VINCIE-NExT: Unlocking Video Editing from Images via In-Context Modeling

Leigang Qu[1] **Feng Cheng**[2] **Ziyan Yang**[2] **Bangbang Yang**[2] **Zhaoyang Huang**[2]
Wei Chow[1] **Yicong Li**[3] **Wenjie Wang**[3] **Tat-Seng Chua**[1] **Yan Zeng**[2]

[1]National University of Singapore [2]ByteDance Seed
[3]University of Science and Technology of China

Project Page: https://vincie-next.github.io/

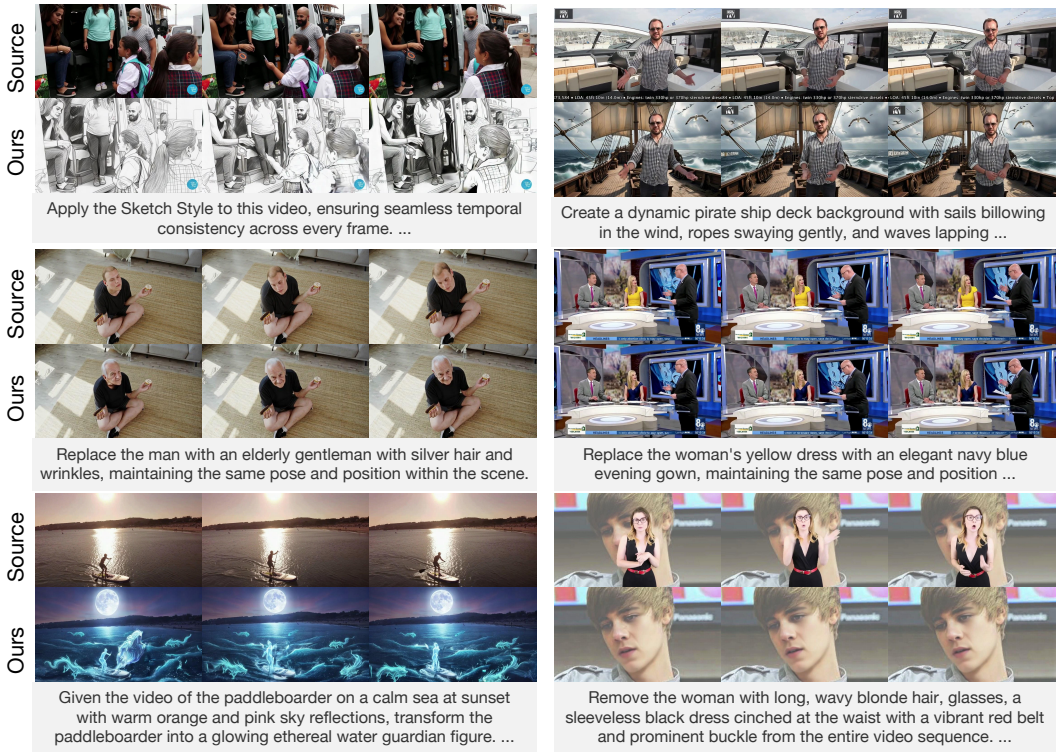


Figure 1: **Qualitative showcase of VINCIE-NExT across diverse video editing scenarios.** Each case presents frames extracted from the source video and the edited output. Our method enables temporally consistent stylization, dynamic background replacement, identity and attribute editing, scene transformation, and object removal, while preserving motion, structure, and visual coherence across frames.

Abstract

Building a capable video editor remains significantly harder than a video generator: editing requires (source, instruction, edited) triplets that are prohibitively expensive to annotate and difficult to synthesize at scale, whereas image editing has already reached maturity with millions of such pairs readily available. In this work, we introduce **VINCIE-NExT**, a unified framework that transfers editing capability from

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

images to videos through in-context visual demonstrations, alleviating the need for large-scale paired video editing data. VINCIE-NExT decomposes video editing into a structured chain of composable sub-tasks (Video $\rightarrow$ Image $\rightarrow$ Image $\rightarrow$ Video), routing editing intent through the image domain and enabling scalable joint training from heterogeneous image and video corpora under a unified diffusion objective. An image editing pair, synthesized by the model or supplied by the user, is prepended as an in-context visual demonstration that serves as a spatial appearance blueprint for every output frame. To ground appearance edits across the interleaved context, we introduce a novel position encoding that links image demonstrations and video frames in a shared spatial coordinate system, enabling pixel-faithful propagation of appearance changes to every output frame. Chain-of-Editing further provides principled test-time scaling: by executing the sub-task chain as progressive diffusion stages, editing quality can be improved by investing additional compute without retraining. Comprehensive experiments on OpenVE-Bench demonstrate the state-of-the-art performance across diverse editing categories, with ablations confirming the effectiveness of each component.

1 Introduction

Videos have become the dominant medium for creative expression, communication, and knowledge sharing, driving strong demand for video content creation. Recent advances in video foundation models [7, 46, 40, 34, 54, 67] have induced unprecedented visual quality [77, 32, 63], yet building a capable video *editor* can precisely modify appearance, style, object attributes, or scene content while preserving temporal coherence, remains significantly challenging. *The fundamental bottleneck is Data*. Compared with video generation which is supervised with large collections of raw video-text pairs, video editing requires *pairwise* data in the form of (source video, editing instruction, edited video) triplets, which are prohibitively expensive to annotate and difficult to synthesize at scale.

Existing approaches to video editing supervision fall into two broad strategies. The first constructs editing pairs through specialized pipelines, *e.g.*, applying optical-flow warping, depth estimation, segmentation, or dedicated image editors frame by frame [24, 29]. These pipeline-based strategies suffer from compounding errors across multiple tools and limit editing diversity to the capabilities of their constituent components. The second strategy applies pre-trained image editors independently per frame at inference time [9, 65], sidestepping training-data requirements but failing to model cross-frame dependencies, leading to temporal flickering and incoherent global edits.

In recent years, *image editing* has reached a level of maturity far surpassing its video counterpart. Large-scale corpora [6, 82, 66, 78] provide millions of (source image, instruction, edited image) triplets covering diverse operations, such as style transfer, object replacement, background modification, and beyond. Crucially, an image editing pair $(\mathbf{I}^s, \mathbf{I}^t)$ is more than a training signal: it constitutes a *visual demonstration* of the desired appearance transformation. If a generative model can leverage such a pair as in-context evidence, it may generalize the same edit to video sequences, substantially reducing the dependence of video editing on large-scale paired video data.

This intuition drives the core design principle of our work: *transferring editing capability from images to videos through in-context visual demonstrations*. To this end, we introduce **V**ideo **IN**-context **Cha**l**I**n-of-**E**diting for next visual generation (**VINCIE-NExT**), a unified framework that decomposes video editing into a structured chain of composable sub-tasks, *i.e.*, Video $\rightarrow$ Image $\rightarrow$ Image $\rightarrow$ Video, and learns to execute this chain via joint in-context image-video modeling. The image editing pair $(\mathbf{I}^s, \mathbf{I}^t)$, either synthesized by the model or supplied directly by users, is prepended to the interleaved context and serves as a spatial appearance blueprint that guides the generation of every frame in the edited video.

To support interleaved in-context modeling, we introduce Time Dual-Frequency 3D Rotary Position Encoding (TDF3D-RoPE), which assigns identical relative positions to spatially corresponding patches across image demonstrations and video frames. This design enables the model to precisely localize appearance changes from the image pair and propagate them faithfully to each output frame. The interleaved formulation further unlocks a two-stage inference strategy, Chain-of-Editing (**CoE**): the first stage (V $\rightarrow$ I $\rightarrow$ I) produces a high-quality edited image, and the second stage (V $\rightarrow$ I $\rightarrow$ I $\rightarrow$ V) uses that image as a visual demonstration to guide full-video generation.

The framework yields three key advantages. First, sub-task decomposition enables scalable joint training from heterogeneous data: the I $\rightarrow$ I stage draws from abundant image editing pairs, and the I $\rightarrow$ V stage from image-conditioned video generation data. Second, the in-context visual demonstration provides a rich, pixel-grounded spatial prior that guides every denoising step during generation, yielding stronger editing fidelity than text prompts alone. Third, CoE offers principled test-time scaling, trading additional compute for higher editing quality without any retraining. In summary, our main contributions are as follows.

- We propose a video editing paradigm that transfers image editing capabilities to video through in-context visual demonstrations, alleviating the need for large-scale paired video editing data.
- We introduce a joint in-context image–video modeling framework with a V $\rightarrow$ I $\rightarrow$ I $\rightarrow$ V sub-task decomposition, enabling scalable training from heterogeneous image and video corpora under a unifed diffusion learning objective.
- We present TDF3D-RoPE to enable pixel-faithful propagation of edits from image demonstrations to video frames, and build upon it a two-stage *Chain-of-Editing* inference strategy that provides principled test-time scaling without retraining.
- We conduct comprehensive experiments on OpenVE-Bench [21], where VINCIE-NExT achieves state-of-the-art performance across diverse editing categories, with ablations validating the effectiveness of each component.

2 Related Work

Video Generation and Editing. Diffusion models have become the dominant paradigm for video generation, progressing from factorized spatial-temporal attention and latent video diffusion [23, 53, 64, 19] to powerful Diffusion Transformer (DiT)-based generators [45, 7, 77, 32, 46, 63, 83, 52]. Video editing has been addressed through inversion-based attention manipulation [41, 38, 47, 8], temporal consistency enforcement via feature propagation or flow-guided attention [18, 30, 12], canonical atlas representations [31, 2, 10, 43], and training-based general editors operating on paired corpora or instruction-following data [42, 13, 70, 15, 68, 33, 48, 28]. Despite this progress, training high-quality video editors remains data-intensive, motivating transfer of editing capability from the more data-rich image domain.

Image Editing. Image editing with diffusion models spans a broad spectrum of approaches: training-free methods manipulate the diffusion process through attention injection or inversion [22, 62], structural conditioning adapters enable fine-grained spatial control [79], instruction-following training on large synthetic corpora yields versatile text-driven editors [6, 17, 56, 82, 39], and reference-guided approaches support visually-grounded modifications beyond text specification [74, 66]. Most relevant to our work is the paradigm of *in-context* image editing, where editing operations are demonstrated by example pairs rather than described textually [25, 71, 73, 58]. These systems demonstrate that diverse, high-quality image editing corpora can train powerful general-purpose editors—a resource we explicitly exploit to supervise video editing in this work.

Editing Ability Transfer from Image to Video. A natural strategy to bootstrap video editing is to transfer capabilities from image models. Architecture inflation—inserting temporal layers into image U-Nets or applying image-level control signals frame-by-frame [5, 4, 3, 81, 20]—transfers visual quality but not semantic editing behaviors. Zero-shot methods apply image editors per-frame with temporal coherence constraints [9, 65, 27, 75, 76], or propagate a single edited frame to the full sequence via optical flow or image-to-video priors [36, 44]; however, these approaches are prone to temporal flickering and cannot generalize to edits requiring holistic temporal reasoning. Recent training-based efforts [48, 33] incorporate image editing supervision into video pipelines, yet treat the two modalities as separate tasks requiring dedicated heads or task-specific fine-tuning. Besides, ICVE [37] and EditVerse [29] explore in-context video editing, but both perform single-pass video-to-video generation without an explicit edited keyframe. Our approach takes a fundamentally different route: *in-context image-video modeling*. Rather than inflating architectures or applying image editors per-frame, we present image editing pairs as in-context demonstrations to a diffusion transformer that jointly attends over image and video tokens, enabling richer and more temporally consistent video editing with minimal video-specific supervision. The resulting chain further provides a persistent image anchor, test-time extensibility, and a plug-in interface for external image editors.

3 Methodology

We present VINCIE-NExT, a video editing framework that *transfers editing capability from images to videos through in-context visual demonstrations*, as demonstrated in Fig. 2. We decompose video editing into a chain: Video → Image → Image → Video, routing editing intent through the image domain where mature large-scale priors are readily available. An image editing pair $(\mathbf{I}^s, \mathbf{I}^t)$, synthesized or user-supplied, serves as a spatial blueprint guiding every output frame. The framework centers on three components: (i) *Decomposition of Video Editing*, (ii) *Joint In-context Image–Video Modeling*, and (iii) *Chain-of-Editing*, a two-stage inference strategy executing V→I→I then V→I→I→V.

