StreamEdit: Training-Free Video Editing via Few-Step Streaming Video Generation

Guanlong Jiao[1,4], Chenyangguang Zhang[2], Jia Jun Cheng Xian[1,4], Zewei Zhang[1,3], and Renjie Liao[1,4,5] †

[1] The University of British Columbia [2] ETH Zürich [3] McMaster University
[4] Vector Institute [5] Canada CIFAR AI Chair
{gjiao, anthony, rjliao}@ece.ubc.ca,
chenyangguang.zhang@inf.ethz.ch, zhanz561@mcmaster.ca

Abstract. Although existing video editing methods are generally feasible, they often require many costly iterations and still struggle to deliver high-quality yet satisfying editing results. We attribute this limitation to the prevalent data-to-data paradigm, which is less compatible with modern generative models than noise-to-data generation. To address this gap, we revisit video editing from a noise-to-data perspective and propose **Stream**ing-Generation-based Video **Edit**ing (**StreamEdit**), which preserves few-step sampling while seamlessly injecting source-video conditions. Built on pre-trained streaming generation models, StreamEdit introduces dual-branch fast sampling with a self-attention bridge and cross-attention grounding/boosting to satisfy both sampling and conditioning requirements. We further propose source-oriented guidance to improve target-generation quality, and a visual prompting strategy to enhance editing flexibility and practicality. The method is effective, robust, and generalizable across different models. Extensive experiments on diverse video editing tasks show that StreamEdit consistently outperforms existing approaches, even in few-step settings with minimal time cost. Code and results are available at: dsl-lab.github.io/StreamEdit/.

Keywords: Video Editing · Few-Step Sampling · Streaming Generation

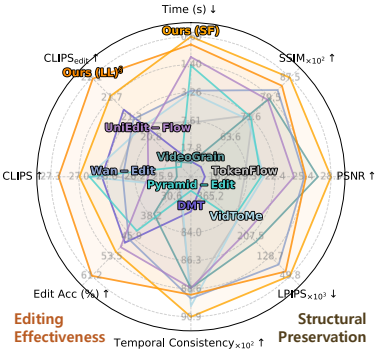
1 Introduction

With the rapid progress of iterative visual generation models [16, 35, 73, 77], training-free visual editing has achieved promising results in images [28,30,33,49], 3D assets [4,93], and videos [31,44,55,71]. This progress has also enabled broader applications, including virtual reality [25,63], embodied intelligence [14,38], *etc.*

Mainstream training-free editing follows a **data** $\rightarrow$ **data** paradigm and is commonly divided into inversion-based [7,28,70] and inversion-free methods [32, 33, 79]. Inversion reverses the generation process to recover reconstructive noise

† Corresponding author.

(a) Text (and image) -driven video editing of StreamEdit



(b) Spider chart of video editing on FiVE-Bench

(c) Autoregressive video editing for long videos using StreamEdit

Fig. 1: Training-free text-driven (optional image-conditioned) streaming video editing of our proposed StreamEdit. StreamEdit supports both text-driven video editing and first-frame-prompted editing, suitable for various editing tasks while having superiority in preservation of editing-irrelevant regions. Developed on text-to-video streaming generation models, StreamEdit supports efficiently video editing for any length, with less than 0.32 second per frame (5 steps) on a single A100 GPU.

[28, 50, 64]. Accordingly, inversion-based methods perform **source** → **noise** → **target** transfer, but incur extra iterations and error accumulation from inversion, which often causes distortion in editing-irrelevant regions for complex videos (*e.g.*, UniEdit-Flow in Fig. 4). Inversion-free methods instead perform direct **source** → **target** transfer with noisy auxiliaries [32, 33]. However, because early samples remain close to the source, visible edits usually emerge only after many iterations. Moreover, their irregular sampling trajectories typically require small, careful step sizes, leading to slow and unstable behavior on challenging cases [32]. Overall, although these methods are feasible, video editing remains constrained by high computational cost and limited edit quality.

This limitation motivates a different view: video editing is fundamentally a video-conditioned video generation problem [10, 27, 76]. Modern generators follow the **noise** → **data** paradigm and are increasingly optimized for few-step, high-quality inference [46, 53, 64]. Representative advances, including consistency models [65], rectified flow [45], and distribution matching distillation [84], have substantially accelerated generation. Yet these benefits are rarely transferred to

editing, because data-to-data pipelines are tightly coupled to their own iterative procedures and therefore cannot directly exploit fast few-step sampling [65].

Our key insight is to reformulate editing as source-conditioned **noise** $\rightarrow$ **target** generation. This reformulation preserves the few-step sampling while injecting source-video conditions, enabling faster and higher-quality editing.

To this end, we propose **Stream**ing-Generation-based Video **Edit**ing, termed **StreamEdit**. As high-quality editing requires both target transformation and faithful preservation of structure, motion, and background, we build StreamEdit on pre-trained streaming generators with a dual-branch few-step sampler that produces parallel source and target features under the shared noise. We then propose a self-attention bridge, combined with cross-attention grounding, to transfer source priors for structural and motion consistency and background preservation. To improve controllability, we further design cross-attention regional boosting as a valve for editing strength. To reduce stochastic instability, we add source-oriented guidance that better aligns editing-irrelevant regions with the source video. Beyond text-only editing, we also provide an optional visual prompting mechanism using a target first frame to improve detail control and reliability.

Overall, StreamEdit establishes a training-free, generation-style paradigm for efficient and effective few-step streaming video editing. Our contributions can be summarized as follows:

- We reformulate training-free video editing as few-step streaming generation and introduce a corresponding sampling-and-guidance framework.
- We design dual-branch sampling with a self-attention bridge and cross-attention grounding/boosting to inject source conditions seamlessly.
- We introduce optional visual prompting, and finally enabling efficient, effective, flexible, and length-unrestricted streaming video editing.
- As shown in Fig. 1, extensive experiments and user studies demonstrate consistent superiority of StreamEdit, supporting a viable new paradigm for training-free video editing.

2 Related work

2.1 Image and Video Editing

Modern generative models are predominantly built on diffusion [23, 64] or flow matching [1, 43, 45], which iteratively map noise distributions to data distributions. Training-free visual editing typically follows a distinct iterative data $\rightarrow$ data paradigm [2, 22, 49], and mainly includes inversion-based and inversion-free approaches. Inversion-based methods first add noise to the source sample and then denoise under target conditions, implicitly performing data transfer through source $\rightarrow$ noise $\rightarrow$ target [12, 13, 20, 28, 30, 50, 58, 71]. Inversion-free methods instead perform direct source $\rightarrow$ target editing, usually by leveraging noised samples to approximate transfer trajectories [18, 32, 33, 79]. Attention manipulation is also widely used to improve editing controllability and reliability [3, 7, 37, 51, 70]. Training-based editing has also advanced, but often depends

on training-free methods for data preparation and inherits corresponding bottlenecks [5, 10, 76, 88, 94]. Although training-free methods are applicable to both images and videos, the higher complexity and dimensionality of video data typically lead to slower inference and reduced editing quality.

2.2 Streaming Video Generation

Streaming generation refers to real-time, chunk-wise dynamic generation; in video generation, it is commonly realized through temporal autoregression [9, 21, 40, 89]. These methods condition each chunk on previously generated frames to produce temporally coherent long videos. Recently, many works distill short-video generators into autoregressive models, preserving pre-trained generation quality and speed while introducing previous-frame conditioning with minimal architectural change [11, 78, 85]. In addition, similar to key-value (KV) caching in language models [34, 69], injecting previous conditions via visual KV modification in attention blocks has become a standard design choice [24, 26, 81]. These distribution-matching-distilled approaches generally support fast few-step video generation. However, constrained by existing training-free editing paradigms, these advances have not yet translated into substantially faster and more effective video editing.

