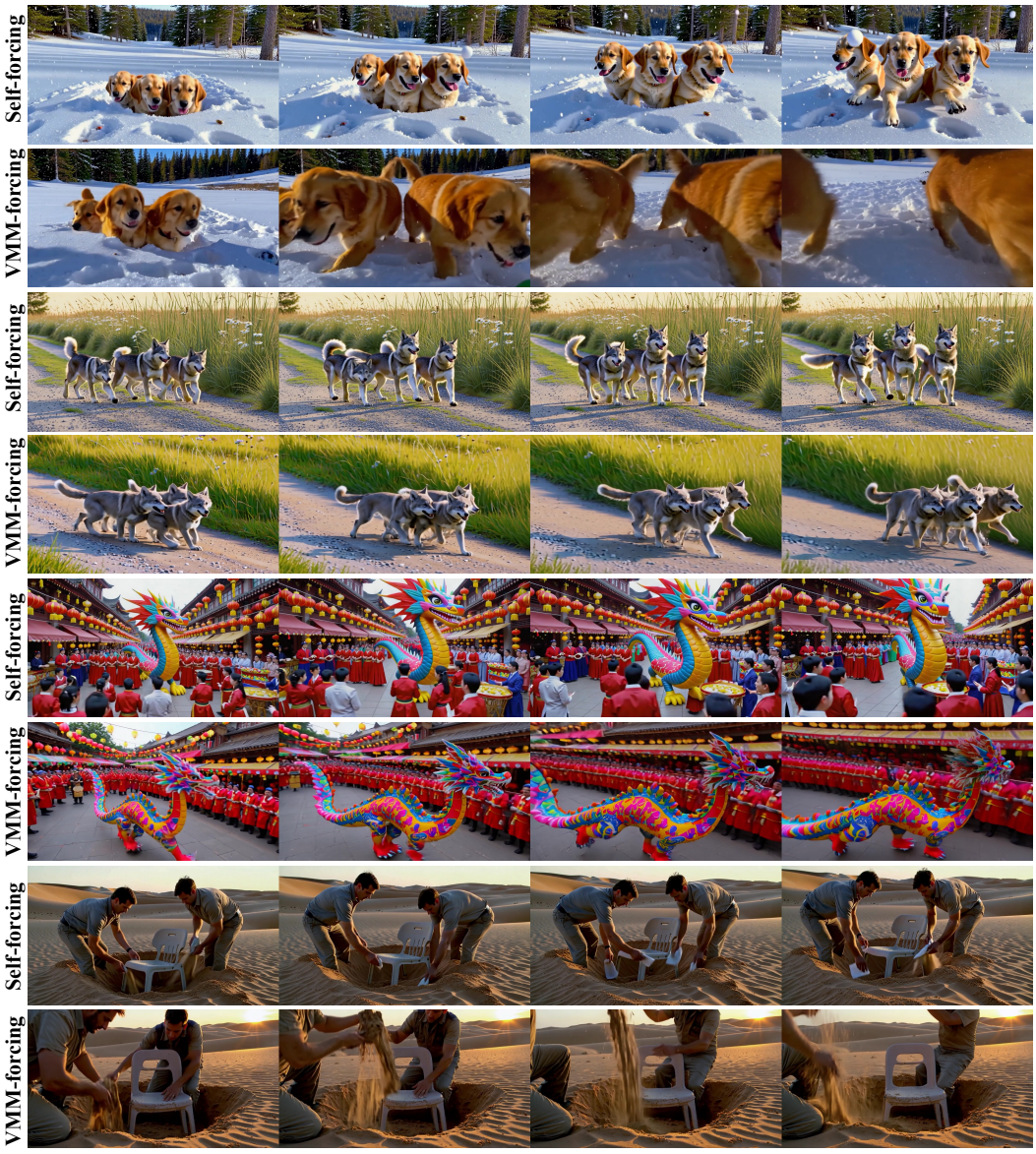


Figure 8: Qualitative Text-to-Video (T2V) results on frequent user prompts. We compare the Self-forcing conversion and the proposed VMM-forcing pipeline. VMM-forcing produces richer motion dynamics and scene evolution while maintaining texture realism and temporal coherence, whereas Self-forcing often yields over-sharpened, plastic-like appearances and reduced temporal diversity.