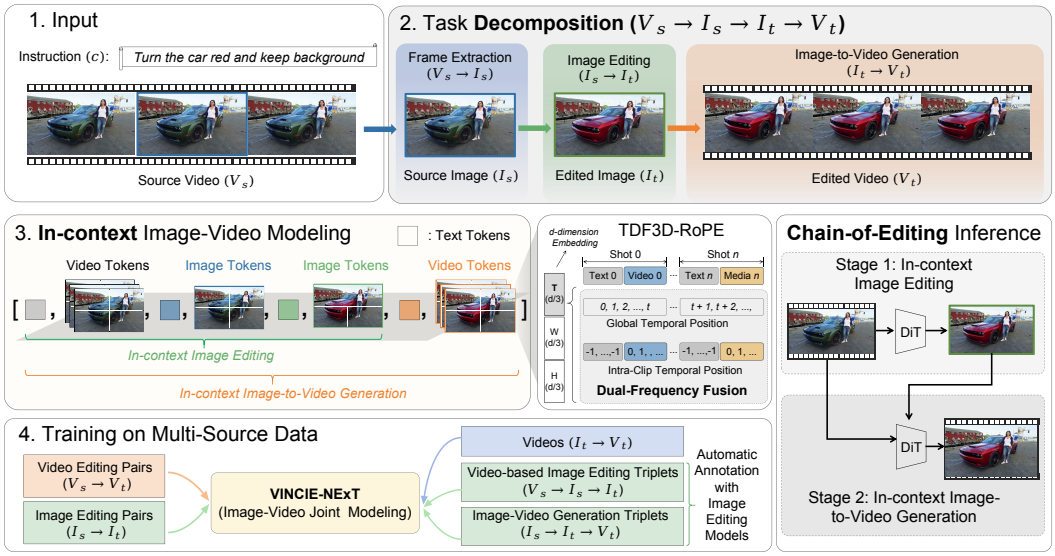


Figure 2: Overview of our proposed VINCIE-NExT framework. The video editing process is decomposed into a structured chain of sub-tasks (Video → Image → Image → Video). The image editing pair $(\mathbf{I}^s, \mathbf{I}^t)$ serves as an in-context visual demonstration that transfers image editing priors to video generation via a unified diffusion backbone.

3.1 Decomposition of Video Editing

Problem formulation. Let $\mathbf{V}^s = (f_1, \ldots, f_T)$ denote a source video of T frames, c an editing instruction, and $\mathbf{V}^t = (f'_1, \ldots, f'_T)$ the corresponding edited video. Conventional video editing approaches model $p(\mathbf{V}^t \mid \mathbf{V}^s, c)$ directly, requiring the model to simultaneously handle high-level semantic editing, fine-grained appearance control, and cross-frame temporal consistency, a joint objective that demands large-scale paired video data which is costly to acquire.

Sub-task decomposition. We instead factorize the editing process by introducing intermediate image representations. Given a representative keyframe $\mathbf{I}^s$ extracted from $\mathbf{V}^s$ and its instruction-edited counterpart $\mathbf{I}^t$, the joint distribution over the full generation chain factorizes as:

$$p(\mathbf{I}^s, \mathbf{I}^t, \mathbf{V}^t \mid \mathbf{V}^s, c) = \underbrace{p(\mathbf{I}^s \mid \mathbf{V}^s, c)}_{\text{V}\to\text{I}} \cdot \underbrace{p(\mathbf{I}^t \mid \mathbf{I}^s, \mathbf{V}^s, c)}_{\text{I}\to\text{I}} \cdot \underbrace{p(\mathbf{V}^t \mid \mathbf{I}^t, \mathbf{I}^s, \mathbf{V}^s, c)}_{\text{I}\to\text{V}} \tag{1}$$

Each factor corresponds to a distinct sub-task with a clearly delineated role. V→I extracts a semantically representative image from the source video, anchoring the editing operation to a single spatial context. I→I applies the instruction-guided edit at the image level, where mature image editing priors offer fine-grained control. I→V recomposes the edited image into a temporally consistent target video, conditioned on both the edited frame and the original video structure. At inference, the temporal mid-frame of $\mathbf{V}^s$ is chosen as $\mathbf{I}^s$, a safe default. In practice, any frame can be used, as training spans diverse temporal positions and Temporal Position Randomization (§3.2.1) makes the model robust to this choice.

Joint training. This decomposition enables scalable supervision by allowing each sub-task to draw from independently collected data: abundant image editing pairs for I→I, image-conditioned video generation data for I→V, and video–keyframe correspondence data for V→I. We define the combined training corpus as $\mathcal{D} = \mathcal{D}_{V\to I} \cup \mathcal{D}_{I\to I} \cup \mathcal{D}_{I\to V} \cup \mathcal{D}_{\text{full}}$, where $\mathcal{D}_{\text{full}}$ contains full-chain samples that supervise the complete pipeline. Sub-tasks are sampled randomly during training, and all are supervised with a unified flow-matching objective over interleaved visual–text sequences:

$$\mathcal{L} = \mathbb{E}_{(x,\,y)\sim\mathcal{D},\;\epsilon\sim\mathcal{N}(0,\mathbf{I}),\;\tau\sim\text{LogitNormal}(0,1)} \left\|(\epsilon - y) - \epsilon_\theta(y_\tau,\, x,\, \tau)\right\|^2 \tag{2}$$

where $y_\tau = (1-\tau)\,y + \tau\,\epsilon$ is the target visual segment noised to timestep $\tau \in [0,1]$, ϵ_θ predicts the velocity $\epsilon - y$, and x denotes the conditioning context (source video tokens, instruction, and any preceding interleaved segments). This joint training regime allows the model to share a single set of parameters across all sub-tasks, facilitating cross-task knowledge transfer and, in particular, enabling image editing to generalize video.

3.2 Joint In-context Image-Video Modeling

Unified visual tokenization. Images and videos are encoded by a shared video VAE, *i.e.*, $\mathcal{E}$, into continuous latents with $8\times$ spatial and $4\times$ temporal compression, which are then patchified with spatial patch size p. A frame $f \in \mathbb{R}^{H\times W\times 3}$, encoded as a single-frame clip, yields a flat sequence of $N = \frac{H}{8p} \times \frac{W}{8p}$ tokens, and a video $\mathbf{V} = (f_1, \ldots, f_T)$ is encoded jointly into $T' = 1 + \frac{T-1}{4}$ latent frames, yielding NT' tokens.

Interleaved sequence construction. The full generation chain (Eq. 1) is flattened into one contiguous token sequence:

$$\mathbf{x} = \left[\, c_1,\ \mathcal{E}(\mathbf{V}^s),\ c_2,\ \mathcal{E}(\mathbf{I}^s),\ c_3,\ \mathcal{E}(\mathbf{I}^t),\ c_4,\ \mathcal{E}(\mathbf{V}^t) \,\right] \tag{3}$$

where each visual segment is paired with its own text prompt: c_1 is a media reference tag for the source video, c_2 is a frame-sampling prompt for V→I, c_3 is the editing instruction for I→I, and c_4 is a image-to-video generation prompt for I→V. The model generates each visual segment via a diffusion process conditioned on all preceding tokens in the interleaved context. Crucially, the image pair $\big(\mathcal{E}(\mathbf{I}^s),\, \mathcal{E}(\mathbf{I}^t)\big)$ forms an *in-context visual demonstration*: when generating $\mathcal{E}(\mathbf{V}^t)$, the model has access to the full source-to-target appearance mapping at the image level as a pixel-grounded spatial prior for every frame. For sub-task training, any contiguous sub-chain of $\mathbf{x}$ with its paired prompts (*e.g.*, $[c_1, \mathcal{E}(\mathbf{I}^s), c_3, \mathcal{E}(\mathbf{I}^t)]$ for standalone image editing) constitutes a self-contained training sample under the loss in Eq. 2, allowing sub-task-specific data to be mixed in without requiring the full chain.

3.2.1 Time Dual-Frequency 3D Rotary Position Encoding (TDF3D-RoPE)

The interleaved sequence (Eq. 3) mixes text with heterogeneous visual segments, so naive sequential indices would conflate content identity with sequence order. We use a factorized 3D RoPE assigning each token a position triplet (p_t, p_h, p_w) with separate frequency tensors for text and visual tokens.

In-context editing imposes two conflicting positional demands: the model should (i) distinguish tokens from different shots (*e.g.*, $\mathbf{I}^s$ vs. $\mathbf{I}^t$) to maintain causal ordering, while (ii) recognizing spatially corresponding patches across shots to propagate edits pixel-faithfully. We resolve this by computing two independent 3D frequency tensors and fusing them via element-wise addition. The **inter-shot component** assigns monotonically increasing global offsets across shots, making every shot globally distinguishable. The **intra-video component** assigns raw spatial positions (t_i, h, w) with zero global offset, so the same spatial location in any two shots shares identical (p_h, p_w), enabling purely content-driven attention between corresponding patches and pixel-faithful edit propagation from $\mathbf{I}^s$→$\mathbf{I}^t$ to $\mathbf{V}^t$. Under RoPE [57], frequency addition encodes both objectives simultaneously within a single rotation with no modality-type embeddings. Formally, for shot k with accumulated global offset τ_k, a visual token at local coordinates (t_i, h, w) receives

$$\mathbf{p}_{\text{inter}} = (\tau_k{+}t_i,\ \tau_k{+}h,\ \tau_k{+}w), \quad \mathbf{p}_{\text{intra}} = (t_i, h, w), \quad \tau_{k+1} = \tau_k + l_k + \max\big(\lceil t_{\text{last}} \rceil, H, W\big), \tag{4}$$

where l_k is the text length of shot k, t_{last} is the timestamp of the last frame in shot k, H, W is the spatial grid size, and the rotation uses the frequency-domain sum $\mathbf{p}_{\text{inter}} + \mathbf{p}_{\text{intra}}$.

Temporal position randomization for image training data. Assigning a fixed temporal index $t_i = 0$ to all training images creates a spurious correlation between temporal position and the image

modality. We replace it with $\tilde{t} \sim \mathcal{U}[0,\, T_{\max}]$, compelling the model to ground correspondences in spatial content rather than temporal identity. Full formulas are given in Appendix A.

3.3 Chain-of-Editing

The decomposition in Eq. 1 is instantiated at inference time as two sequential diffusion stages. *Chain-of-Editing* (CoE) advances one sub-task at a time, carrying the outputs of prior stages as context for all subsequent ones.

Stage 1 — Image editing (V→I→I). The first stage generates the edited keyframe $\mathbf{I}^t$ by assembling an interleaved 3-shot context from the source video $\mathbf{V}^s$, source keyframe $\mathbf{I}^s$, and the target image slot, then running a full diffusion sampling loop to yield the denoised latent $\hat{z}^t$.

Stage 2 — Video generation (V→I→I→V). The second stage generates $\mathbf{V}^t$ conditioned on $\mathbf{V}^s$, $\mathbf{I}^s$, and $\hat{z}^t$, using a 4-shot context ($\mathbf{V}^s$, $\mathbf{I}^s$, $\hat{z}^t$, target video slot) processed by the *same* backbone, which now attends jointly across frames to model cross-frame dependencies. Critically, $\hat{z}^t$ is inserted directly in latent space, without any VAE decode and re-encode cycle, preserving the output fidelity at Stage 1 and avoiding compounding reconstruction errors.

Per-shot timestep conditioning. Conditioning shots receive timestep $t = 0$ (fully denoised clean signals), while only the target shot receives the active diffusion timestep $\tau \in [0, 1]$. The entire multi-shot sequence is passed to the DiT in a single forward call, so conditioning and target tokens attend to each other at every denoising step.

Test-time scaling. Because each stage is an independent diffusion process over an explicit interleaved context, the chain can be extended at test time without retraining, *e.g.*, iterating image editing multiple times before video generation, trading additional compute for higher fidelity. See Appendix A for implementation details of all stages.

3.4 In-context Video-based Image Editing Data Construction

Beyond video editing pairs, we augment training with *video-based in-context image editing* (V↔I↔I) data: each sample pairs an original video keyframe with an edited counterpart produced by a state-of-the-art image editor under an instruction annotated by GPT-4o (see Appendix B). Unlike conventional image editing pairs trained in isolation, each sample keeps the source video attached as context, *i.e.*, $[\mathcal{E}(\mathbf{V}^s), \mathcal{E}(\mathbf{I}^s), \mathcal{E}(\mathbf{I}^t)]$, teaching exactly the V→I→I sub-step whose output is reused as the demonstration for V→I→I→V, so that image editing ability transfers into video editing rather than remaining a standalone skill. This augmentation provides three key benefits: **(1) Broader editing knowledge.** Diverse categories (*e.g.*, attribute modification, expression transfer, object removal) expose the model to a wider range of editing concepts than video pairs alone. **(2) Stronger intermediate editing.** Training explicitly on the image-editing stage of Chain-of-Edit improves the quality of the edited keyframe that ground subsequent video generation. **(3) Scalability.** High-quality $(\mathbf{I}^s, c, \mathbf{I}^t)$ triplets, can be produced from arbitrary video frames without the video pairs that are costly to collect.

4 Experiments

4.1 Experimental Setup

Training Data. VINCIE-NExT is trained on three complementary data sources corresponding to the **I2I**, **V2V***, and **V↔I↔I** splits in the ablation study (Tab. 2). **I2I** uses paired image editing data from OmniEdit [66]. **V2V*** uses paired video editing data from OpenVE [21], where * denotes that each pair is reformatted into sub-chains of our decomposed V→I→I→V format (*e.g.*, V→V, I→I, I→V) rather than used as a plain V2V pair. **V↔I↔I** is our core novel data source: for each raw video, GPT-4o generates editing instructions that are applied per-frame by Seedream 4.5 [55], forming (source video, sampled frame, edited frame) triplets *without* any paired video editing supervision. The three sources contain 1.20M, 2.45M, and 1.63M samples, respectively. Full construction details are in Appendix C.