3 Preliminary

3.1 Flow Matching

Most of the existing high-quality generation models are based on diffusion [23, 57, 64], which learns the transformation from known distributions (π_1, *e.g.*, $\mathcal{N}(0, I)$) to complex data distributions π_0. One special case, flow matching [1, 43, 45], has gained significant traction and is widely applied to video generation tasks [29, 54, 73] due to its effectiveness and simple training objective:

$$\min_{\theta} \mathbb{E}_{x_0, x_1, t}\left[\left\|(x_1 - x_0) - v_\theta\left(x_t, t\right)\right\|^2\right], \quad x_t = (1-t)x_0 + tx_1, \quad t \in [0, 1], \tag{1}$$

which aims to learn a straight trajectory between data $x_0 \in \pi_0$ and noise $x_1 \in \pi_1$ by training the velocity function v_θ with $x_1 - x_0$ as the objective.

3.2 Consistency Models

Consistency models are designed for fast generation, establishing the direct mapping between noise and data [65]. The sampling strategy of consistency models is widely adopted by few-step models [48, 60]. It first predicts the clean data from a noisy sample, then obtains the next-step noisy data via a linear interpolation from the prediction and a random noise. Its flow-based formula is:

$$z_{t_{i-1}} = x_{t_i} - t_i v_\theta(x_{t_i}, t_i), \quad x_{t_{i-1}} = (1-t)z_{t_{i-1}} + t_{i-1}\epsilon, \quad \epsilon \sim \mathcal{N}(0, 1), \tag{2}$$

where $z_{t_{i-1}}$ is the one-step prediction of clean data from the previous noisy sample x_{t_i}, the final output is z_{t_0}, and $i \in [N]$, $t_N = 1$, $t_0 = 0$, $x_{t_N} \sim \mathcal{N}(0, 1)$.

3.3 KV Caching in Streaming Video Generation

To reduce redundant computations, recent work has increasingly favored storing and incorporating prior information as a KV cache through one-time computation [24,26,81], as opposed to methods using overlapping sliding windows [6,9,92]. Take Self Forcing [24] and LongLive [81] as examples, these methods conduct one model forward step using previously generated clean data as the input, caching its KV for providing previous condition and then injecting previous KV into self-attention, thus making the streaming generation coherent and consistent.

4 Method

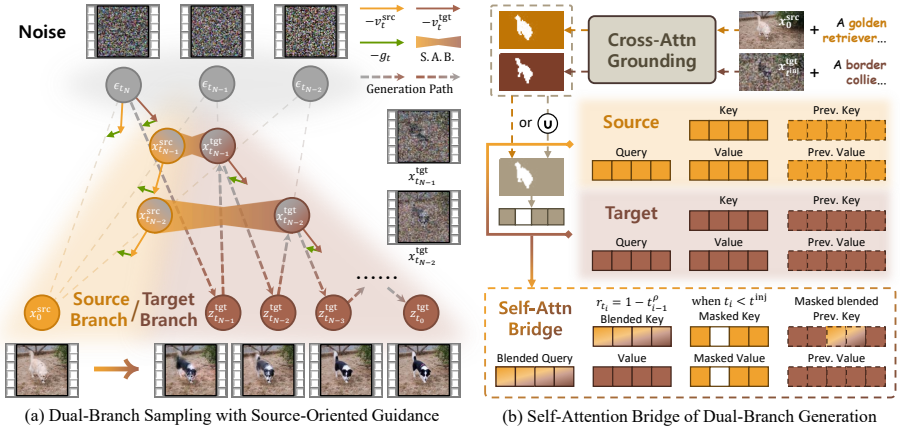
We reformulate video editing as fast streaming generation. Built on representative streaming generators, including Self Forcing [24] and LongLive [81], our method develops source-video-conditioned target generation through few-step sampling and attention manipulation. First, we extend stochastic few-step sampling to a dual-branch framework that propagates noised source information in parallel with the target sample (Fig. 2 (a)). We then design a self-attention bridge to inject source conditions into target generation through the model's dynamic attention mechanism (Fig. 2 (b)). To suppress stochastic artifacts in editing-irrelevant regions, we introduce source-oriented guidance (green arrow in Fig. 2 (a)), analogous to classifier-free guidance for quality improvement. We further introduce cross-attention grounding and boosting, which provide the mask for the self-attention bridge and a controllable valve for editing strength. In addition, we propose visual prompting to enhance fine-grained visual control. Together, these components form a streaming-generation-style pipeline for fast, controllable, and effective video editing.

Our method has two core components: sampling strategy (Sec. 4.1 and 4.4) and attention manipulation (Sec. 4.2 and 4.3). Sec. 4.1 presents the sampling paradigm. Sec. 4.2 and 4.3 describe the attention designs. Sec. 4.4 introduces sampling guidance, and Sec. 4.5 presents visual prompting for practical control.

4.1 Dual-Branch Fast Sampling

Inversion-based editing methods typically incur high computational cost and cumulative errors due to imperfect inversion, often resulting in poor background preservation [19,30,50,70]. Inversion-free methods, in contrast, progressively edit from the source video over many iterations, but often remain limited in both efficiency and effectiveness [32, 33, 79]. To address these limitations, we reformulate video editing as conditional generation through training-free operations, leveraging the few-step capability of distilled models [61, 65, 84].

To this end, we first extract source-video representations that can be propagated in parallel with target generation during few-step sampling [65]. Specifically, we treat intermediate states from the source sampling process as features that can be synchronously injected into target generation, yielding a dual-branch



(a) Dual-Branch Sampling with Source-Oriented Guidance (b) Self-Attention Bridge of Dual-Branch Generation

Fig. 2: Overview of the framework and components of our proposed StreamEdit. z_t^{tgt} means the one-step prediction at timestep t. x_t^{src} indicates the linear interpolation of x_0^{src} and ϵ_t while x_t^{tgt} denotes it of z_t^{tgt} and ϵ_t. (a) illustrates the generation process and its corresponding samples. We use different colored regions to distinguish the dual branches. Dashed arrow lines depict the stochastic few-step sampling process of target generation. The green arrow $-g_t$ indicates source-oriented guidance, where the latent mask is omitted here. (b) presents self-attention bridge (S.A.B.) for source condition injection and the mask obtained from cross-attention grounding as editing-related regions. The trigger words used for grounding are manually provided by users.

framework. Given a source video with its source prompt and target prompt, we denoise source and target samples in parallel to provide source information for editing. Under stochastic few-step sampling, we formulate the source branch (src) as fixed-endpoint sampling with a constant clean sample:

$$\begin{aligned} x_{t_i}^{\text{src}} &= (1-t_i)x_0^{\text{src}} + t_i\epsilon_{t_i}, \quad x_{t_i}^{\text{tgt}} = (1-t_i)z_{t_i}^{\text{tgt}} + t_i\epsilon_{t_i}, \quad \epsilon_{t_i} \sim \mathcal{N}(0,1), \\ v_{t_i}^{\text{src}} &= v_\theta(x_{t_i}^{\text{src}}, t_i), \quad v_{t_i}^{\text{tgt}} = v_\theta^*(x_{t_i}^{\text{tgt}}, t_i \mid x_{t_i}^{\text{src}}), \quad i \in [1, N], \end{aligned} \tag{3}$$

where v_θ^* is the target velocity (tgt) with the subsequent self-attention bridge and cross-attention boosting. This framework satisfies the basic requirement for injecting source conditions into target generation at every denoising step.