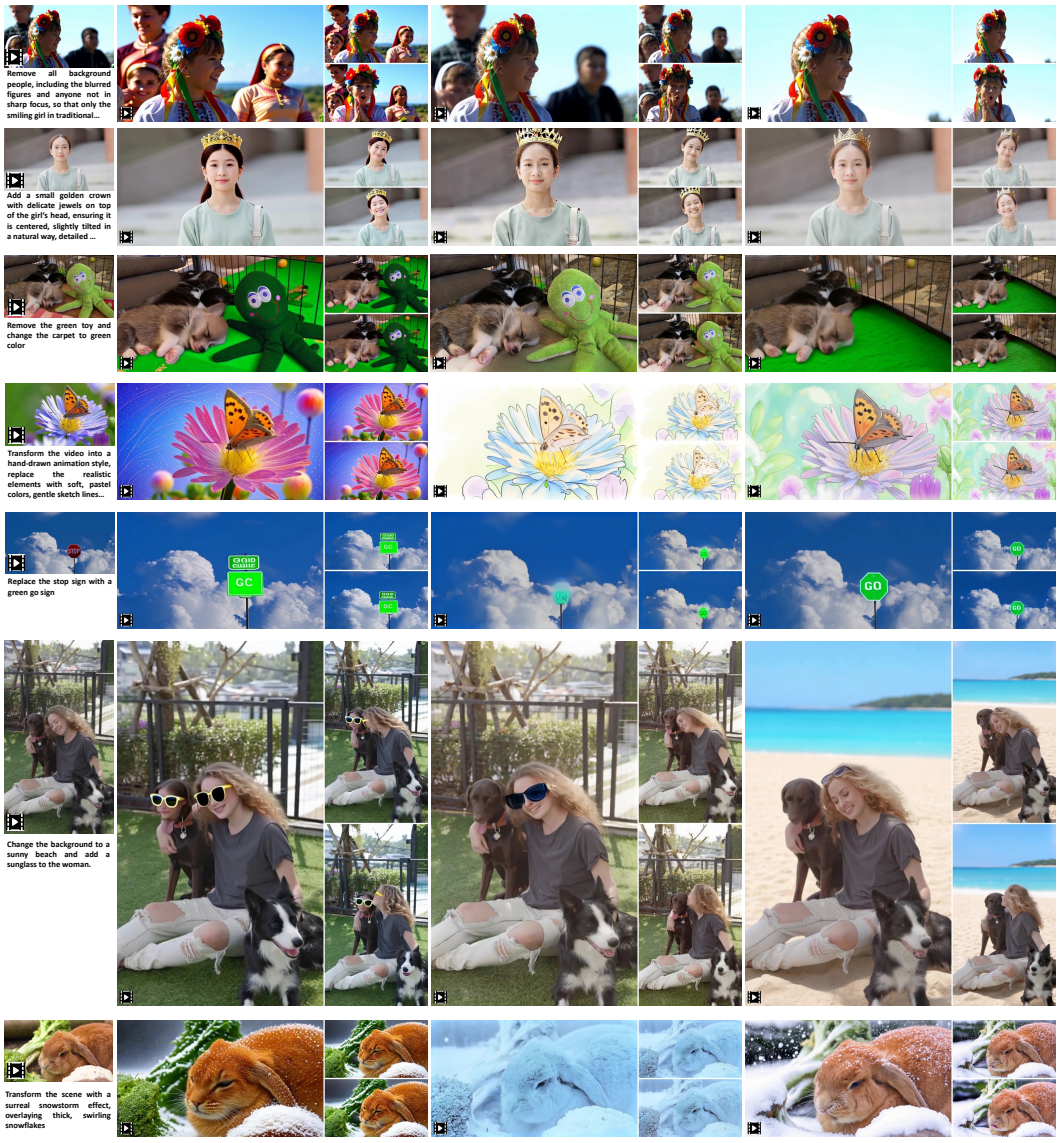


Figure 9: Video-to-Video (V2V) editing results on EditVerse Benchmark compared with other streaming video editing baselines. Across a wide range of editing tasks, including object addition, removal, replacement, style transfer, text rendering, and complex compositional edits requiring precise pixel alignment and instruction following, EditStream consistently exhibits stronger semantic understanding, higher generation fidelity, and better color quality.



Figure 10: Reference Video-to-Video (refV2V) results on RefVIE Benchmark. Qualitatively, EditStream (online streaming model) consistently outperforms Kiwi-Edit (offline model) in preserving the reference subject while generating more realistic edited videos. Compared with the baseline, it produces more faithful identity preservation, more appropriate object scale, more coherent scene composition with the reference background, and richer pixel-level details.



Figure 11: Camera Pose (ReShoot) Control Results on the collected V2VCameraPose Benchmark. TrajCrafter and Vista4D are both strong offline baselines. However, they occasionally suffer from over-hallucination, introducing implausible objects or environmental elements. EditStream generally produces milder distortions while avoiding unnecessary hallucinated content.