Benchmark and Evaluation. We evaluate on OpenVE-Bench [21], comprising 431 video clips across eight editing categories (six spatially-aligned: Global Style, Background Change, Local

Change, Local Remove, Local Add, Subtitle Edit; two non-spatially-aligned: Creative Edit, Camera Edit). Following OpenVE-Bench, we use an MLLM-based evaluator (Gemini 2.5 Pro) that scores each output on a 1–5 scale along three dimensions: *Instruction Compliance*, *Consistency & Detail Fidelity*, and *Visual Quality & Stability*. We compare against seven open-source baselines (VACE [28], OmniVideo [59], InsViE [72], Lucy-Edit [60], ICVE [37], DITTO [1], OpenVE-Edit [21]) and the proprietary Runway Aleph. We further compare with AnyV2V [33], a representative image-editing-driven method, in Appendix D.9. Full benchmark and baseline details are in Appendix C.

Implementation Details. The DiT backbone ϵ_θ is a 3B MM-DiT initialized from the same text-to-video pretrained backbone as VINCIE [51], the visual tokenizer $\mathcal{E}$ is a video VAE inflated from SD3 [14] ($\times 8$ spatial, $\times 4$ temporal downsampling, 16 channels; $p = (1, 2, 2)$), and the text encoder is Flan-T5 [11]. Only the DiT is trained, and it is shared by all sub-tasks and both inference stages. We train on 32 H100 GPUs for about 150 hours (42k steps, lr $= 5 \times 10^{-5}$) in two stages on the same data: Stage 1 at 256×256, then Stage 2 at 640×480 initialized from Stage 1. Inference uses up to 65 frames at 12 FPS, 32 sampling steps, and a classifier-free guidance (CFG) scale of 2.5 (full settings in Tab. 3). Tab. 1 reports the Stage-2 model. Due to limited compute, all ablations use the Stage-1 model with otherwise identical settings.

4.2 Performance Comparison

Table 1: Performance comparison of video editing methods across eight editing categories, evaluated by Gemini 2.5 Pro, with automatic ratings on a 1–5 scale following the OpenVE-Bench protocol. #Reso. denotes the output resolution. Grey rows denote closed-source commercial models.

Methods	#Reso.	Overall	Global Style	Background Change	Local Change	Local Remove	Local Add	Subtitle Edit	Creative Edit	Camera Edit
Runway Aleph	1280 × 720	3.65	3.72	2.62	4.18	4.16	2.78	3.62	3.64	4.53
VACE [28]	1280 × 720	1.57	1.49	1.55	2.07	1.46	1.26	1.48	1.47	1.62
OmniVideo [59]	640 × 352	1.31	1.11	1.18	1.14	1.14	1.36	1.00	2.26	1.00
InsViE [72]	720 × 480	1.53	2.20	1.06	1.48	1.36	1.17	2.18	2.02	1.09
Lucy-Edit [60]	1280 × 704	2.15	2.27	1.57	3.20	1.75	**2.30**	1.61	2.86	1.61
ICVE [37]	384 × 240	2.07	2.22	1.62	2.57	2.51	1.97	2.09	2.41	1.11
DITTO [1]	832 × 480	1.98	4.01	1.68	2.03	1.53	1.41	2.81	1.23	1.32
OpenVE-Edit [21]	1280 × 704	2.49	3.16	2.36	2.98	1.85	2.15	2.91	2.31	2.02
Ours	640 × 480	**3.08**	**4.17**	**2.55**	**3.48**	**3.24**	2.27	**3.45**	**3.41**	**2.06**

Overall results. As shown in Tab. 1, VINCIE-NExT achieves an overall score of **3.08**, surpassing all open-source methods by a substantial margin, with 24% relative gain over OpenVE-Edit, closing the gap to the proprietary Runway Aleph, while substantially reducing the dependence on large-scale paired video editing data. Results are consistent under Seed1.6-VL as an independent judge (Appendix D).

Category-level analysis. Global Style is the standout category (4.17 vs. Runway Aleph's 3.72): the in-context editing paradigm directly encodes the full appearance transformation, and TDF3D-RoPE propagates it temporally. Local Remove also sees dramatic improvement (1.85→3.24 over OpenVE-Edit), as routing edits through a mature image editor yields coherent inpainting. Most other appearance-editing categories, such as Local Change, Subtitle Edit, and Creative Edit, similarly benefit from the design. The primary limitation is Camera Edit, compared with Runway Aleph, which requires novel-viewpoint synthesis, *i.e.*, a geometric operation outside the scope of the appearance editing. Local Add and Background Change also show room for improvement, as both demand synthesizing content absent from the source video.

4.3 Ablation Study

Effect of data composition. As shown in Tab. 2, I2I-only training fails to generalize to video, while V↔I↔I transfers image editing priors without direct video supervision, and V2V* provides the strongest spatially-aligned signal. Their contributions are additive: V2V* anchors spatial fidelity while V↔I↔I broadens coverage of open-domain appearance edits.

Effect of Chain-of-Editing. CoE benefits are contingent on chain-structured training. Applied to an I2I-only model, CoE degrades performance as the two-stage format is entirely out of distribution.

Table 2: Ablation study on data composition and Chain-of-Editing (CoE) on OpenVE-Bench, evaluated by Gemini 2.5 Pro. Rows with grey background indicate that CoE is applied at inference time. I2I, V2V*, and V↔I↔I are the three training data sources. Training on V↔I↔I data alone already enables meaningful video editing, and CoE provides consistent gains when this chain data is present. The full combination of all three data types with CoE achieves the best overall score. All rows use the model after the first training stage at 256×256 resolution.

Data Composition			CoE	Overall	Global Style	Background Change	Local Change	Local Remove	Local Add	Subtitle Edit	Creative Edit	Camera Edit
I2I	V2V*	V↔I↔I										
✓				1.14	1.29	1.00	1.05	1.05	1.00	1.60	1.20	1.04
✓			✓	1.05	1.05	1.00	1.01	1.03	1.00	1.27	1.06	1.00
		✓		1.34	1.68	1.00	1.12	1.03	1.06	2.23	2.09	1.03
		✓	✓	1.80	2.58	1.86	2.03	1.46	1.48	1.62	2.30	1.15
	✓			2.49	3.47	1.96	2.67	2.71	2.08	3.40	2.03	1.29
	✓		✓	2.57	3.32	2.11	3.07	2.87	2.03	3.49	1.88	1.32
✓	✓	✓		2.53	3.06	1.94	2.88	2.85	1.97	3.49	2.69	1.29
✓	✓	✓	✓	2.65	3.64	2.15	3.03	3.11	1.82	3.20	2.89	1.22

The V↔I↔I model benefits most, since its training already includes an intermediate edited frame that CoE exploits as an appearance anchor. Notably, CoE also improves the V2V*-only model (2.49→2.57). Since V2V* is already organized in our decomposed chain format, this shows that CoE generalizes beyond the V↔I↔I data and serves as a training-free boost for paired video data as well.

Synergy of all components. The full model achieves the best overall performance, with CoE gains concentrated in categories where appearance is hardest to specify through text alone, such as global style transfer and creative edits. This confirms CoE functions as a visual grounding mechanism, and the three data types together unlock it as an effective strategy without additional training.

Is paired video data the main driver? The V2V*-only rows in Tab. 2 already use our decomposed chain format. A *Pure V2V* model (same backbone and V2V data, no decomposition or CoE, in Tab. 8 in Appendix D.5) scores only 2.32, below OpenVE-Edit (2.49), which uses a superset[1] of our V2V data. In contrast, full data + CoE reaches 2.65 (+14.2%), and +74.1% on Creative Edit, a category without any direct V2V supervision, while adding only ∼6% visual training tokens. Besides, a blinded pairwise human study on all 431 clips with 10 raters (Fleiss' $\kappa = 0.721$) also prefers full data + CoE over V2V*-only + CoE, most clearly Prompt Following (38% vs. 24%) and Consistency (28% vs. 9%), showing that the seemingly small MLLM-judged gap between the two (2.57 vs. 2.65) is human-perceptible (Appendix D.6).

Effect of intermediate image editing quality. Fig. 3 shows how intermediate image quality influences editing performance through the CoE chain. Self-editing underperforms on categories requiring geometric transformation or novel content synthesis, while replacing it with an external editor yields consistent overall improvement. The two external editors reach the similar overall score with complementary category-

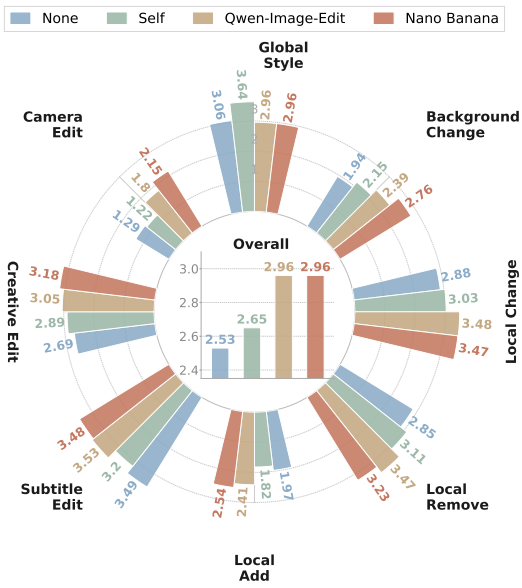


Figure 3: Impact of **intermediate image editing** quality on CoE video editing performance (OpenVE-Bench, scored by Gemini 2.5 Pro). Stronger external editors (*Qwen-Image-Edit* [69] and *Nano Banana* [61]) consistently outperform self-editing, with complementary strengths highlighting the modular advantage of CoE.

[1] OpenVE-Edit is trained on the full OpenVE V2V data, whereas our V2V split excludes the Creative Edit and Camera Multi-Shot Edit categories, which are not publicly released.

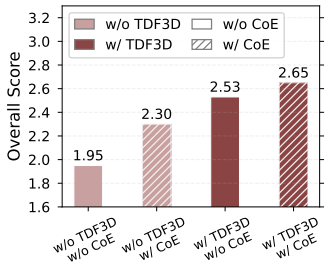


Figure 4: Ablation on TDF3D-RoPE and CoE (Overall score on OpenVE-Bench, scored by Gemini 2.5 Pro). Bar color indicates TDF3D-RoPE; hatch pattern indicates CoE. Both components contribute, and their combination achieves the best result.

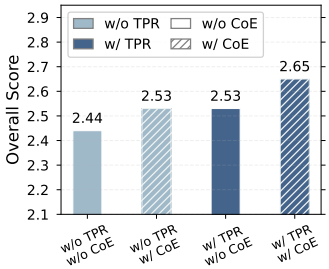


Figure 5: Ablation on TPR and CoE (Overall score on OpenVE-Bench, scored by Gemini 2.5 Pro). Bar color indicates TPR; hatch pattern indicates CoE. Both components contribute, and their combination achieves the best result.

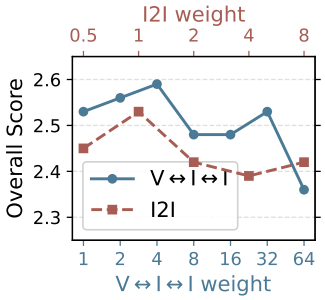


Figure 6: Overall score on OpenVE-Bench scored by Gemini 2.5 Pro as the training data sampling weight of V↔I↔I (bottom axis, blue) and I2I (top axis, red) are each varied independently, with all other mixture proportions held fixed.

level strengths, highlighting a key advantage of the modular I→I interface: advances in image editing transfer directly to video without retraining.

Effect of TDF3D-RoPE. Fig. 4 summarizes the contributions of TDF3D-RoPE and CoE. Both components improve overall performance, and their combination achieves the best score of 2.65. TDF3D-RoPE provides the larger standalone gain by establishing pixel-level spatial correspondence across shots, a prerequisite for faithful localized edits. CoE further refines edits by routing generation through an explicit intermediate image, and the two components together are complementary. Detailed per-category analysis is provided in Appendix D.

Effect of Temporal Position Randomization. Fig. 5 shows that TPR and CoE interact multiplicatively. Without TPR, a fixed temporal index creates a spurious structural correlation that prevents CoE's intermediate keyframe from serving as a genuine appearance anchor. TPR eliminates this bias, and only their combination reaches peak performance. Per-category details are in Appendix D.

Effect of training data sampling weights. Fig. 6 shows that each data source contributes complementary supervision, and performance is best when they are balanced. Moderately upsampling V↔I↔I improves chain exposure, while overly large weights reduce paired video supervision. I2I performs best at the default ratio. Overall, the default mixture provides a good trade-off among the three sources.

4.4 Qualitative Results

Fig. 7 shows global scene stylization (summer-to-snow) and localized compositional effects (fire, water, wind), both maintaining temporal consistency without task-specific fine-tuning. Fig. 8 isolates the CoE contribution on subject replacement and removal. V2V direct inference produces flickering and ghosting, while CoE with self-editing eliminates this instability by anchoring to a reference image. Routing through an external editor further improves semantic accuracy with no retraining, demonstrating that the modular I→I interface lets image editing advances transfer directly to video.