4.2 Self-Attention Bridge of Dual-Branch Generation

Self-attention is central to adaptive visual representation in DiT models [15, 52] and has been widely adopted in editing methods [7, 79, 87]. We therefore build a self-attention bridge between the source and target branches, enabling source-feature-conditioned target generation. Given paired source-target attention features from dual-branch sampling, we design the bridge in three parts: query blending for structure and motion, key blending for editing effectiveness and consistency, and source KV injection for background preservation (Fig. 2 (b)).

Fig. 3: Visualization of the impact of self-attention bridge's components. Query blending ensures basic structural preservation, while key blending keeps continuous editing effects. Source KV injection provides meticulous details for editing-irrelevant regions, with its delay mechanism ($t^{\text{inj}} = 1$) preventing editing failures.

Query Blending: Structure and Motion Preservation. Many attribute-editing tasks require changing object attributes while preserving global motion and trajectory. To enforce this property, we blend source and target self-attention queries, since queries encode spatiotemporal structure and motion cues [55, 67]. Because early denoising steps mainly shape semantics and later steps refine details [47, 66, 80], we use a time-dependent blending ratio:

$$\tilde{Q}_{\text{SA}}^{\text{tgt}} = r_{t_i} Q_{\text{SA}}^{\text{tgt}} + (1 - r_{t_i}) Q_{\text{SA}}^{\text{src}}, \quad r_{t_i} = 1 - t_{i-1}^{\rho}, \tag{4}$$

where Q_{SA} denotes the original self-attention query across layers, r_{t_i} is the blending ratio, and ρ controls the preservation strength. As r_{t_i} increases from 0 to 1, source priors dominate early for structural guidance while later steps retain flexibility for target adaptation.

Key Blending: Editing Effectiveness and Consistency. Query blending alone can still yield unstable edits when query-key matching is weak. To stabilize attention and improve editing consistency, we propose to blend keys jointly with queries. Meanwhile, to together strengthen the matching between queries and previous keys, we further apply similar blending to previous-frame keys. Nevertheless, since source and generated target backgrounds are naturally similar in our framework, we propose to apply this alignment mainly to foreground regions to improve temporal consistency, and thus respectively design our key blending of current keys and masked blending of previous keys:

$$\begin{aligned} \tilde{K}_{\text{SA}}^{\text{tgt}} &= r_{t_i} K_{\text{SA}}^{\text{tgt}} + (1 - r_{t_i}) K_{\text{SA}}^{\text{src}}, \\ \tilde{K}_{\text{prev}}^{\text{tgt}} &= M_{\text{prev}} \odot (r_{t_i} K_{\text{prev}}^{\text{tgt}} + (1 - r_{t_i}) K_{\text{prev}}^{\text{src}}) + (1 - M_{\text{prev}}) \odot K_{\text{prev}}^{\text{tgt}}, \end{aligned} \tag{5}$$

where M_{prev} indicates the union mask of previous video chunks (introduced in Sec. 4.3). As shown in Fig. 3, this design encourages edited regions to aggregate relevant information while reducing interference from unrelated regions, dropping it will lead to editing degradation (w/o Key Blending).

Source KV Injection: Background Preservation. Despite query/key blending, precise background details are still insufficient without explicit source KV injection (Fig. 3, w/o Source KV Injection). However, injecting source KV through-

out all timesteps can suppress target concept emergence (Fig. 3, $t^{\text{inj}} = 1$). Empirically, source KV is most beneficial in low-noise steps, where denoising primarily refines details [47, 66, 80]. The final target-branch self-attention uses:

$$\begin{aligned} Q = \tilde{Q}^{\text{tgt}}_{\text{SA}}, \quad K &= [\tilde{K}^{\text{tgt}}_{\text{SA}}, \tilde{K}^{\text{tgt}}_{\text{prev}}, [t < t^{\text{inj}}] \cdot (M_{\text{curr}} \odot K^{\text{src}}_{\text{SA}})], \\ V &= [V^{\text{tgt}}_{\text{SA}},\ V^{\text{tgt}}_{\text{prev}},\ [t < t^{\text{inj}}] \cdot (M_{\text{curr}} \odot V^{\text{src}}_{\text{SA}})], \end{aligned} \tag{6}$$

where $[t < t^{\text{inj}}]$ is the Iverson bracket; if the condition is false, the corresponding term is excluded from concatenation. t^{inj} denotes the timestep at which source KV injection starts and is fixed to 0.5 in our experiments. M_{curr} denotes the current-chunk mask (introduced in Sec. 4.3). Overall, the full self-attention bridge jointly preserves background and strengthens edit reliability (Fig. 3, Full Self-Attention Bridge).

4.3 Cross-Attention Grounding and Boosting

Grounding: Editing-Related Mask. The self-attention bridge requires reliable editing-region masks, for which we design a cross-attention grounding module without introducing extra models. Given user provided trigger words, prior works typically binarize trigger-word attention maps using fixed thresholds [7, 87], which often require per-model or per-sample tuning. Instead, we use foreground-background attention differences for adaptive grounding. Given the cross-attention vector $A_{\text{CA}}(p, Q_{\text{CA}})$ for word p in prompt $\mathcal{P}$ and query sequence Q_{CA}, we compute the trigger-word mask $\mathcal{T}$ from the difference between averaged attentions over word groups:

$$M^{\phi} = H\left(\frac{\sum_{p \in \mathcal{T}} A^{\phi}_{\text{CA}}(p, Q^{\phi}_{\text{CA}})}{|\mathcal{T}|} - \frac{\sum_{p \in \mathcal{P} \setminus \mathcal{T}} A^{\phi}_{\text{CA}}(p, Q^{\phi}_{\text{CA}})}{|\mathcal{P} \setminus \mathcal{T}|} \right), \tag{7}$$

where $\phi \in \{\text{src}, \text{tgt}\}$ denotes the branch, $H(\cdot)$ is the Heaviside step function, and $|\cdot|$ denotes set cardinality. We obtain M^{src} at $t = 0$ from the clean source latent, leveraging the distilled model's ability to retain spatial structure across timesteps. For the target branch, M^{tgt} is extracted at $t = t^{\text{inj}}$, after target semantics have formed. We then compute the union mask $M = \cup_{\phi} M^{\phi}$. The union mask covers both source regions to be modified and target regions requiring editing. Additionally, we denote the current-chunk mask as M_{curr} and the masks from previous chunks as M_{prev}.

Boosting: Regional Editing Enhancement. To provide a controllable valve for editing strength, we design cross-attention boosting based on the role of cross-attention in controlling prompt-word expression [8, 17]. Multiplicative score enhancement can over-polarize token expression [22, 86]; moreover, negative scores may unintentionally suppress trigger concepts. Using masks from grounding, we therefore apply additive regional enhancement to trigger words inside editing

regions, promoting edits while protecting background regions:

$$A_{\mathrm{CA}}^{\mathrm{tgt}}(p,q) = \frac{\exp\left(S(p,q) + \ln w_{p,q}\right)}{\sum_{j \in Q_{\mathrm{CA}}^{\mathrm{tgt}}} \exp\left(S(j,q) + \ln w_{j,q}\right)}, \\ w_{p,q} = \begin{cases} \omega, & \text{if } p \in \mathcal{T} \text{ and } M(q) = 1 \\ 1, & \text{otherwise} \end{cases}, \tag{8}$$

where $S(p,q)$ is the pre-softmax attention score between text token p and visual query token q, and ω controls editing strength. $M(q) \in \{0,1\}$ denotes the mask value of query token q; we use $M_{\mathrm{curr}}^{\mathrm{src}}$ in high-noise steps and M_{curr} when $t < t^{\mathrm{inj}}$. This regional boosting improves controllability of edit intensity while reducing unintended background changes.