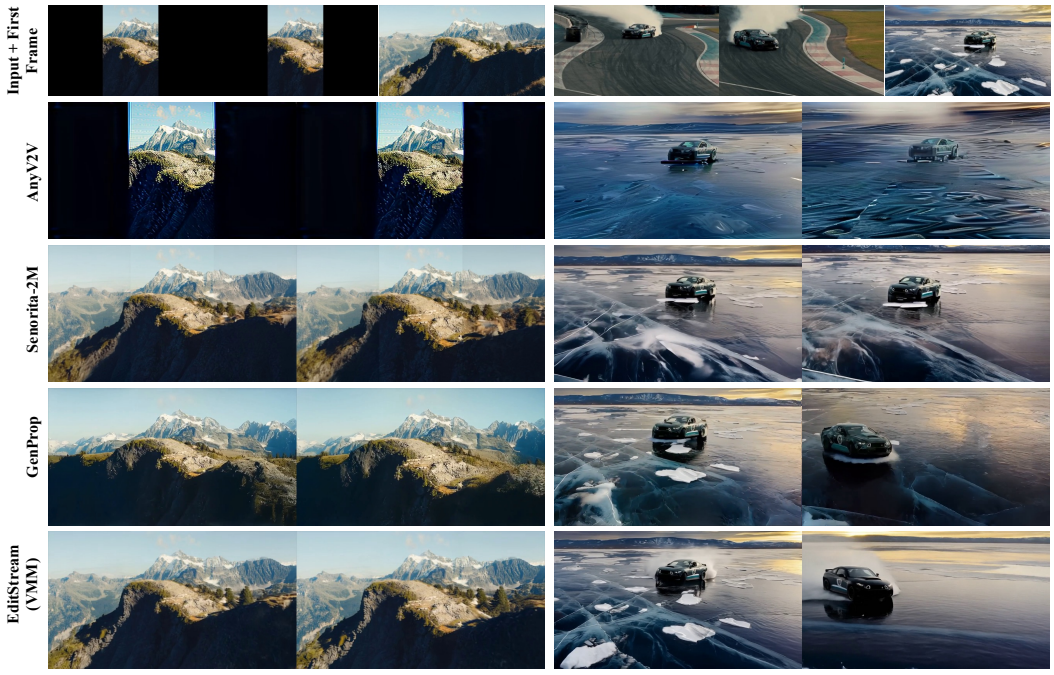


Figure 12: Editing Propagation Results. AnyV2V and Señorita-2M struggle in challenging scenarios. Although EditStream is not trained specifically for outpainting, it demonstrates strong generalization capability. EditStream results are more realistic and detailed than that of GenProp (e.g. it produces a more realistic effect behind the car). It may partly benefit from differences in the underlying foundation model.

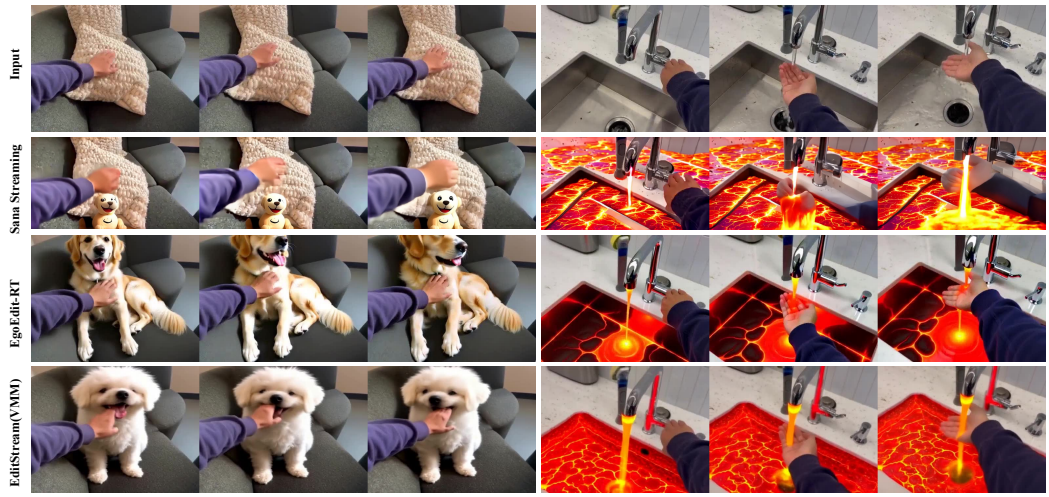


Figure 13: Egocentric Editing (Ego V2V) Results. EgoEdit tends to produce sharper outputs, but they often appear less realistic. It may partly reflect the sharpness and mode-seeking behavior associated with DMD-based training. In the dog editing example, EgoEdit generates overly pronounced fur textures and introduces unrealistic specular highlights on the dog's body. Similarly, in the lava editing example, the generated lava shows a CGI-like texture. EditStream with VMM produces outputs that are slightly less sharp but exhibit more faithful material properties, more realistic textures, and a more natural overall appearance.