5 Conclusion

In this paper, we presented VINCIE-NExT, a video editing framework that substantially reduces the dependence on large-scale paired video data by routing editing intent through the image domain via a structured V→I→I→V chain, enabling scalable joint training from heterogeneous image and video corpora under a unified flow-matching objective. TDF3D-RoPE propagates pixel-level appearance changes faithfully across every frame, while Chain-of-Editing provides a principled test-time scaling mechanism that trades compute for fidelity without retraining. Evaluations on OpenVE-Bench confirm state-of-the-art performance among state-of-the-art methods, and we hope this progressive

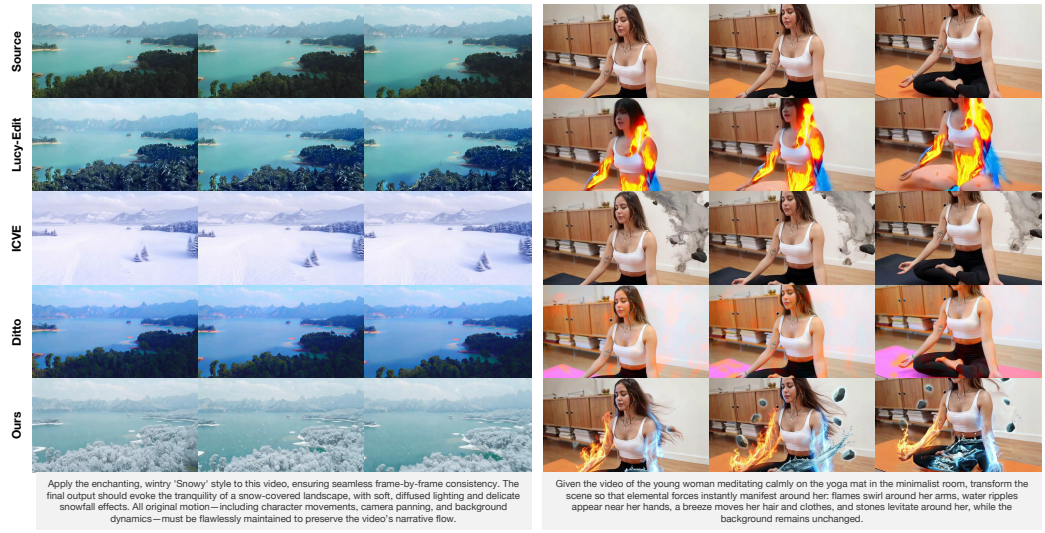


Figure 7: Qualitative results on global scene stylization (**left**, summer-to-snow) and localized compositional editing (**right**, injecting fire/water/wind).



Figure 8: Comparison of V2V direct inference, CoE with self editing, and CoE with an external editor on subject replacement (**left**) and removal (**right**).

editing paradigm, *i.e.*, grounding video editing in mature image priors, opens a practical and scalable path for future complex video creation research.

Acknowledgments and Disclosure of Funding

This research is supported by the National Research Foundation, Singapore under its National Large Language Models Funding Initiative (AISG Award No: AISG-NMLP-2024-002). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

References

[1] Qingyan Bai, Qiuyu Wang, Hao Ouyang, Yue Yu, Hanlin Wang, Wen Wang, Ka Leong Cheng, Shuailei Ma, Yanhong Zeng, Zichen Liu, et al. Scaling instruction-based video editing with a high-quality synthetic dataset. *arXiv preprint arXiv:2510.15742*, 2025.

[2] Omer Bar-Tal, Dolev Ofri-Amar, Rafail Fridman, Yoni Kasten, and Tali Dekel. Text2LIVE: Text-driven layered image and video editing. In *European conference on computer vision*, pages 707–723. Springer, 2022.

[3] Omer Bar-Tal, Hila Chefer, Omer Tov, Charles Herrmann, Roni Paiss, Shiran Zada, Ariel Ephrat, Junhwa Hur, Guanghui Liu, Amit Raj, et al. Lumiere: A space-time diffusion model for video generation. pages 1–11, 2024.

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

[5] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22563–22575, 2023.

[6] Tim Brooks, Aleksander Holynski, and Alexei A Efros. InstructPix2Pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18392–18402, 2023.

[7] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Leo Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. Technical Report 8, 2024.

[8] Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. MasaCtrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 22560–22570, 2023.

[9] Duygu Ceylan, Chun-Hao P Huang, and Niloy J Mitra. Pix2Video: Video editing using image diffusion. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 23206–23217, 2023.

[10] Wenhao Chai, Xun Guo, Gaoang Wang, and Yan Lu. StableVideo: Text-driven consistency-aware diffusion video editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23040–23050, 2023.

[11] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.

[12] Yuren Cong, Mengmeng Xu, Christian Simon, Shoufa Chen, Jiawei Ren, Yanping Xie, Juan-Manuel Perez-Rua, Bodo Rosenhahn, Tao Xiang, and Sen He. FLATTEN: Optical flow-guided attention for consistent text-to-video editing. 2023.

[13] Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7346–7356, 2023.

[14] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning*, 2024.

[15] Ruoyu Feng, Wenming Weng, Yanhui Wang, Yuhui Yuan, Jianmin Bao, Chong Luo, Zhibo Chen, and Baining Guo. CCEdit: Creative and controllable video editing via diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6712–6722, 2024.

[16] Joseph L Fleiss. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382, 1971.

[17] Zigang Geng, Binxin Yang, Tiankai Hang, Chen Li, Shuyang Gu, Ting Zhang, Jianmin Bao, Zheng Zhang, Houqiang Li, Han Hu, et al. Instructdiffusion: A generalist modeling interface for vision tasks. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 12709–12720, 2024.

[18] Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. TokenFlow: Consistent diffusion features for consistent video editing. In *The Twelfth International Conference on Learning Representations*, 2024.

[19] Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. 2023.

[20] Yuwei Guo, Ceyuan Yang, Anyi Rao, Maneesh Agrawala, Dahua Lin, and Bo Dai. SparseCtrl: Adding sparse controls to text-to-video diffusion models. In *European Conference on Computer Vision*, pages 330–348. Springer, 2024.

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

[22] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. 2022.

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

[24] Jiahao Hu, Tianxiong Zhong, Xuebo Wang, Boyuan Jiang, Xingye Tian, Fei Yang, Pengfei Wan, and Di Zhang. VIVID-10M: A dataset and baseline for versatile and interactive video local editing. *arXiv preprint arXiv:2411.15260*, 2024.

[25] Lianghua Huang, Wei Wang, Zhi-Fan Wu, Yupeng Shi, Huanzhang Dou, Chen Liang, Yutong Feng, Yu Liu, and Jingren Zhou. In-context LoRA for diffusion transformers. *arXiv preprint arXiv:2410.23775*, 2024.

[26] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21807–21818, 2024.

[27] Hyeonho Jeong and Jong Chul Ye. Ground-A-Video: Zero-shot grounded video editing using text-to-image diffusion models. 2023.

[28] Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. VACE: All-in-one video creation and editing. pages 17191–17202, 2025.

[29] 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.

[30] Ozgur Kara, Bariscan Kurtkaya, Hidir Yesiltepe, James M Rehg, and Pinar Yanardag. 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*, pages 6507–6516, 2024.

[31] Yoni Kasten, Dolev Ofri, Oliver Wang, and Tali Dekel. Layered neural atlases for consistent video editing. *ACM Transactions on Graphics (TOG)*, 40(6):1–12, 2021.

[32] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

[33] Max Ku, Cong Wei, Weiming Ren, Harry Yang, and Wenhu Chen. AnyV2V: A tuning-free framework for any video-to-video editing tasks. *arXiv preprint arXiv:2403.14468*, 2024.

[34] Kuaishou Technology. Kling AI: High-fidelity video generation model, 2024. URL https://klingai.com/.

[35] J Richard Landis and Gary G Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174, 1977.

[36] Feng Liang, Bichen Wu, Jialiang Wang, Licheng Yu, Kunpeng Li, Yinan Zhao, Ishan Misra, Jia-Bin Huang, Peizhao Zhang, Peter Vajda, et al. FlowVid: Taming imperfect optical flows for consistent video-to-video synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8207–8216, 2024.

[37] Xinyao Liao, Xianfang Zeng, Ziye Song, Zhoujie Fu, Gang Yu, and Guosheng Lin. In-context learning with unpaired clips for instruction-based video editing. *arXiv preprint arXiv:2510.14648*, 2025.

[38] Shaoteng Liu, Yuechen Zhang, Wenbo Li, Zhe Lin, and Jiaya Jia. Video-p2p: Video editing with cross-attention control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8599–8608, 2024.

[39] Shiyu Liu, Yucheng Han, Peng Xing, Fukun Yin, Rui Wang, Wei Cheng, Jiaqi Liao, Yingming Wang, Honghao Fu, Chunrui Han, et al. Step1x-edit: A practical framework for general image editing. *arXiv preprint arXiv:2504.17761*, 2025.

[40] Guoqing Ma, Haoyang Huang, Kun Yan, Liangyu Chen, Nan Duan, Shengming Yin, Changyi Wan, Ranchen Ming, Xiaoniu Song, Xing Chen, et al. Step-video-t2v technical report: The practice, challenges, and future of video foundation model. *arXiv preprint arXiv:2502.10248*, 2025.

[41] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6038–6047, 2023.

[42] Eyal Molad, Eliahu Horwitz, Dani Valevski, Alex Rav Acha, Yossi Matias, Yael Pritch, Yaniv Leviathan, and Yedid Hoshen. Dreamix: Video diffusion models are general video editors. *arXiv preprint arXiv:2302.01329*, 2023.

[43] Hao Ouyang, Qiuyu Wang, Yuxi Xiao, Qingyan Bai, Juntao Zhang, Kecheng Zheng, Xiaowei Zhou, Qifeng Chen, and Yujun Shen. CoDeF: Content deformation fields for temporally consistent video processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8089–8099, 2024.

[44] Wenqi Ouyang, Yi Dong, Lei Yang, Jianlou Si, and Xingang Pan. I2VEdit: First-frame-guided video editing via image-to-video diffusion models. pages 1–11, 2024.

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

[46] 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.

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

[48] Bosheng Qin, Juncheng Li, Siliang Tang, Tat-Seng Chua, and Yueting Zhuang. InstructVid2Vid: Controllable video editing with natural language instructions. In *2024 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2024.

[49] Leigang Qu, Meng Liu, Jianlong Wu, Zan Gao, and Liqiang Nie. Dynamic modality interaction modeling for image-text retrieval. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1104–1113, 2021.

[50] Leigang Qu, Haochuan Li, Tan Wang, Wenjie Wang, Yongqi Li, Liqiang Nie, and Tat-Seng Chua. TIGER: Unifying text-to-image generation and retrieval with large multimodal models. In *International Conference on Learning Representations*, 2025.

[51] Leigang Qu, Feng Cheng, Ziyan Yang, Qi Zhao, Shanchuan Lin, Yichun Shi, Yicong Li, Wenjie Wang, Tat-Seng Chua, and Lu Jiang. VINCIE: Unlocking in-context image editing from video. In *International Conference on Learning Representations*, 2026.

[52] Leigang Qu, Ziyang Wang, Na Zheng, Wenjie Wang, Liqiang Nie, and Tat-Seng Chua. TTOM: Test-time optimization and memorization for compositional video generation. In *International Conference on Learning Representations*, 2026.

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

[54] Team Seedance, De Chen, Liyang Chen, Xin Chen, Ying Chen, Zhuo Chen, Zhuowei Chen, Feng Cheng, Tianheng Cheng, Yufeng Cheng, et al. Seedance 2.0: Advancing video generation for world complexity. *arXiv preprint arXiv:2604.14148*, 2026.

[55] Team Seedream, Yunpeng Chen, Yu Gao, Lixue Gong, Meng Guo, Qiushan Guo, Zhiyao Guo, Xiaoxia Hou, Weilin Huang, Yixuan Huang, et al. Seedream 4.0: Toward next-generation multimodal image generation. *arXiv preprint arXiv:2509.20427*, 2025.

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

[57] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[58] Quan Sun, Yufeng Cui, Xiaosong Zhang, Fan Zhang, Qiying Yu, Yueze Wang, Yongming Rao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Generative multimodal models are in-context learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14398–14409, 2024.

[59] Zhiyu Tan, Hao Yang, Luozheng Qin, Jia Gong, Mengping Yang, and Hao Li. Omni-video: Democratizing unified video understanding and generation. *arXiv preprint arXiv:2507.06119*, 2025.

[60] DecartAI Team. Lucy Edit: Open-weight text-guided video editing. 2025. URL https://d2drjpuinn46lb.cloudfront.net/Lucy_Edit__High_Fidelity_Text_Guided_Video_Editing.pdf.

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

[62] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1921–1930, 2023.

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

[64] Jiuniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shiwei Zhang. Modelscope text-to-video technical report. *arXiv preprint arXiv:2308.06571*, 2023.

[65] Wen Wang, Yan Jiang, Kangyang Xie, Zide Liu, Hao Chen, Yue Cao, Xinlong Wang, and Chunhua Shen. Zero-shot video editing using off-the-shelf image diffusion models. *arXiv preprint arXiv:2303.17599*, 2023.

[66] Cong Wei, Zheyang Xiong, Weiming Ren, Xeron Du, Ge Zhang, and Wenhu Chen. OmniEdit: Building image editing generalist models through specialist supervision. 2024.

[67] Thaddäus Wiedemer, Yuxuan Li, Paul Vicol, Shixiang Shane Gu, Nick Matarese, Kevin Swersky, Been Kim, Priyank Jaini, and Robert Geirhos. Video models are zero-shot learners and reasoners. Technical report, 2025.

[68] Bichen Wu, Ching-Yao Chuang, Xiaoyan Wang, Yichen Jia, Kapil Krishnakumar, Tong Xiao, Feng Liang, Licheng Yu, and Peter Vajda. Fairy: Fast parallelized instruction-guided video-to-video synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8261–8270, 2024.