4.4 Source-Oriented Guidance for Visual Quality

The self-attention bridge provides stable source-condition injection, while stochastic sampling improves editing flexibility and feasibility [58,62]. However, stochasticity also makes the source branch a mixed trajectory of source video and random noise [43,45], which introduces noisy source conditions and causes undesirable jitter in editing-irrelevant target regions, even with the self-attention bridge (Fig. 6 (b)). This error is observable through the difference between the predicted source velocity and the ground-truth velocity of the corresponding linear interpolation:

$$v_{t_i}^{\mathrm{gt}} = \epsilon_{t_i} - x_0^{\mathrm{src}}, \quad g_{t_i} = v_{t_i}^{\mathrm{gt}} - v_{t_i}^{\mathrm{src}}, \tag{9}$$

where g_{t_i} denotes the observable error. Since randomness in editing-related regions remains beneficial for effective edits, we mainly apply correction to editing-irrelevant regions via a mask coefficient:

$$\tilde{v}_{t_i}^{\mathrm{tgt}} = v_{t_i}^{\mathrm{tgt}} + \mathrm{AMN}(v_{t_i}^{\mathrm{tgt}} - v_{t_i}^{\mathrm{src}}) \odot g_{t_i}, \quad z_{t_{i-1}}^{\mathrm{tgt}} = x_{t_i}^{\mathrm{tgt}} - t_i \tilde{v}_{t_i}^{\mathrm{tgt}}, \tag{10}$$

where $\odot$ is the element-wise product and $\mathrm{AMN}(\cdot)$ denotes the Abs-Mean-Norm operation (get the channel-wise mean of the input's absolute value, and then perform a min-max normalization on the mean to obtain a soft mask). Because both branches share the same random noise, source-branch prediction errors relative to ground truth can be projected to the target branch, while inter-branch velocity differences also provide latent-level cues of editing-related regions. The overall process is illustrated in Fig. 2 (a), where translucent dashed lines denote linear interpolation and bold dashed arrows indicate the target generation.

4.5 Visual Prompting for Autoregressive Editing

Text-driven editing typically lacks fine-grained visual control. The outcome depends on difficult-to-fine-tune natural language prompts and how well the model understands them, yet success is not well ensured. Controllability and success rate are both critical aspects in video editing [39]. Since one image is usually

easily adjustable by users, we propose regarding the user-provided first frame as a visual prompt by treating it as a previous chunk of the generated video, thereby easily incorporating detailed editing effects users applied to the provided frame into the target generation. Meanwhile, this approach also enables video editing to leverage existing effective image editing models [36, 75] by using the model-edited first frame as a visual prompt, thus ensuring easier success. Such a simple design requires only one additional forward of the video generation model to provide visual prompts, easily making the video editing more desirable.

5 Experiments

5.1 Experimental Setups

Implementation. We build StreamEdit on two pre-trained streaming video generation models: Self Forcing (SF) [24] and LongLive (LL) [81]. We evaluate both text-driven video editing (Ours (SF), Ours (LL)) and image-prompted text-driven editing (Ours (SF)§, Ours (LL)§), where first-frame visual prompts are generated by Qwen-Image-Edit [75]. For both models, we apply the self-attention bridge and cross-attention boosting to all attention layers, and perform grounding on the first 20 of 30 cross-attention layers. We fix $\rho = 2$ in all evaluations, and set $\omega = 4$ for Ours (SF) and $\omega = 2$ for the other settings. We use 15 steps for standard comparisons and 5 steps for long-video streaming editing. A uniform timestep scheduler is adopted. Classifier-free guidance is disabled; therefore, the number of function evaluations is $2 \times$ (steps $+ 1$), consistent with two-sample generation. All experiments are conducted on a single NVIDIA A100 GPU.

Baselines. We focus on text-driven video editing and compare against diverse state-of-the-art methods: image-model-based (TokenFlow [20], VidToMe [41], VideoGrain [82]), video-diffusion-based (DMT [83]), and flow-matching-based (Pyramid-Edit [39], Wan-Edit [39], UniEdit-Flow [28]) methods. For long-video settings, we additionally compare with StreamV2V [42] and AdaFlow [91].

Benchmark and Metrics. We primarily evaluate on FiVE-Bench [39], a fine-grained video editing benchmark with 100 collected videos and 420 editing prompt pairs across six edit types: color alteration, material modification, object substitution with/without non-rigid deformation, add, and remove. FiVE-Bench reports five metric groups: structure/background preservation (structure distance [68], PSNR, LPIPS [90], MSE, SSIM [72]), text alignment (CLIP score [56, 74], edit-region CLIP score), image quality (NIQE [59]), temporal consistency (motion fidelity [83]), and VLM-based editing success (FiVE-Acc). For baselines already evaluated on FiVE-Bench, we report official numbers obtained on one H100 GPU. For UniEdit-Flow, we run evaluation using its official implementation. For long-video evaluation, we collect 30-second videos at 16 FPS from the internet and build 21 editing cases for user study. The user study evaluates three aspects: overall video quality, background consistency, and editing fidelity. We report method rankings for each aspect.

Table 1: Comparison of video editing methods on FiVE-Bench. * and † denote methods that require optimization and depth/segmentation maps, respectively. § indicates methods that using pre-provided edited first frame as additional input (the first frame gathering time is not included in the Time Per Frame). The best and second best results are **bolded** and underlined, with cells highlighted from worse to better.

Methods	Struc.	Background Preservation				Text Alignment		IQA	Temp.	FiVE	Time (s)
	Dist.$_{\times 10^3}$↓	PSNR↑	LPIPS$_{\times 10^3}$↓	MSE$_{\times 10^4}$↓	SSIM$_{\times 10^2}$↑	CLIPS.↑	CLIPS.$_{\text{edit}}$↑	NIQE↓	C.$_{\times 10^2}$↑	Acc↑	Per Frame↓
TokenFlow [20]	35.62	19.06	263.61	138.65	72.51	26.46	21.15	**4.01**	89.00	27.43	8.04
DMT* [83]	85.95	14.71	404.60	372.78	51.64	26.66	21.44	5.24	82.30	48.42	25.98
VidToMe [41]	22.37	21.15	263.91	88.75	70.69	26.84	21.05	4.68	90.06	26.77	3.25
VideoGrain† [82]	12.40	27.05	185.21	25.10	79.13	25.69	20.31	4.08	88.57	37.23	27.12
Pyramid-Edit [39]	28.65	20.84	276.59	95.63	71.72	26.82	20.20	5.48	80.59	43.84	1.44
Wan-Edit [39]	12.53	25.57	94.61	41.84	82.55	26.39	21.23	6.54	89.43	46.97	3.07
UniEdit-Flow [28]	18.31	24.43	223.07	45.23	78.60	26.06	20.92	4.30	88.57	50.10	1.11
Ours (SF)	**10.27**	**28.47**	**49.84**	**21.88**	**87.52**	26.95	21.73	4.38	**90.93**	51.94	**0.60**
Ours (LL)	15.77	25.43	65.35	44.36	84.15	**27.32**	**22.13**	4.41	89.47	55.04	0.69
Ours (SF)§	13.09	26.85	65.32	37.58	85.36	26.99	21.64	4.57	89.91	58.64	0.71§
Ours (LL)§	15.69	25.74	67.04	40.70	84.29	27.22	22.11	4.61	89.12	**61.19**	0.76§

5.2 Comparison with State-of-the-Art Methods

Quantitative Results. Tab. 1 reports quantitative comparisons on FiVE-Bench. Compared with prior methods, StreamEdit—both text-driven and visual-prompted variants—achieves strong gains in text alignment while better preserving structure and context. Superior LPIPS and SSIM indicate improved photorealism and structural consistency. The combined results on image quality, motion fidelity, and editing accuracy further show that StreamEdit is both effective and reliable. Notably, StreamEdit performs strongly on both Self Forcing and LongLive, demonstrating good generalizability. Self Forcing favors structural/context preservation, while LongLive provides stronger editing effects, offering flexible choices for different user preferences. Overall, StreamEdit delivers consistently better results with minimal time cost, marking a clear advance over existing methods.
Qualitative Results. Fig. 4 compares StreamEdit with prior methods. On challenging cases, advanced flow-based editors (Pyramid-Edit, Wan-Edit, UniEdit-Flow) often fail, especially in color edits (*e.g.*, 4th column), and may produce low-quality, unfinished-looking videos. In contrast, StreamEdit achieves the intended edits more reliably (*e.g.*, 1st and 2nd columns). Methods based on video diffusion or image models also struggle to balance editing success with background preservation. StreamEdit maintains stronger structural/background consistency with the source video while preserving editing fidelity (*e.g.*, 3rd and 4th columns).