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

[70] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-Video: One-shot tuning of image diffusion models for text-to-video generation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7623–7633, 2023.

[71] Shaojin Wu, Mengqi Huang, Wenxu Wu, Yufeng Cheng, Fei Ding, and Qian He. Less-to-more generalization: Unlocking more controllability by in-context generation. pages 18682–18692, 2025.

[72] Yuhui Wu, Liyi Chen, Ruibin Li, Shihao Wang, Chenxi Xie, and Lei Zhang. InsViE-1M: Effective instruction-based video editing with elaborate dataset construction. *arXiv preprint arXiv:2503.20287*, 2025.

[73] Shitao Xiao, Yueze Wang, Junjie Zhou, Huaying Yuan, Xingrun Xing, Ruiran Yan, Chaofan Li, Shuting Wang, Tiejun Huang, and Zheng Liu. OmniGen: Unified image generation. pages 13294–13304, 2025.

[74] Binxin Yang, Shuyang Gu, Bo Zhang, Ting Zhang, Xuejin Chen, Xiaoyan Sun, Dong Chen, and Fang Wen. Paint by example: Exemplar-based image editing with diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18381–18391, 2023.

[75] Shuai Yang, Yifan Zhou, Ziwei Liu, and Chen Change Loy. Rerender a video: Zero-shot text-guided video-to-video translation. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–11, 2023.

[76] Shuai Yang, Yifan Zhou, Ziwei Liu, and Chen Change Loy. FRESCO: Spatial-temporal correspondence for zero-shot video translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8703–8712, 2024.

[77] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.

[78] Yang Ye, Xianyi He, Zongjian Li, Bin Lin, Shenghai Yuan, Zhiyuan Yan, Bohan Hou, and Li Yuan. Imgedit: A unified image editing dataset and benchmark. *arXiv preprint arXiv:2505.20275*, 2025.

[79] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3836–3847, 2023.

[80] Shiwei Zhang, Jiayu Wang, Yingya Zhang, Kang Zhao, Hangjie Yuan, Zhiwu Qin, Xiang Wang, Deli Zhao, and Jingren Zhou. I2VGen-XL: High-quality image-to-video synthesis via cascaded diffusion models. *arXiv preprint arXiv:2311.04145*, 2023.

[81] Y Zhang, Y Wei, D Jiang, X Zhang, W Zuo, and Q Tian. ControlVideo: Training-free controllable text-to-video generation. *arXiv preprint arXiv:2305.13077*, 3.

[82] Haozhe Zhao, Xiaojian Ma, Liang Chen, Shuzheng Si, Rujie Wu, Kaikai An, Peiyu Yu, Minjia Zhang, Qing Li, and Baobao Chang. UltraEdit: Instruction-based fine-grained image editing at scale. *Advances in Neural Information Processing Systems*, 37:3058–3093, 2024.

[83] Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-Sora: Democratizing efficient video production for all. *arXiv preprint arXiv:2412.20404*, 2024.

Contents

1 Introduction **2**

2 Related Work **3**

3 Methodology **4**

3.1 Decomposition of Video Editing 4

3.2 Joint In-context Image-Video Modeling 5

3.3 Chain-of-Editing 6

3.4 In-context Video-based Image Editing Data Construction 6

4 Experiments **6**

4.1 Experimental Setup 6

4.2 Performance Comparison 7

4.3 Ablation Study 7

4.4 Qualitative Results 9

5 Conclusion **9**

Appendix **17**

A Additional Details of Methodology **19**

A.1 TDF3D-RoPE: Position Assignment Formulas 19

A.2 Chain-of-Editing: Implementation Details 19

B In-context Video-based Image Editing Annotation **19**

B.1 Editing Category Analysis 20

B.2 Text Instruction Annotation 21

B.3 Frame Editing Annotation 22

C Detailed Experimental Settings **22**

C.1 Training Data 22

C.2 Benchmark Details 23

C.3 Implementation Details 24

D Additional Experimental Results **24**

D.1 Performance Comparison Evaluated by Seed1.6-VL 24

D.2 TDF3D-RoPE and Chain-of-Editing 25

D.3 Temporal Position Randomization 25

D.4 Data Composition and Image Editing Quality 26

D.5 Comparison with Pure V2V Training 26

D.6 Human Evaluation 27

D.7 Evaluation with Objective VBench Metrics 28
D.8 Test-time Scaling and Efficiency 28
D.9 Comparison with AnyV2V 29

E More Qualitative Results **29**
E.1 Qualitative Comparison between Baselines and Ours 29
E.2 Qualitative Comparison between V2V and CoE Variants of VINCIE-NExT 30

A Additional Details of Methodology

A.1 TDF3D-RoPE: Position Assignment Formulas

We partition the interleaved sequence into *shots*, each consisting of one text segment followed by one visual segment. A global offset τ_{start} is accumulated across shots. Within shot k of text length l_k, text tokens receive sequential 1D positions $[\tau_{\text{start}},\ \tau_{\text{start}} + l_k)$, broadcast to all three axes (p_t, p_h, p_w). For the visual segment, floating-point frame timestamps $\{t_i\} \subset \mathbb{R}_{\geq 0}$ serve as temporal positions, natively accommodating non-uniform sampling and variable frame rates. All three axes of the visual tokens are uniformly shifted by $\tau_{\text{start}} + l_k$, keeping every shot in a non-overlapping position range. The offset advances as:

$$\tau_{\text{start}} \leftarrow \tau_{\text{start}} + l_k + \max\big(\lceil t_{\text{last}} \rceil,\ H,\ W\big), \tag{5}$$

where t_{last} is the last frame timestamp of the current shot, and H, W are the spatial grid dimensions. These assignments constitute the **inter-shot component**.

For the **intra-video component**, visual tokens are assigned their raw positions (t_i, h, w) with zero global offset, independent of shot index; text tokens are fixed at position -1. Two patches at the same spatial location (h, w) in any pair of shots therefore share an identical intra-video (p_h, p_w) and have zero relative spatial offset, making their attention scores purely content-driven. The head dimension d is partitioned equally across the three axes ($d/3$ channels each), and the final per-token frequency is the element-wise sum of the inter-shot and intra-video components.

A.2 Chain-of-Editing: Implementation Details

Stage 1 — Image editing (V→I→I). The interleaved context is assembled from three shots: the source video $\mathbf{V}^s$, the source keyframe $\mathbf{I}^s$, and the target image slot to be denoised. Frame indices are assigned using the actual temporal timestamps of the selected keyframes, normalized by the total video length. The visual segments are processed frame-independently (`frame_separate=True`), reflecting the spatial nature of image editing.

Stage 2 — Video generation (I→V). The interleaved context contains four shots: $\mathbf{V}^s$, $\mathbf{I}^s$, and $\hat{z}^t$ from Stage 1 (three conditioning shots), followed by the target video slot. The four-shot frame index sequence is $\big[\{0,\ldots,T{-}1\},\ \{t_i^{\text{mf}}\},\ \{t_i^{\text{mf}}\},\ \{0,\ldots,T{-}1\}\big]$, aligning the target video temporally with the source. Visual segments are processed jointly as a continuous sequence (`frame_separate=False`) to model cross-frame dependencies.

Per-shot timestep conditioning. This design keeps the architecture identical to training, where reference segments are likewise observed as clean tokens in the interleaved sequence. No architectural modifications—such as additional cross-attention modules or conditioning adapters—are required at inference time.

Test-time scaling. Longer chains—*e.g.*, iterating the image-editing stage multiple times before advancing to video generation—progressively enrich the interleaved context and allow the model to refine intermediate results, trading additional compute for higher editing fidelity. Because the chain extension operates purely by prepending more conditioning shots to the context, it requires no retraining or fine-tuning.

B In-context Video-based Image Editing Annotation

This section describes the pipeline for annotating and constructing the V↔I↔I (video-to-image-to-image) training data, where each sample pairs an original video frame with an edited counterpart under a text instruction. The annotation process involves three coordinated stages, each with explicit design constraints to ensure data quality. First, **category annotation** is performed before instruction writing to control the distribution of editing types: a live feedback mechanism steers GPT-4o toward under-represented categories, preventing the long-tailed imbalance that would arise from unconstrained annotation. Second, **category selection rules** are applied so that the assigned editing type is grounded in what is actually visible in the raw video, ensuring the suggested edit aligns with natural human aesthetics and realistic editing intent rather than being arbitrary or implausible. Third, **text instruction constraints** are imposed during annotation to encourage lexical and semantic diversity: by tracking

annotation statistics and injecting them into the prompt, we discourage the model from repeatedly gravitating toward salient but overused edits (*e.g.*, always annotating sunset or color-grading effects), producing a richer and more varied instruction vocabulary across the dataset.

B.1 Editing Category Analysis

To ensure a balanced and diverse editing category distribution across the dataset, we run a dedicated category analysis pass on each raw video before the full instruction-annotation stage. Given $N = 5$ frames uniformly sampled from a video, **GPT-4o** is prompted to (i) produce a structured multi-granularity caption covering the overall scene, individual characters, prominent objects, and camera behavior, and (ii) propose 1–5 applicable editing categories with subcategories and motivations drawn from a predefined 13-category taxonomy: `remove`, `replace`, `attribute`, `background`, `global`, `effect`, `camera`, `action`, `expression`, `dynamics`, `position`, `pose`, and `orientation`.

A key design feature is a **live distribution feedback** mechanism. As annotations accumulate, a process-safe shared dictionary tracks the running per-category count. At each inference call, the prompt is dynamically populated with three derived statistics: (a) the current category distribution (`category_dist`), (b) the five most over-represented categories (`high_category`), and (c) the five most under-represented categories (`low_category`). Injecting this live context steers the model toward less-covered categories, progressively balancing the taxonomy without any manual curation or post-hoc resampling.

The category labels produced in this step are used downstream to guide the text instruction annotation stage (Appendix B.2), where each sample is constrained to one editing category and the same low-frequency steering statistics are re-injected to maintain global balance.

Prompt for Editing Category Analysis

You are a professional video editing assistant. Given {N_FRAME} frames from a video, your tasks are:
1. Describe the video content: overall scene, characters (appearance, relations), objects, and camera.
2. List 1–5 applicable editing categories with subcategory and motivation, chosen from: `remove`, `replace`, `attribute`, `background`, `global`, `effect`, `camera`, `action`, `expression`, `dynamics`, `position`, `pose`, `orientation`.

Selection rules (prefer categories supported by visible evidence):

Humans & faces

- Large/clear face (close-up) → `expression` (smile, surprise, anger), then `orientation` (look direction), then `effect` (tears, sweat, glow).
- Upper body visible → `action` (wave, point, clap) and interaction (handing/holding), then `pose`.
- Full body visible → `pose` (stance, posture), then `action`, then `position` (move left/right).
- Multiple people → `action`/interaction (hug, handshake), then `position` (swap places), then `expression` (match reactions).
- Occluded/crowded → `remove`/`replace` character only if practical; otherwise `camera` (reframe) or `background` (blur crowd).

Objects & props

- Single prominent object → `remove`/`replace` object, then `attribute` (color, material, texture), then `effect` (transform).
- Many repeated objects → `position` (count/layout), then `remove` clutter.
- Text/logos/signage visible → `remove`/`replace` text; fall back to `camera` (crop/zoom).
- Unsafe/sensitive items → `remove`/`replace` object (make non-threatening).

Scene & background

- Distracting or iconic background → `background` (replace), or `remove` distracting objects, then `camera` (reframe).
- Strong location context (beach, street) → `global` (weather/lighting/season/style), only if consistent with visible cues (shadows, clothing).
- Indoor with lamps/windows → `global` (lighting direction, contrast).

Motion & dynamics

- Natural phenomena (fire, smoke, water, rain, fog) → `dynamics` (intensify/slow/stop), then `global` (stormy/clear), then `effect` (sparks, embers).
- Vehicles / fast motion → `camera` (tracking, stabilization) and `dynamics` (skid, splash, dust trail).
- Explosions / breaking / spills / collapsing → `dynamics` first, then `effect`.

Camera & style

• Shaky footage → `camera` (stabilize, smooth transitions).

• Subject too small / too wide → `camera` (crop-in, push-in, reframe).

• Rapid cuts / inconsistent angles → `camera` (match cuts), then `global` (color consistency).

Global mood

• Night/dark cues → `global` (darker atmosphere, stronger contrast); bright daylight → `global` (clearer, punchier look).

• Prefer concrete global edits: lighting direction, fog, rain, storm, season, cinematic style. Avoid contradicting clothing or shadow cues.

Effects

• Choose `effect` for a localized, visually obvious transformation (glow, freeze, dissolve, morph) on one focal element (face, hand, single object).

Balance constraint: current category distribution – {category_dist}. Avoid over-represented categories; prefer under-represented ones: {low_category}.

Output: a single JSON object – {"caption": {"video": ⟨·⟩, "scene": ⟨·⟩, "char1": ⟨·⟩, ... }, "editing1": {"category": ⟨·⟩, "subcategory": ⟨·⟩, "motivation": ⟨·⟩}, "editing2": {... }, ... }

B.2 Text Instruction Annotation

Each video clip in the dataset is pre-assigned an editing category from the 13-category taxonomy described in Appendix B.1. To generate text editing instructions grounded in actual video content, we uniformly sample N frames from each clip and send them to **GPT-4o** together with a category-specific prompt.

The model is instructed to: (1) carefully describe the visual content of the sampled frames, including the scene, characters, objects, and camera; (2) propose a plausible and visually meaningful editing operation consistent with the assigned category; (3) formulate a concise forward editing instruction; and (4) formulate a *reversed* instruction that describes how to edit the result back toward the original, *without* assuming any prior knowledge of the original video state (*e.g.*, if the forward instruction is "remove the dog," the reversed instruction is "add a dog"). The model additionally identifies which frames are affected by the edit.

The output is a structured JSON record containing the video caption, editing motivation, forward instruction, reversed instruction, and the set of affected frame indices. The reversed instructions are included to support bidirectional training, ensuring the model learns both how to apply an edit and how to undo it without relying on privileged knowledge of the source content.

The 13 editing categories each have a distinct scope and constraint, summarized as follows (these definitions are used as *Definition of {CATEGORY}* in the prompt):

- **Remove**: completely eliminate a visible character, animal, or object so it is absent from every frame in which it originally appeared.
- **Replace**: substitute an existing subject with a different one while preserving the original motion, timing, camera movement, and scene layout.
- **Attribute**: modify intrinsic properties of an existing object (color, texture, material, brightness, or shape) without adding, removing, or replacing the object itself; the change is applied consistently across all frames.
- **Background**: swap the environment behind the main foreground subject(s) throughout the entire clip, leaving foreground content untouched.
- **Global**: apply a uniform visual transformation to the whole video (such as color grading, lighting style, season, weather, or artistic look), affecting every frame equally without targeting any specific element.
- **Effect**: overlay a localized visual treatment (glow, particle, distortion, hologram, motion trail, *etc.*) on a specific subject or object, sustained consistently across frames.
- **Camera**: simulate post-production virtual camera movements (zoom, pan, dolly, tilt, rotation, or handheld shake) that alter the viewer's perspective without modifying scene content.

- **Action**: change what the main subject is *doing* (modifying motion type (*e.g.*, walk → run), adding continuous movement to a static subject, or altering motion intensity) while leaving appearance and scene unchanged.
- **Expression**: alter the visible emotional state of a character's face (*e.g.*, neutral → smile, calm → angry) through adjustments to brow, eye, and mouth configuration, without changing action or scene.
- **Dynamics**: introduce a physical state change in an object over time (such as igniting, extinguishing, melting, blooming, exploding, or dissolving), emphasizing the transformation process itself.
- **Position**: reposition a subject within the frame (left/right/up/down) to alter composition, without changing its size, color, or temporal behavior.
- **Pose**: reconfigure the body arrangement of a character (limb placement, stance, or body posture) without changing facial expression, spatial orientation, or camera angle.
- **Orientation**: rotate or flip a subject to change the direction it faces or the angle at which it appears, without altering pose, expression, or scene content.

Prompt for Text Instruction Annotation

You are a professional video editing assistant. You will be provided with {N_FRAME} frames from a video and an assigned editing type: **{CATEGORY}**.

Definition of {CATEGORY}: {CATEGORY_DEFINITION}

Your task is to write a text instruction to edit the video following the assigned editing type:
1. Carefully observe the frames and describe the content: environment, main subjects (appearance, relations, interactions), and background. Identify characters and objects without using real names.
2. Following the editing type, propose how to edit the video to make it more engaging, explaining your reasoning.
3. Summarize the editing idea into a concise user instruction.
4. Write a reversed instruction describing how to edit the result back, *without* any knowledge of the original state (*e.g.*, "remove the dog" → reversed: "add a dog"). Do **not** use words like "restore", "revert", or "return".
5. Identify which frames should be edited (*e.g.*, only frames where the subject appears).

Note: Each frame must be individually editable; the instruction must be valid when applied per frame.

Output: a single JSON object in one line –
{"caption": {"video": ⟨description⟩, "scene": ⟨·⟩, "char1": ⟨·⟩, "obj1": ⟨·⟩, "camera": ⟨·⟩}, "motivation": ⟨what and why to edit⟩, "instruction": ⟨concise editing instruction⟩, "instruction_reversed": ⟨reversed instruction⟩, "frame_idx_edited": ⟨list of frame indices <{N_FRAME}⟩}

B.3 Frame Editing Annotation

Given the text editing instruction produced in Appendix B.2 and the corresponding set of affected frame indices, we apply **Seedream 4.5** as an instruction-guided image editor to produce an edited counterpart for each selected keyframe. Each frame is edited independently: the text instruction is passed to Seedream 4.5 together with a concise constraint that the model should maintain the original composition and background, preserve the main subject's identity, and produce edits that are subtle yet clearly visible. This per-frame formulation is consistent with the design of the text annotation stage (Appendix B.2), where each instruction is required to be self-contained and valid when applied to a single image. The final output for each clip is a set of *(original frame, edited frame, instruction)* triplets over the affected indices, which serve as the video-based in-context image editing data for training our model.

C Detailed Experimental Settings

C.1 Training Data

VINCIE-NExT is trained on three complementary data sources, which correspond to the **I2I**, **V2V***, and **V**↔**I**↔**I** splits used in the ablation study (Tab. 2). Their sizes are 1.20M image editing pairs (I2I, OmniEdit [66]), 2.45M video editing pairs (V2V, OpenVE [21]), and 1.63M chain triplets (V↔I↔I,

ours). Our pairwise video split is a subset of the training set of OpenVE-Edit [21], as it excludes the unreleased Creative Edit and Camera Multi-Shot Edit categories.

Image-to-Image data (I2I). I2I uses paired image editing data from OmniEdit [66], covering a diverse range of editing categories with high-quality before–after image pairs.

Video-to-Video data (V2V*). V2V* uses paired video editing data from OpenVE [21], where each sample consists of a source video, an editing instruction, and a corresponding edited video. The asterisk indicates that these pairs are not used as plain video-to-video samples: during training, each pair is reformatted into a randomly chosen sub-chain of $\mathrm{V} \to \mathrm{I} \to \mathrm{I} \to \mathrm{V}$, including direct video-to-video editing, multi-frame-to-multi-frame editing (frames sampled from both videos), multi-frame-to-video generation, and the full interleaved chain. This distinguishes V2V* from the Pure V2V baseline (Appendix D.5), which trains on the same pairs without decomposition.

Video↔Image↔Image chain data (V↔I↔I). This is the core novel data source that enables VINCIE-NExT to bridge video editing through the image domain. We start from raw videos in the OpenVE collection *without* using any of their pairwise editing annotations. For each video, we uniformly sample $N = 5$ frames and feed them to GPT-4o with a structured prompt that generates a balanced editing instruction across 12 categories (*e.g.*, global style, local add/remove, background, dynamics). The instruction is then applied independently to each sampled frame using Seedream 4.5 [55], an expressive image editor, to produce per-frame before–after pairs. Together with the original video, these form a Video → Image → Image chain triplet: the source video, an unedited representative frame, and its edited counterpart. This construction avoids the need for any paired video editing supervision while providing rich, instruction-grounded visual demonstrations for training.

C.2 Benchmark Details

Benchmark. We evaluate on OpenVE-Bench [21], a comprehensive benchmark for instruction-guided video editing comprising 431 video clips organized into eight editing categories. Six categories involve *spatially-aligned* edits, where the output must preserve the original spatial layout: **Global Style** (transferring artistic or weather styles globally), **Background Change** (replacing scene backgrounds while keeping foreground subjects intact), **Local Change** (modifying the appearance of a specific object or region), **Local Remove** (erasing a target element and plausibly filling the vacated region), **Local Add** (inserting a new object into the scene), and **Subtitle Edit** (modifying on-screen text overlays). The remaining two categories involve *non-spatially-aligned* edits that do not require strict spatial correspondence: **Creative Edit** (free-form stylistic or compositional transformations) and **Camera Edit** (synthesizing novel viewpoints or camera motions). This taxonomy spans a wide spectrum of real-world editing demands, from fine-grained local manipulations to global cinematic effects.

Evaluation protocol. Following OpenVE-Bench, we adopt an MLLM-based evaluation protocol that closely mirrors human judgment: the MLLM is prompted with the same rubric and scoring dimensions a human rater would use. All benchmark scores reported in our tables are produced automatically by the MLLM judge; the separate human study is described in Appendix D.6. For each test sample, the source video, editing instruction, and edited video are jointly fed to a multimodal evaluator, which scores the output along three orthogonal dimensions on a **1–5** scale: (i) *Instruction Compliance*: how faithfully the edit follows the specified instruction; (ii) *Consistency & Detail Fidelity*: how well unedited regions and fine-grained visual details are preserved; and (iii) *Visual Quality & Stability*: overall perceptual quality and temporal coherence. Crucially, Instruction Compliance acts as an upper bound for the remaining two dimensions: a visually polished but semantically incorrect edit is penalized even if it scores high on quality alone. Per-category scores are averaged across all three dimensions to yield a final overall score, and we report both category-level and overall performance.

Baselines. We compare VINCIE-NExT against a range of representative methods covering both open-source and commercial systems. Open-source baselines include VACE [28], a unified video creation and editing framework; OmniVideo [59], an omni-directional video editing model; InsViE [72], an instruction-driven video editor; Lucy-Edit [60], a training-efficient video editing method; ICVE [37], an in-context video editor; DITTO [1], a disentangled video editing approach; and OpenVE-Edit [21], the official model released alongside OpenVE-Bench. We also compare against the proprietary

system **Runway Aleph**, which represents the current commercial frontier. All methods are evaluated under their default inference settings on the full OpenVE-Bench test set.

Baseline selection. Our selection follows three criteria. (1) *Family coverage*: we pick one or two representative methods per active family of instruction-guided video editing (unified creation/editing frameworks, in-context editors, and dataset-centric editors) rather than several methods from the same family. (2) *Recency*: all baselines were released within roughly one year before our submission, *i.e.*, VACE (Mar. 2025), InsViE (Mar. 2025), OmniVideo (Jul. 2025), Lucy-Edit (Sep. 2025), ICVE (Oct. 2025), DITTO (Oct. 2025), and OpenVE-Edit (Dec. 2025). (3) *Backbone fairness*: these baselines build on modern DiT-based video foundation models (*e.g.*, VACE and Lucy-Edit on Wan [63], ICVE on HunyuanVideo [32]), so the comparison isolates our technical contribution rather than gains from a newer backbone. AnyV2V [33] (Mar. 2024) is built on the U-Net-based I2VGen-XL [80], but since it is the most closely related image-editing-driven method, we additionally compare against it in Appendix D.9. EditVerse [29] is not open-sourced and therefore cannot be evaluated.

C.3 Implementation Details

Tab. 3 consolidates the architecture, data, training, and inference settings of VINCIE-NExT. Training and inference launch scripts and full hyperparameter configurations are provided in our code repository.

Table 3: Implementation details of VINCIE-NExT.

Architecture	
DiT backbone	3B MM-DiT, initialized from the same T2V-pretrained backbone as VINCIE [51]
VAE	Video VAE inflated from SD3 [14]; spatial $\times 8$, temporal $\times 4$, 16 channels
Patch size (t, h, w)	$(1, 2, 2)$
Text encoder	Flan-T5 [11]
Trainable / frozen	DiT trained; VAE and text encoder frozen
Training data	
V2V (OpenVE)	2.45M pairs
I2I (OmniEdit)	1.20M pairs
V↔I↔I (ours)	1.63M triplets
Training	
Hardware	$32 \times$ H100 GPUs
Total training	$\approx$150 hours, 42k steps
Stage 1	256×256, $\approx$80 hours
Stage 2	480×640, $\approx$70 hours, initialized from Stage 1
Learning rate	5×10^{-5}
Inference	
Output frames	65 (maximum multi-shot budget)
FPS	12
Sampling steps	32
CFG scale	2.5
Source keyframe $\mathbf{I}^s$	Temporal mid-frame of $\mathbf{V}^s$

D Additional Experimental Results

D.1 Performance Comparison Evaluated by Seed1.6-VL

Tab. 4 reports full per-category results evaluated by Seed1.6-VL as an independent MLLM judge. The rank ordering is consistent with the Gemini 2.5 Pro results in the main paper: VINCIE-NExT achieves an overall score of **3.09**, surpassing all open-source baselines by a substantial margin and narrowing the gap to Runway Aleph to 0.41 points. Category-level trends are likewise stable: Global Style (4.25) and Creative Edit (3.39) are consistent strengths, while Camera Edit (2.01) remains the primary weakness, supporting the reliability of our main conclusions.

Table 4: Performance comparison of video editing methods across eight editing categories, evaluated by **Seed1.6-VL**. Scores are automatic MLLM ratings on a 1–5 scale following the OpenVE-Bench protocol. #Reso. denotes the output resolution. Grey rows denote closed-source commercial models.