5.3 Ablation Study

Main Components. As shown in Tab. 2, we ablate the two main components of StreamEdit on Self Forcing: source-oriented guidance in stochastic dual-branch sampling (S.O.G.) and the self-attention bridge (S.A.B.). Source-oriented guidance compensates stochastic errors; removing it may slightly increase edit freedom but notably weakens stable background alignment with the source video

Fig. 4: Qualitative comparisons. We show text-only StreamEdit with Self Forcing (Ours (SF)) and image-conditioned StreamEdit with LongLive (Ours (LL)[§]), demonstrating clear advantages over prior state-of-the-art methods.

(2nd row in Fig. 6 (b)). The self-attention bridge is the key mechanism for injecting source conditions into target generation. Without it, generation becomes weakly constrained by the source and drifts toward the target prompt (2nd row in Tab. 2), producing outputs largely unrelated to the source video (3rd row in Fig. 6 (b)). Together, these two components convert stochastic generation into controllable video-conditioned generation, enabling fast and effective video editing (4th row in Fig. 6 (b)).

Hyper-parameter Efficacy. The two tunable hyper-parameters, the self-attention blending ratio ρ and the cross-attention boosting scale ω, act as control knobs for structural preservation and editing strength, respectively. The left panel of Fig. 6 (c) shows the role of ρ: smaller ρ improves structural consistency but can weaken edits (no jeep at $\rho = 1$), whereas larger ρ allows more flexible edits and better handles complex cases (successful jeep at $\rho = 3$). The right panel of Fig. 6 (c) shows that larger ω increases edit intensity (*e.g.*, the elephant becomes bluer as ω increases). Fig. 5 further shows that tuning these hyper-parameters yields clear trade-off curves, demonstrating robustness and controllability.

Visual Prompting for video editing. To further illustrate visual prompting, we provide representative examples in Fig. 6 (a). Its first role is fine-grained visual control over the editing target. As shown in the left example, text-only

Table 2: Quantitative ablation of the proposed main components. S.O.G. indicates the source-oriented guidance and S.A.B. denotes the self-attention bridge.

Methods	Background Preservation			Editing Effectiveness		
	PSNR↑	LPIPS$_{\times 10^3}$↓	SSIM$_{\times 10^2}$↑	CLIPS.↑	CLIPS.$_{\text{edit}}$ ↑	FiVE$_{\text{Acc}}$ ↑
w/o S.O.G. (SF)	21.86	122.43	72.40	27.13	21.99	53.58
w/o S.A.B. (SF)	13.42	381.19	47.70	27.46	21.67	67.74
Ours (SF)	28.47	49.84	87.52	26.95	21.73	51.94

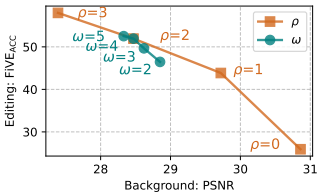
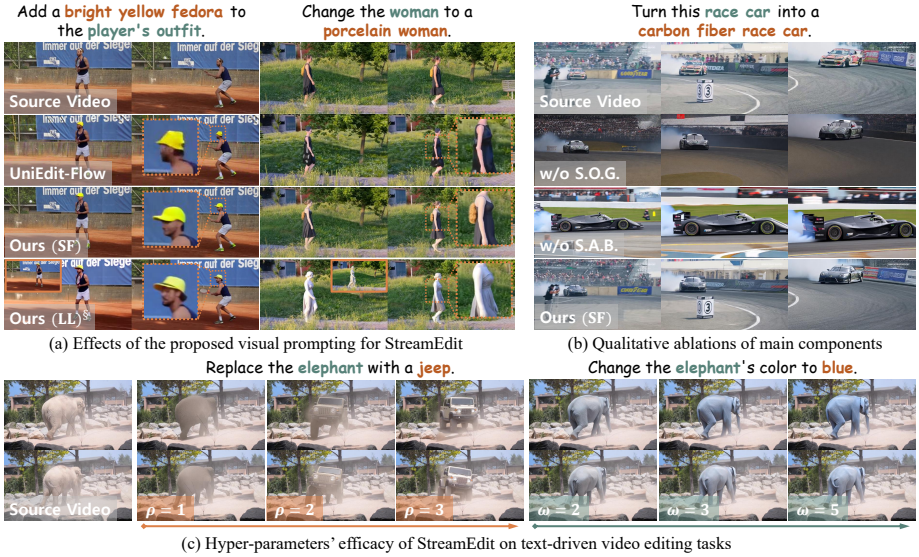


Fig. 5: Trade-off of ρ and ω.



(a) Effects of the proposed visual prompting for StreamEdit

(b) Qualitative ablations of main components

(c) Hyper-parameters' efficacy of StreamEdit on text-driven video editing tasks

Fig. 6: Qualitative ablation studies of the proposed StreamEdit.

editing can be ambiguous across methods/models: for a target fedora, the model may generate a round cap or baseball cap instead. With an explicit visual prompt (orange solid box), StreamEdit follows the intended target much more reliably. Its second role is to support difficult edits. The right example shows a challenging transformation of a woman into a porcelain figure. Without visual prompting, methods often only smooth the skin. With a specific visual target, StreamEdit successfully realizes the intended edit.

5.4 Analyses of the Editing Process

Fig. 7 visualizes the editing process of our proposed StreamEdit. Few-step generation first yields target semantics and then gradually refines them. Our approach works in a similar way. Thus, benefiting from the generation-style paradigm, the target concept rapidly takes shape in the early stages (z_t^{tgt} $(t > t^{\text{inj}})$). After injecting source KV, video details are gradually refined towards precise editing (z_t^{tgt} $(t < t^{\text{inj}})$), and finally yield stable and satisfactory results (z_0^{tgt}).

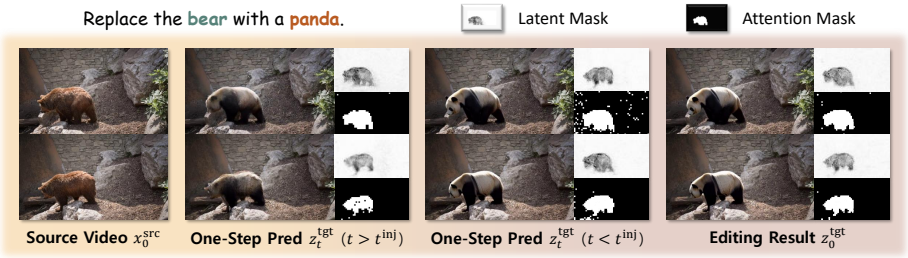


Fig. 7: Visualization of the editing process. We present the editing process from the source video to one-step predictions at $t > t^{\text{inj}}$ and $t < t^{\text{inj}}$ and finally to the editing result. We visualize the corresponding latent mask and attention mask at the upperright and lower right of each frame. Latent masks dynamically update using current timestep's velocity predictions. Attention masks change from $M_{\text{curr}}^{\text{src}}$ to the union mask M_{curr} then to the stored mask M_{prev} in the figure.