Methods	#Reso.	Overall	Global Style	Background Change	Local Change	Local Remove	Local Add	Subtitle Edit	Creative Edit	Camera Edit
Runway Aleph	1280 × 720	3.50	3.47	2.84	3.88	3.88	2.79	3.50	3.23	4.48
VACE [28]	1280 × 720	1.17	1.41	1.16	1.43	1.00	1.05	1.02	1.13	1.16
OmniVideo [59]	640 × 352	1.02	1.02	1.00	1.00	1.00	1.00	1.16	1.00	1.00
InsViE [72]	720 × 480	1.40	2.25	1.23	1.60	1.00	1.23	1.22	1.68	1.02
Lucy-Edit [60]	1280 × 704	1.95	2.17	2.20	3.30	1.03	2.37	1.06	2.35	1.14
ICVE [37]	384 × 240	2.25	2.35	1.86	2.91	2.68	2.27	2.04	1.94	1.38
DITTO [1]	832 × 480	2.06	3.70	2.23	2.28	1.00	2.08	1.01	2.61	1.51
OpenVE-Edit [21]	1280 × 704	2.41	3.11	**2.72**	3.19	1.42	**2.41**	2.56	2.01	1.24
Ours	640 × 480	**3.09**	**4.25**	2.57	**3.56**	**3.26**	2.24	**3.45**	**3.39**	**2.01**

D.2 TDF3D-RoPE and Chain-of-Editing

Table 5: Ablation study on TDF3D-RoPE, which fuses inter-shot and intra-shot position encodings so that spatially corresponding patches across shots share zero relative position, enabling content-driven pixel-level correspondence. Evaluated on OpenVE-Bench with Gemini 2.5 Pro.

TDF3D-RoPE	CoE	Overall	Global Style	Background Change	Local Change	Local Remove	Local Add	Subtitle Edit	Creative Edit	Camera Edit
		1.95	2.78	2.02	2.17	2.15	1.71	1.39	2.09	1.04
	✓	2.30	2.83	1.69	2.67	2.33	1.97	3.12	2.41	1.33
✓		2.53	3.06	1.94	2.88	2.85	1.97	3.49	2.69	1.29
✓	✓	2.65	3.64	2.15	3.03	3.11	1.82	3.20	2.89	1.22

Tab. 5 isolates the contributions of TDF3D-RoPE and CoE, revealing that spatial grounding is the more fundamental capability: TDF3D-RoPE alone yields a larger standalone gain than CoE alone, because establishing where to edit is a prerequisite for executing the edit faithfully. The advantage of TDF3D-RoPE is sharpest on tasks that demand fine-grained localization (Subtitle Edit, Local Remove, and Local Change), where zero relative position between inter-shot patches resolves an otherwise ambiguous correspondence problem. Without TDF3D-RoPE, the model must infer spatial alignment from content similarity alone, which is unreliable when foreground objects deform or when the editing instruction refers to a region not uniquely identifiable by appearance. The two components are complementary but not fully additive: combining them lifts the overall score, yet the joint gain falls short of the sum of individual contributions, suggesting they partially address the same failure mode: the model misplacing edits when inter-shot context is weak. An instructive exception is Subtitle Edit, where TDF3D-RoPE alone *outperforms* the full model: the intermediate I→I step in CoE introduces a degree of freedom that is unnecessary when TDF3D-RoPE already provides precise text-region anchoring, and the additional step adds noise rather than signal. Background Change shows the reverse: CoE without TDF3D-RoPE actively *hurts* performance, because the external image editor can freely alter large homogeneous regions that are semantically unconstrained, and without patch-level anchoring the video backbone has no mechanism to suppress these spurious changes. Camera Edit remains consistently weak regardless of configuration, underscoring that neither positional encoding restructuring nor chain-based appearance editing addresses the core deficit: the model lacks geometric transformation capability.

D.3 Temporal Position Randomization

Tab. 6 reveals that TPR is a prerequisite for CoE to work as intended, rather than an independent source of gain. Without TPR, assigning a fixed temporal index to all training images creates a spurious correlation between position zero and the image modality (Section 3): the model learns to treat temporally-anchored tokens as modality identifiers rather than as spatial content references. This has a concrete consequence at inference time: when CoE inserts the intermediate edited image into the context, the model partially ignores its appearance content and instead interprets it as a structural cue, so the spatial editing signal encoded in the keyframe fails to propagate faithfully to the generated video. Categories demanding holistic appearance transfer, such as Global Style, are most sensitive

Table 6: Ablation study on Temporal Position Randomization (TPR) for image editing training data, evaluated on OpenVE-Bench with Gemini 2.5 Pro.

TPR	CoE	Overall	Global Style	Background Change	Local Change	Local Remove	Local Add	Subtitle Edit	Creative Edit	Camera Edit
		2.44	3.30	2.01	2.78	2.41	1.84	3.27	2.48	1.33
	✓	2.53	3.25	2.37	2.68	2.63	1.99	3.36	2.58	1.22
✓		2.53	3.06	1.94	2.88	2.85	1.97	3.49	2.69	1.29
✓	✓	2.65	3.64	2.15	3.03	3.11	1.82	3.20	2.89	1.22

to this failure mode: CoE without TPR actually *degrades* performance there, because the anchor misleads more than it guides. TPR resolves this by randomly assigning temporal positions to training images, forcing the model to ground correspondence in spatial content irrespective of when a frame nominally occurs. Once this bias is removed, the edited keyframe serves as a genuine appearance anchor: the model can read the full $\mathbf{I}^s \rightarrow \mathbf{I}^t$ transformation and propagate it frame-by-frame, which is precisely the mechanism CoE is designed to exploit. The interaction between the two components is therefore not additive but multiplicative: TPR prepares the representational substrate that makes CoE's test-time chain structurally meaningful, and tasks that require fine-grained spatial reasoning (Local Remove, Global Style) show the sharpest compounding benefit.

D.4 Data Composition and Image Editing Quality

Tab. 7 shows a clear dissociation: PQ remains stable across configurations while SQ varies significantly, confirming that data composition governs instruction fidelity rather than low-level visual quality. V↔I↔I yields the strongest single-source SQ, as its chain structure forces an explicit semantic commitment before video generation. Adding V↔I↔I on top of I2I+V2V* produces the largest SQ gain, demonstrating that paired data and chain data play orthogonal roles.

Table 7: Ablation study on data composition evaluated on GEdit-Bench [39], with evaluation metrics including SQ (Semantic Consistency), PQ (Perceptual Quality), and O (Overall Score).

I2I	V2V*	V↔I↔I	SQ	PQ	O
✓			3.61	5.94	3.67
	✓		3.65	5.90	3.65
		✓	4.65	5.88	4.43
✓	✓		4.54	5.82	4.42
✓	✓	✓	5.23	5.66	4.86

D.5 Comparison with Pure V2V Training

The V2V*-only rows of Tab. 2 are trained on paired video data that is already reformatted into our decomposed, interleaved chain format. To isolate the effect of decomposition and of the backbone itself, we train a *Pure V2V* model with the same backbone on the same V2V data, but with no decomposition, no in-context formulation, and no CoE. We compare it with OpenVE-Edit [21], which is trained on OpenVE's released V2V data, a strict superset of our pairwise video split (ours excludes the unreleased Creative Edit category).

Table 8: Comparison with Pure V2V training on OpenVE-Bench (Gemini 2.5 Pro). V2V: paired video editing data. Creat.: Creative Edit V2V data. Decomp.: decomposed, interleaved chain-of-editing format of V2V. IC: our V↔I↔I in-context data. I2I: OmniEdit image editing data.

Method	V2V	Creat.	Decomp.	IC	I2I	Overall	Creative Edit	Global Style	Local Remove
OpenVE-Edit	✓	✓				2.49	2.31	3.16	1.85
Pure V2V (ours)	✓					2.32	1.66	3.13	2.33
V2V* + CoE (ours)	✓		✓			2.57	1.88	3.32	2.87
Full data + CoE (ours)	✓		✓	✓	✓	**2.65**	**2.89**	**3.64**	**3.11**

Table 9: Per-category gains of full data + CoE over Pure V2V on OpenVE-Bench (Gemini 2.5 Pro). Local Edit averages Local Change, Local Remove, and Local Add.

Method	Overall	Global Style	Background Change	Local Edit	Creative Edit
Pure V2V (w/o decomp., w/o CoE)	2.32	3.13	1.89	2.22	1.66
Full data + CoE	2.65	3.64	2.15	2.65	2.89
Δ	+0.33 (+14.2%)	+0.51 (+16.3%)	+0.26 (+13.8%)	+0.43 (+19.5%)	+1.23 (+74.1%)

With strictly less V2V data, Pure V2V underperforms OpenVE-Edit (2.32 vs. 2.49), confirming that our gains do not stem from a stronger backbone. Applying decomposition and CoE to the same reduced data already surpasses it (2.57), and full data + CoE further improves both the overall score (2.65) and Creative Edit (2.89 vs. 2.31), despite having zero V2V supervision for that category. As shown in Tab. 9, the overall gain over Pure V2V is $+14.2\%$, whereas the gain on Creative Edit is $+74.1\%$, roughly $5\times$ larger. This gain comes from two sources, neither of which is paired video data: (1) I2I and V$\leftrightarrow$I$\leftrightarrow$I data enabling implicit transfer via joint image–video modeling, and (2) CoE inference providing explicit transfer via the in-context $\mathbf{I}^t$ demonstration. Moreover, although I2I and V$\leftrightarrow$I$\leftrightarrow$I add 2.83M samples on top of V2V*, they increase the total number of visual training tokens by only $\sim$6%, as image pairs are far cheaper than full videos; this yields a 14% overall gain, a $> 2\times$ efficiency ratio. We therefore do not claim that video data is eliminated—the V$\rightarrow$I$\rightarrow$I$\rightarrow$V chain still requires video data to learn the final I$\rightarrow$V step—but that the dependence on large-scale *paired* video editing data is substantially reduced, and we expect further scaling of image-level data to compound this improvement.

D.6 Human Evaluation

Protocol. To verify that the MLLM-measured gap between V2V*-only + CoE (2.57) and full data + CoE (2.65) reflects a human-perceptible difference, we conduct a pairwise human evaluation over all 431 OpenVE-Bench samples. We built a blinded side-by-side annotation tool: for each sample, the source video, the instruction, and the two edited outputs are shown as anonymized "Video 1" and "Video 2"; left/right placement is randomized per sample to remove position bias, and the order of samples is shuffled. For each of three dimensions (Prompt Following, Consistency, and Visual Quality), the rater selects "Video 1 better", "Video 2 better", or "Tie". Ten independent raters each annotated all 431 samples. We will release the annotation tool and the raw per-clip, per-rater votes.

Table 10: Pairwise human evaluation on OpenVE-Bench (431 samples). A: V2V*-only + CoE; B: full data + CoE.

Dimension	A wins	B wins	Tie
Prompt Following	24%	**38%**	39%
Consistency	9%	**28%**	63%
Visual Quality	14%	**15%**	71%

Table 11: Inter-rater agreement (10 raters), aggregated over all three dimensions.

Fleiss' κ [16]	0.721
Unanimous rate (10/10)	59.3%
Mean pairwise agreement	90.4%

Results and agreement. As shown in Tab. 10, B wins more often than A on every dimension, most clearly on Prompt Following and Consistency, in line with the MLLM scores. Since OpenVE-Bench uses a 1–5 rating scale, the absolute range is inherently narrow, and a 0.08-point gap should be read in that context. Fleiss' $\kappa = 0.721$ falls in the "substantial agreement" band (0.61–0.80) [35], and the 90.4% mean pairwise agreement indicates that almost any two raters agree on the vast majority of items. The unanimous rate is lower (59.3%), as expected: the boundary between "Tie" and a narrow preference is soft, and 10-way unanimity is a much stricter bar than pairwise agreement.

Why ties dominate Visual Quality and Consistency. (1) Visual Quality depends heavily on the foundation model, which is shared identically between the two configurations (they differ only in training data composition), so raters have little basis to prefer either. (2) For Consistency, V2V*-only + CoE more often fails to execute the requested edit, and a failed or partial edit trivially leaves unrelated regions unchanged, which reads as good consistency even though the edit did not succeed. (3) More generally, whether an instruction was followed is usually an unambiguous judgment, whereas subtle background drift or per-frame artifacts are harder to perceive and compare, so raters default to "Tie" more often on Consistency and Visual Quality than on Prompt Following.

D.7 Evaluation with Objective VBench Metrics

Motivation. All benchmark scores in the main paper are produced by MLLM judges, which may share systematic biases. Although the rank ordering is consistent across two independent MLLM judges (Appendix D.1), we further seek evidence that does not rely on any MLLM. We therefore evaluate with VBench [26], a widely used benchmark suite for video generative models whose dimensions are computed by dedicated perception models and hand-crafted measures rather than by an MLLM.

Metrics. We report two VBench quality metrics: `imaging_quality`, which measures per-frame low-level distortions such as over-exposure, noise, and blur using the MUSIQ image quality predictor, and `quality_avg`, which aggregates VBench's video-quality dimensions to reflect the overall visual and temporal quality of the generated video. Both are computed on the edited videos that each method produces for OpenVE-Bench; higher is better.

Results. As shown in Tab. 12, VINCIE-NExT achieves the best scores on both metrics among recent baselines. The margin is clearest on `imaging_quality` ($+0.027$ over the strongest baseline, ICVE), indicating that routing the edit through an explicit keyframe does not degrade per-frame fidelity, while `quality_avg` remains on par with or slightly above DITTO.

Scope. These metrics assess only the visual and temporal quality of the output; they do not measure whether the edit follows the instruction or preserves unedited content. They are therefore complementary to, rather than a replacement for, the MLLM-based evaluation, which explicitly scores instruction compliance and consistency.

Table 12: VBench objective metrics on OpenVE-Bench.