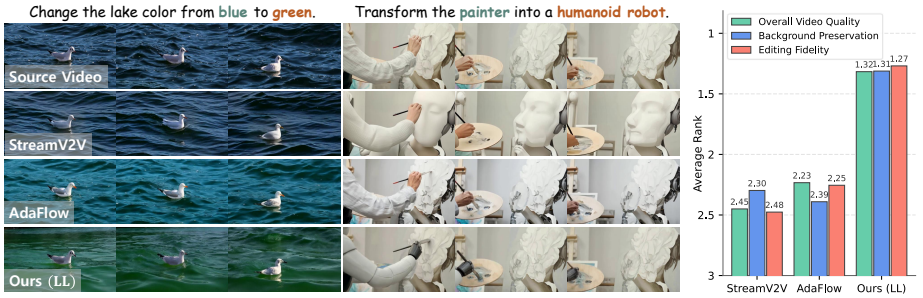
In terms of the masks, the upper right of each frame shows the latent masks that are dynamically updated at each timestep to fit the current target concepts. Since the denoising process performs differently at different stage [80], this continuous updating ensures more stable and smoother results. The lower right of each frame indicates its corresponding attention mask. Attention masks are updated threefold. Before denoising, we first forward the source clean data to obtain the source mask $M_{\text{curr}}^{\text{src}}$ while caching the source KV as the next-chunk previous condition. At $t = t^{\text{inj}}$, we extract the target mask $M_{\text{curr}}^{\text{tgt}}$ during denoising, and then obtain the union mask M_{curr}. Finally after denoising, we forward the target clean data for gaining KV cache while extract a new target mask which will merge with $M_{\text{curr}}^{\text{src}}$ to be the previous mask M_{prev} for subsquent chunks. With this design, our masks align accurately with the editing-related regions and require no additional NFEs, thus facilitating precise editing without notable extra cost.

5.5 Autoregressive Editing for Long Videos

Fig. 8 compares StreamEdit with existing long-video editing methods [42, 91]. We conduct a user study with 20 participants on long-video editing, evaluating overall video quality, background preservation, and editing fidelity. Our method (Ours (LL), 5 steps) ranks first across all criteria, demonstrating strong effectiveness and reliability for text-driven long-video editing.

6 Conclusion

We reconsider video editing as a source-conditioned noise-to-data streaming generation problem, moving beyond the conventional data-to-data paradigm. Based on this perspective, we propose StreamEdit, which addresses key limitations of existing methods on complex videos, including heavy iterative cost, limited generation quality, and unstable editing performance. Extensive experiments



(a) Quantitative comparison between methods of long video editing tasks (b) Long video editing user study

Fig. 8: Comparisons of long video editing using videos with over 470 frames.

on text-driven and image-prompted, short- and long-video editing demonstrate the superiority of this paradigm in both effectiveness and efficiency. In future work, we will further improve scalability and real-time capability, and extend the framework to broader video editing scenarios.

Acknowledgements

This work was supported, in part, by the NSERC DG Grant (No. RGPIN-2022-04636), the NSERC Alliance Grant (ALLRP 604621-25), the Vector Institute for AI, and the Canada CIFAR AI Chair program, and a gift fund from Snap Inc. Resources used in preparing this research were provided, in part, by the Province of Ontario, the Government of Canada through the Digital Research Alliance of Canada `alliance.can.ca`, and companies sponsoring the Vector Institute `www.vectorinstitute.ai/#partners`, and Advanced Research Computing at the University of British Columbia. Additional resource was provided by the Canada Foundation for Innovation (CFI) via the John R. Evans Leaders Fund (JELF). G.J. is supported by a UBC Four Year Doctoral Fellowship.

References

1. Albergo, M.S., Boffi, N.M., Vanden-Eijnden, E.: Stochastic interpolants: A unifying framework for flows and diffusions. arXiv preprint arXiv:2303.08797 (2023)
2. Avrahami, O., Lischinski, D., Fried, O.: Blended diffusion for text-driven editing of natural images. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 18208–18218 (2022)
3. Bai, J., He, T., Wang, Y., Guo, J., Hu, H., Liu, Z., Bian, J.: Uniedit: A unified tuning-free framework for video motion and appearance editing. In: Proceedings of the 33rd ACM International Conference on Multimedia. pp. 10171–10180 (2025)
4. Bar-On, R., Cohen-Bar, D., Cohen-Or, D.: Editp23: 3d editing via propagation of image prompts to multi-view. arXiv preprint arXiv:2506.20652 (2025)
5. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 18392–18402 (2023)

6. Cai, M., Cun, X., Li, X., Liu, W., Zhang, Z., Zhang, Y., Shan, Y., Yue, X.: Ditctrl: Exploring attention control in multi-modal diffusion transformer for tuning-free multi-prompt longer video generation. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 7763–7772 (2025)
7. Cao, M., Wang, X., Qi, Z., Shan, Y., Qie, X., Zheng, Y.: Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 22560–22570 (October 2023)
8. Chefer, H., Alaluf, Y., Vinker, Y., Wolf, L., Cohen-Or, D.: Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. ACM transactions on Graphics (TOG) **42**(4), 1–10 (2023)
9. Chen, B., Martí Monsó, D., Du, Y., Simchowitz, M., Tedrake, R., Sitzmann, V.: Diffusion forcing: Next-token prediction meets full-sequence diffusion. Advances in Neural Information Processing Systems **37**, 24081–24125 (2024)
10. Cheng, J., Xiao, T., He, T.: Consistent video-to-video transfer using synthetic dataset. arXiv preprint arXiv:2311.00213 (2023)
11. Cui, J., Wu, J., Li, M., Yang, T., Li, X., Wang, R., Bai, A., Ban, Y., Hsieh, C.J.: Self-forcing++: Towards minute-scale high-quality video generation. arXiv preprint arXiv:2510.02283 (2025)
12. Deng, Y., He, X., Mei, C., Wang, P., Tang, F.: Fireflow: Fast inversion of rectified flow for image semantic editing (2024), https://arxiv.org/abs/2412.07517
13. Dong, W., Xue, S., Duan, X., Han, S.: Prompt tuning inversion for text-driven image editing using diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7430–7440 (2023)
14. Dong, Z., Wang, X., Zhu, Z., Wang, Y., Wang, Y., Zhou, Y., Wang, B., Ni, C., Ouyang, R., Qin, W., et al.: Emma: Generalizing real-world robot manipulation via generative visual transfer. arXiv preprint arXiv:2509.22407 (2025)
15. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
16. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al.: Scaling rectified flow transformers for high-resolution image synthesis. In: Forty-first international conference on machine learning (2024)
17. Feng, W., He, X., Fu, T.J., Jampani, V., Akula, A., Narayana, P., Basu, S., Wang, X.E., Wang, W.Y.: Training-free structured diffusion guidance for compositional text-to-image synthesis. arXiv preprint arXiv:2212.05032 (2022)
18. Gao, W., Fan, J., Zeng, J., Yang, S.: Flowportal: Residual-corrected flow for training-free video relighting and background replacement. arXiv preprint arXiv:2511.18346 (2025)
19. Garibi, D., Patashnik, O., Voynov, A., Averbuch-Elor, H., Cohen-Or, D.: Renoise: Real image inversion through iterative noising (2024)
20. Geyer, M., Bar-Tal, O., Bagon, S., Dekel, T.: Tokenflow: Consistent diffusion features for consistent video editing. In: The Twelfth International Conference on Learning Representations (2024)
21. Guo, Y., Yang, C., Yang, Z., Ma, Z., Lin, Z., Yang, Z., Lin, D., Jiang, L.: Long context tuning for video generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 17281–17291 (2025)

22. Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., Cohen-Or, D.: Prompt-to-prompt image editing with cross attention control. arXiv preprint arXiv:2208.01626 (2022)
23. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020)
24. Huang, X., Li, Z., He, G., Zhou, M., Shechtman, E.: Self forcing: Bridging the train-test gap in autoregressive video diffusion. arXiv preprint arXiv:2506.08009 (2025)
25. Jeong, H., Lee, S., Ye, J.C.: Reangle-a-video: 4d video generation as video-to-video translation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11164–11175 (2025)
26. Ji, S., Chen, X., Yang, S., Tao, X., Wan, P., Zhao, H.: Memflow: Flowing adaptive memory for consistent and efficient long video narratives. arXiv preprint arXiv:2512.14699 (2025)
27. Jiang, Z., Han, Z., Mao, C., Zhang, J., Pan, Y., Liu, Y.: Vace: All-in-one video creation and editing. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 17191–17202 (2025)
28. Jiao, G., Huang, B., Wang, K.C., Liao, R.: Uniedit-flow: Unleashing inversion and editing in the era of flow models. arXiv preprint arXiv:2504.13109 (2025)
29. Jin, Y., Sun, Z., Li, N., Xu, K., Jiang, H., Zhuang, N., Huang, Q., Song, Y., Mu, Y., Lin, Z.: Pyramidal flow matching for efficient video generative modeling. arXiv preprint arXiv:2410.05954 (2024)
30. Ju, X., Zeng, A., Bian, Y., Liu, S., Xu, Q.: Pnp inversion: Boosting diffusion-based editing with 3 lines of code. International Conference on Learning Representations (ICLR) (2024)
31. Kara, O., Kurtkaya, B., Yesiltepe, H., Rehg, J.M., Yanardag, P.: Rave: Randomized noise shuffling for fast and consistent video editing with diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6507–6516 (2024)
32. Kim, J., Hong, Y., Park, J., Ye, J.C.: Flowalign: Trajectory-regularized, inversion-free flow-based image editing. arXiv preprint arXiv:2505.23145 (2025)
33. Kulikov, V., Kleiner, M., Huberman-Spiegelglas, I., Michaeli, T.: Flowedit: Inversion-free text-based editing using pre-trained flow models. arXiv preprint arXiv:2412.08629 (2024)
34. Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C.H., Gonzalez, J., Zhang, H., Stoica, I.: Efficient memory management for large language model serving with pagedattention. In: Proceedings of the 29th symposium on operating systems principles. pp. 611–626 (2023)
35. Labs, B.F.: Flux. `https://github.com/black-forest-labs/flux` (2024)
36. Labs, B.F., Batifol, S., Blattmann, A., Boesel, F., Consul, S., Diagne, C., Dockhorn, T., English, J., English, Z., Esser, P., Kulal, S., Lacey, K., Levi, Y., Li, C., Lorenz, D., Müller, J., Podell, D., Rombach, R., Saini, H., Sauer, A., Smith, L.: Flux.1 kontext: Flow matching for in-context image generation and editing in latent space (2025), `https://arxiv.org/abs/2506.15742`
37. Li, G., Yang, Y., Song, C., Zhang, C.: Flowdirector: Training-free flow steering for precise text-to-video editing. arXiv preprint arXiv:2506.05046 (2025)
38. Li, H., Zhang, I., Ouyang, R., Wang, X., Zhu, Z., Yang, Z., Zhang, Z., Wang, B., Ni, C., Qin, W., et al.: Mimicdreamer: Aligning human and robot demonstrations for scalable vla training. arXiv preprint arXiv:2509.22199 (2025)

39. Li, M., Xie, C., Wu, Y., Zhang, L., Wang, M.: Five: A fine-grained video editing benchmark for evaluating emerging diffusion and rectified flow models. arXiv preprint arXiv:2503.13684 (2025)
40. Li, W., Pan, W., Luan, P.C., Gao, Y., Alahi, A.: Stable video infinity: Infinite-length video generation with error recycling. arXiv preprint arXiv:2510.09212 (2025)
41. Li, X., Ma, C., Yang, X., Yang, M.H.: Vidtome: Video token merging for zero-shot video editing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7486–7495 (2024)
42. Liang, F., Kodaira, A., Xu, C., Tomizuka, M., Keutzer, K., Marculescu, D.: Looking backward: Streaming video-to-video translation with feature banks. arXiv preprint arXiv:2405.15757 (2024)
43. Lipman, Y., Chen, R.T., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. arXiv preprint arXiv:2210.02747 (2022)
44. Liu, S., Zhang, Y., Li, W., Lin, Z., Jia, J.: Video-p2p: Video editing with cross-attention control. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8599–8608 (2024)
45. Liu, X., Gong, C., Liu, Q.: Flow straight and fast: Learning to generate and transfer data with rectified flow. arXiv preprint arXiv:2209.03003 (2022)
46. Liu, X., Zhang, X., Ma, J., Peng, J., et al.: Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In: The Twelfth International Conference on Learning Representations (2023)
47. Luo, G., Dunlap, L., Park, D.H., Holynski, A., Darrell, T.: Diffusion hyperfeatures: Searching through time and space for semantic correspondence. Advances in Neural Information Processing Systems **36**, 47500–47510 (2023)
48. Luo, S., Tan, Y., Huang, L., Li, J., Zhao, H.: Latent consistency models: Synthesizing high-resolution images with few-step inference. arXiv preprint arXiv:2310.04378 (2023)
49. Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J.Y., Ermon, S.: SDEdit: Guided image synthesis and editing with stochastic differential equations. In: International Conference on Learning Representations (2022)
50. Mokady, R., Hertz, A., Aberman, K., Pritch, Y., Cohen-Or, D.: Null-text inversion for editing real images using guided diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6038–6047 (2023)
51. Ouyang, Z., Zheng, D., Wu, X.M., Jiang, J.J., Lin, K.Y., Meng, J., Zheng, W.S.: Proedit: Inversion-based editing from prompts done right. arXiv preprint arXiv:2512.22118 (2025)
52. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4195–4205 (2023)
53. Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., Rombach, R.: Sdxl: Improving latent diffusion models for high-resolution image synthesis. arXiv preprint arXiv:2307.01952 (2023)
54. Polyak, A., Zohar, A., Brown, A., Tjandra, A., Sinha, A., Lee, A., Vyas, A., Shi, B., Ma, C.Y., Chuang, C.Y., et al.: Movie gen: A cast of media foundation models. arXiv preprint arXiv:2410.13720 (2024)
55. Qi, C., Cun, X., Zhang, Y., Lei, C., Wang, X., Shan, Y., Chen, Q.: Fatezero: Fusing attentions for zero-shot text-based video editing. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15932–15942 (2023)

56. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
57. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
58. Rout, L., Chen, Y., Ruiz, N., Caramanis, C., Shakkottai, S., Chu, W.S.: Semantic image inversion and editing using rectified stochastic differential equations. arXiv preprint arXiv:2410.10792 (2024)
59. Saad, M.A., Bovik, A.C.: Blind quality assessment of videos using a model of natural scene statistics and motion coherency. In: 2012 Conference Record of the Forty Sixth Asilomar Conference on Signals, Systems and Computers (ASILOMAR). pp. 332–336. IEEE (2012)
60. Sabour, A., Fidler, S., Kreis, K.: Align your flow: Scaling continuous-time flow map distillation (2025)
61. Sauer, A., Lorenz, D., Blattmann, A., Rombach, R.: Adversarial diffusion distillation. In: European Conference on Computer Vision. pp. 87–103. Springer (2024)
62. Singh, S., Fischer, I.: Stochastic sampling from deterministic flow models. arXiv preprint arXiv:2410.02217 (2024)
63. Song, C., Yang, Y., Zhao, T., Li, R., Zhang, C.: Worldforge: Unlocking emergent 3d/4d generation in video diffusion model via training-free guidance. arXiv preprint arXiv:2509.15130 (2025)
64. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502 (2020)
65. Song, Y., Dhariwal, P., Chen, M., Sutskever, I.: Consistency models. arXiv preprint arXiv:2303.01469 (2023)
66. Tinaz, B., Fabian, Z., Soltanolkotabi, M.: Emergence and evolution of interpretable concepts in diffusion models (2025)
67. Tu, S., Dai, Q., Cheng, Z.Q., Hu, H., Han, X., Wu, Z., Jiang, Y.G.: Motioneditor: Editing video motion via content-aware diffusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7882–7891 (2024)
68. Tumanyan, N., Bar-Tal, O., Bagon, S., Dekel, T.: Splicing vit features for semantic appearance transfer. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10748–10757 (2022)
69. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
70. Wang, J., Pu, J., Qi, Z., Guo, J., Ma, Y., Huang, N., Chen, Y., Li, X., Shan, Y.: Taming rectified flow for inversion and editing. arXiv preprint arXiv:2411.04746 (2024)
71. Wang, Y., Wang, L., Ma, Z., Hu, Q., Xu, K., Guo, Y.: Videodirector: Precise video editing via text-to-video models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2589–2598 (2025)
72. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004)
73. WanTeam, Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.W., Chen, D., Yu, F., Zhao, H., Yang, J., Zeng, J., Wang, J., Zhang, J., Zhou, J., Wang, J., Chen, J., Zhu, K.,

Zhao, K., Yan, K., Huang, L., Feng, M., Zhang, N., Li, P., Wu, P., Chu, R., Feng, R., Zhang, S., Sun, S., Fang, T., Wang, T., Gui, T., Weng, T., Shen, T., Lin, W., Wang, W., Wang, W., Zhou, W., Wang, W., Shen, W., Yu, W., Shi, X., Huang, X., Xu, X., Kou, Y., Lv, Y., Li, Y., Liu, Y., Wang, Y., Zhang, Y., Huang, Y., Li, Y., Wu, Y., Liu, Y., Pan, Y., Zheng, Y., Hong, Y., Shi, Y., Feng, Y., Jiang, Z., Han, Z., Wu, Z.F., Liu, Z.: Wan: Open and advanced large-scale video generative models. arXiv preprint arXiv:2503.20314 (2025)

74. Wu, C., Huang, L., Zhang, Q., Li, B., Ji, L., Yang, F., Sapiro, G., Duan, N.: Godiva: Generating open-domain videos from natural descriptions. arXiv preprint arXiv:2104.14806 (2021)
75. Wu, C., Li, J., Zhou, J., Lin, J., Gao, K., Yan, K., ming Yin, S., Bai, S., Xu, X., Chen, Y., Chen, Y., Tang, Z., Zhang, Z., Wang, Z., Yang, A., Yu, B., Cheng, C., Liu, D., Li, D., Zhang, H., Meng, H., Wei, H., Ni, J., Chen, K., Cao, K., Peng, L., Qu, L., Wu, M., Wang, P., Yu, S., Wen, T., Feng, W., Xu, X., Wang, Y., Zhang, Y., Zhu, Y., Wu, Y., Cai, Y., Liu, Z.: Qwen-image technical report (2025), https://arxiv.org/abs/2508.02324
76. Wu, Y., Chen, L., Li, R., Wang, S., Xie, C., Zhang, L.: Insvie-1m: Effective instruction-based video editing with elaborate dataset construction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16692–16701 (2025)
77. Xiang, J., Lv, Z., Xu, S., Deng, Y., Wang, R., Zhang, B., Chen, D., Tong, X., Yang, J.: Structured 3d latents for scalable and versatile 3d generation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 21469–21480 (2025)
78. Xie, D., Xu, Z., Hong, Y., Tan, H., Liu, D., Liu, F., Kaufman, A., Zhou, Y.: Progressive autoregressive video diffusion models. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 6322–6332 (2025)
79. Xu, S., Huang, Y., Pan, J., Ma, Z., Chai, J.: Inversion-free image editing with language-guided diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9452–9461 (June 2024)
80. Yang, D., Zhang, Y., Yu, X., Hou, L., Tao, X., Wan, P., Qi, X., Liao, R.: Stable velocity: A variance perspective on flow matching. arXiv preprint arXiv:2602.05435 (2026)
81. Yang, S., Huang, W., Chu, R., Xiao, Y., Zhao, Y., Wang, X., Li, M., Xie, E., Chen, Y., Lu, Y., et al.: Longlive: Real-time interactive long video generation. arXiv preprint arXiv:2509.22622 (2025)
82. Yang, X., Zhu, L., Fan, H., Yang, Y.: Videograin: Modulating space-time attention for multi-grained video editing. arXiv preprint arXiv:2502.17258 (2025)
83. Yatim, D., Fridman, R., Bar-Tal, O., Kasten, Y., Dekel, T.: 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)
84. Yin, T., Gharbi, M., Zhang, R., Shechtman, E., Durand, F., Freeman, W.T., Park, T.: One-step diffusion with distribution matching distillation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6613–6623 (2024)
85. Yin, T., Zhang, Q., Zhang, R., Freeman, W.T., Durand, F., Shechtman, E., Huang, X.: 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)

86. Yin, X., Li, Z., Zhang, J., Li, C., Zhang, Y.: Coloredit: Training-free image-guided color editing with diffusion model. arXiv preprint arXiv:2411.10232 (2024)
87. Yin, Z., Chen, L.H., Ni, L., Dai, X.: Consistedit: Highly consistent and precise training-free visual editing. In: Proceedings of the SIGGRAPH Asia 2025 Conference Papers. pp. 1–11 (2025)
88. Zhang, K., Mo, L., Chen, W., Sun, H., Su, Y.: Magicbrush: A manually annotated dataset for instruction-guided image editing. In: Advances in Neural Information Processing Systems (2023)
89. Zhang, L., Agrawala, M.: Packing input frame context in next-frame prediction models for video generation. arXiv e-prints pp. arXiv–2504 (2025)
90. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)
91. Zhang, S., Liu, Y., Zhou, H., Peng, J., Zhou, Y., Sun, X., Ji, R.: Adaflow: Efficient long video editing via adaptive attention slimming and keyframe selection. arXiv preprint arXiv:2502.05433 (2025)
92. Zhang, S., Yang, H., Lim, S.N.: Videomerge: Towards training-free long video generation. arXiv preprint arXiv:2503.09926 (2025)
93. Zhou, Z., Ma, F., Gui, C., Xia, X., Fan, H., Yang, Y., Chua, T.S.: Anchorflow: Training-free 3d editing via latent anchor-aligned flows. arXiv preprint arXiv:2511.22357 (2025)
94. Zi, B., Ruan, P., Chen, M., Qi, X., Hao, S., Zhao, S., Huang, Y., Liang, B., Xiao, R., Wong, K.F.: Señorita-2m: A high-quality instruction-based dataset for general video editing by video specialists. arXiv preprint arXiv:2502.06734 (2025)