Method	imaging_quality	quality_avg
Lucy-Edit [60]	0.652	0.800
ICVE [37]	0.688	0.812
DITTO [1]	0.687	0.816
Ours	**0.715**	**0.819**

D.8 Test-time Scaling and Efficiency

Test-time scaling. We vary the number of denoising steps of the intermediate image-editing stage of CoE (1, 4, 16, 64), holding everything else fixed, and evaluate on the full OpenVE-Bench (Tab. 13, Fig. 9). The overall score improves monotonically from 2.30 to 2.81 ($+22.2\%$ relative), *i.e.*, additional test-time compute yields consistent quality gains without retraining, and the trend holds across the reported editing categories. These results are scored with Gemini 3.1 Flash Lite, since Gemini 2.5 Pro was temporarily inaccessible during this experiment; absolute scores are therefore not directly comparable with the Gemini 2.5 Pro numbers elsewhere in the paper, but the judge is held fixed across all four settings, so the trend itself is valid.

Table 13: Test-time scaling via the number of denoising steps of the intermediate image-editing stage, evaluated on OpenVE-Bench by Gemini 3.1 Flash Lite. Inference time is measured on one H200 GPU.

Steps	Time	Overall	Background Change	Creative Edit	Global Style	Local Add	Local Change	Local Remove	Subtitle Edit
1	15.02s	2.30	1.72	2.03	3.59	1.64	2.48	2.44	3.03
4	15.93s	2.55	2.29	2.49	3.60	1.80	2.61	2.85	3.28
16	18.69s	2.79	2.69	2.72	**3.67**	**2.24**	2.96	**3.15**	3.42
64	35.20s	**2.81**	**2.76**	**2.97**	3.55	2.10	**3.05**	3.06	**3.57**

Inference latency. Since CoE consists of two sequential stages, we measure its inference latency against two baselines from Tab. 1 (ICVE and Lucy-Edit) and AnyV2V, all on the same single H200 GPU with 50 denoising steps (Tab. 14). Despite the two-stage design, VINCIE-NExT is faster than ICVE, comparable to Lucy-Edit, and over an order of magnitude faster than AnyV2V, so the explicit intermediate edit does not come at a large cost to practical usability.

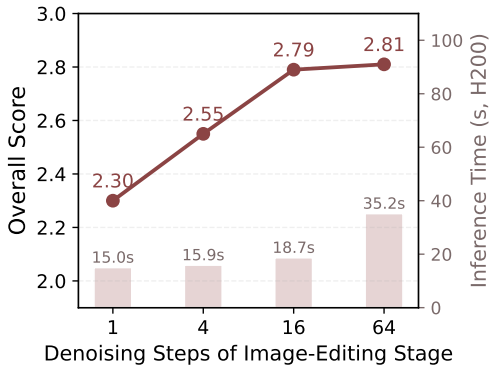


Figure 9: Test-time scaling of CoE: overall score (line, left axis) and inference time (bars, right axis) as the number of denoising steps of the intermediate image-editing stage increases. Scored by Gemini 3.1 Flash Lite.

Table 14: Inference latency on one H200 GPU (50 denoising steps for all methods).

Method	Inference Time
AnyV2V [33]	11min 46s
ICVE [37]	118.1s
Lucy-Edit [60]	39.8s
Ours (Stage 1 + Stage 2)	18.4s + 35.1s = 53.5s

D.9 Comparison with AnyV2V

AnyV2V [33] is closely related to our I→V design, as it also leverages image editing for video editing. We evaluate it on the full OpenVE-Bench and report the results in Tab. 15. These scores are produced by Gemini 3.0 Flash Lite because Gemini 2.5 Pro was inaccessible at the time, so they are not directly comparable with Tab. 1. Both of our checkpoints outperform AnyV2V by a large margin, and AnyV2V is also over 13× slower (Appendix D.8).

Table 15: Comparison with AnyV2V on OpenVE-Bench, evaluated by Gemini 3.0 Flash Lite.

Method	#Reso.	Overall	Background Change	Camera Edit	Creative Edit	Global Style	Local Add	Local Change	Local Remove	Subtitle Edit
AnyV2V [33]	360 × 640	1.62	1.38	1.16	2.17	3.05	1.29	1.56	1.16	1.37
Ours (Stage 1)	256 × 256	2.79	2.71	1.15	2.88	3.52	2.33	2.96	3.17	3.36
Ours (Stage 2)	480 × 640	**3.26**	**3.16**	**2.26**	**3.76**	**3.95**	**2.55**	**3.62**	**3.45**	**3.37**

E More Qualitative Results

This section provides extended qualitative results complementing the main paper. Figs. 10–16 compare VINCIE-NExT against all open-source baselines across seven editing categories. Fig. 17 then isolates the contribution of Chain-of-Editing by contrasting V2V direct inference with CoE using self-editing and an external editor.

E.1 Qualitative Comparison between Baselines and Ours

Each figure displays the source video, the editing instruction, and the outputs of all seven open-source baselines alongside VINCIE-NExT. For spatially-aligned categories (Background Change, Global Style, Local Add, Local Change, Local Remove, Subtitle Edit), VINCIE-NExT consistently produces edits that precisely follow the instruction while preserving the identity of untouched regions and maintaining temporal coherence across frames. Baseline methods tend to fail in one of two modes: either the edit is not applied at all (instruction non-compliance), or the edit bleeds into unrelated

regions and introduces flicker. For the non-spatially-aligned Creative Edit category, most baselines struggle with semantic reinterpretation; VINCIE-NExT benefits from routing the edit through the image domain, which provides a concrete appearance blueprint before video generation begins.

E.2 Qualitative Comparison between V2V and CoE Variants of VINCIE-NExT

Fig. 17 isolates the effect of Chain-of-Editing on two challenging tasks: subject replacement and subject removal. V2V direct inference, while capable of approximate edits, produces temporal flickering and incomplete object removal because the model must simultaneously infer *what* to change and generate consistent video dynamics. CoE with self-editing addresses this by committing to an edited keyframe first: the intermediate image explicitly resolves the semantic ambiguity before the video backbone generates the remaining frames, yielding sharper boundaries and more stable motion. Substituting the self-editor with a stronger external image editor further improves semantic accuracy—particularly for subject replacement, where the external model can synthesize convincing appearance details that the self-editor occasionally blurs. Crucially, this substitution requires no retraining: the modular I→I interface means that improvements in image editing transfer directly to video editing, demonstrating a clear path for future scaling.

Future Work

Two directions are promising for extending VINCIE-NExT. First, a more unified model could span image, video, and other modalities (*e.g.*, audio and 3D) and couple understanding with generation in a single backbone [58, 73, 59], allowing it to reason about the source video before deciding how to edit it rather than following a fixed V→I→I→V decomposition. Second, Chain-of-Editing can be generalized into an agentic framework that plans multi-step edits, invokes external tools (*e.g.*, segmentation, image editors, and quality verifiers) on demand, and leverages retrieval-augmented generation [49, 50] to ground edits involving rare concepts or novel content.

Limitations

Despite promising results, VINCIE-NExT has several limitations. First, output quality is bottlenecked by the intermediate I→I step: errors in the edited keyframe (*e.g.*, hallucinated details or incomplete edits) propagate to every output frame. As shown in Fig. 3, self-editing underperforms on edits requiring geometric transformation or novel-content synthesis, while stronger external editors consistently improve video quality. Since Stage 2 treats $\mathbf{I}^t$ as a hard spatial anchor, a badly mismatched edit will propagate rather than be "corrected"—which is the intended behavior for faithfully preserving the $\mathbf{I}^s \to \mathbf{I}^t$ edit. This risk is mitigated by image editing being a far more mature task than video editing; we leave a systematic stress test to future work. Second, a single keyframe may not adequately characterize videos with significant temporal variation or large camera motion. Third, Chain-of-Editing requires two sequential diffusion stages; although its latency (53.5s on one H200) is lower than ICVE and comparable to Lucy-Edit (Appendix D.8), longer test-time chains add cost. Finally, evaluation is limited to OpenVE-Bench; generalization to long or highly dynamic videos remains open.

Scope

VINCIE-NExT is designed for instruction-driven appearance editing of short video clips, including style transfer, object attribute modification, and background replacement. It is not designed for structural edits that require inserting or removing objects across frames, nor for edits that depend on physical simulation or long-range temporal reasoning. The framework assumes a static or slowly varying camera; videos with rapid cuts or severe occlusion may challenge the keyframe-grounded editing paradigm. User-supplied image demonstrations extend the framework beyond text instructions, but crafting effective demonstrations for highly complex edits remains non-trivial.

Broader Impact

VINCIE-NExT lowers the barrier to high-quality video editing by substantially reducing the need for large-scale paired video data and expensive per-video fine-tuning, making capable editing tools

more accessible to individual creators and small studios. At the same time, the same capability can be misused to produce deceptive or non-consensual manipulations of video content. We encourage the community to pair advances in generative editing with robust provenance and detection tools, and we endorse established responsible-disclosure norms when deploying such systems in consumer-facing products.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state four concrete contributions: the V→I→I→V sub-task decomposition, joint in-context image–video modeling, Two3dFused RoPE, and Chain-of-Editing, all of which are experimentally validated in Section 4. The scope (open-source methods, OpenVE-Bench, no paired video editing data required) is clearly delimited, and limitations such as Camera Edit are acknowledged.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in Section 4: Camera Edit performance lags substantially behind the commercial baseline because the appearance-editing chain cannot handle geometric/viewpoint transformations; Local Add and Background Change are also identified as harder cases requiring novel content synthesis. The IC-RoPE ablation further discusses when CoE degrades performance (Subtitle Edit) and the conditions under which it helps.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not include formal theorems or proofs. The factorization in Eq. 1 is a probabilistic decomposition presented as a modeling choice rather than a formal theoretical result.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The methodology section fully describes the sub-task decomposition, interleaved sequence construction, Two3dFused RoPE formulation (including the dual-component frequency tensors and fusion rule), Chain-of-Editing inference procedure, and temporal position randomization. Training data sources (OpenVE, OmniEdit) and the V↔I↔I data construction pipeline are described in Section 4 and Appendix B. The evaluation benchmark (OpenVE-Bench, 431 clips, 8 categories) and scoring protocol (Gemini 2.5 Pro, Seed1.6-VL, 1–5 scale) are specified. Code is available at the anonymous repository linked in the abstract.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example

(a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code and qualitative results are provided at the anonymous repository linked in the abstract. Training data sources (OpenVE, OmniEdit) are publicly available datasets cited in the paper. The V↔I↔I chain data construction pipeline, including the full GPT-4o annotation prompt, is detailed in Appendix B.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Training data composition (three splits: I2I from OmniEdit, V2V from OpenVE, V↔I↔I constructed via GPT-4o+Seedream 4.5) and the sub-task sampling strategy are described in Section 4. The evaluation protocol (OpenVE-Bench, MLLM-based scoring on 1–5 scale, two independent evaluators) is fully specified. Inference details (Chain-of-Editing two-stage procedure, per-shot timestep conditioning, `frame_separate` flag) are given in the methodology section.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Error bars are not reported. The evaluation relies on MLLM-based scoring (Gemini 2.5 Pro and Seed1.6-VL) over the full 431-clip OpenVE-Bench test set; since MLLM scoring is deterministic given fixed inputs, per-run variance is negligible. Cross-evaluator consistency (two independent MLLMs yielding the same rank ordering) is used as the primary evidence of result stability.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: Compute resource details (GPU type, memory, training time) are not currently reported in the paper. We will include this information in the final version.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [Yes]

Justification: The paper conducts no human subjects research. Training data is sourced from publicly available datasets (OpenVE, OmniEdit) with appropriate citations. The annotation pipeline using GPT-4o operates on video content without collecting personal data. No privacy violations, safety risks, or discriminatory applications are introduced by the work.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The positive impact is enabling accessible, high-quality video editing without expensive paired data collection, lowering the barrier for creative content production. The primary negative impact is potential misuse for generating manipulated video content (deepfakes, disinformation). This risk is inherent to all generative video editing systems; responsible release practices (e.g., usage guidelines) should accompany any public deployment.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [No]

Justification: Explicit technical safeguards (e.g., content filters, watermarking) are not described in this submission. As a research prototype, the model is released for reproducibility and academic review rather than public deployment; we will document usage guidelines alongside any broader release.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All datasets and models used are cited: OpenVE [21], OmniEdit [66], Seedream 4.5 [55], Qwen-Image-Edit-2511 [69], Nano Banana [61], and baseline methods (VACE, OmniVideo, InsViE, Lucy-Edit, ICVE, DITTO). GPT-4o is used via the OpenAI API under its terms of service for annotation purposes only.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The V↔I↔I chain dataset is the primary new asset. Its construction is documented in Section 4 (data pipeline, GPT-4o annotation with Seedream 4.5 editing on OpenVE videos, $N = 5$ frames per video, 12 editing categories) and in Appendix B (full annotation prompt). The dataset will be released under an appropriate open license alongside the code.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.

- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or human subject research. Evaluation is conducted entirely via MLLM-based scoring (Gemini 2.5 Pro and Seed1.6-VL).

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects research; no IRB approval is required.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: GPT-4o is used as a core component of the V↔I↔I data construction pipeline to generate editing instructions for training data. This usage is non-standard and directly impacts the training data; it is described in Section 4 and Appendix B. Seedream 4.5 is used as the image editor to produce the per-frame before–after pairs in the same pipeline.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.

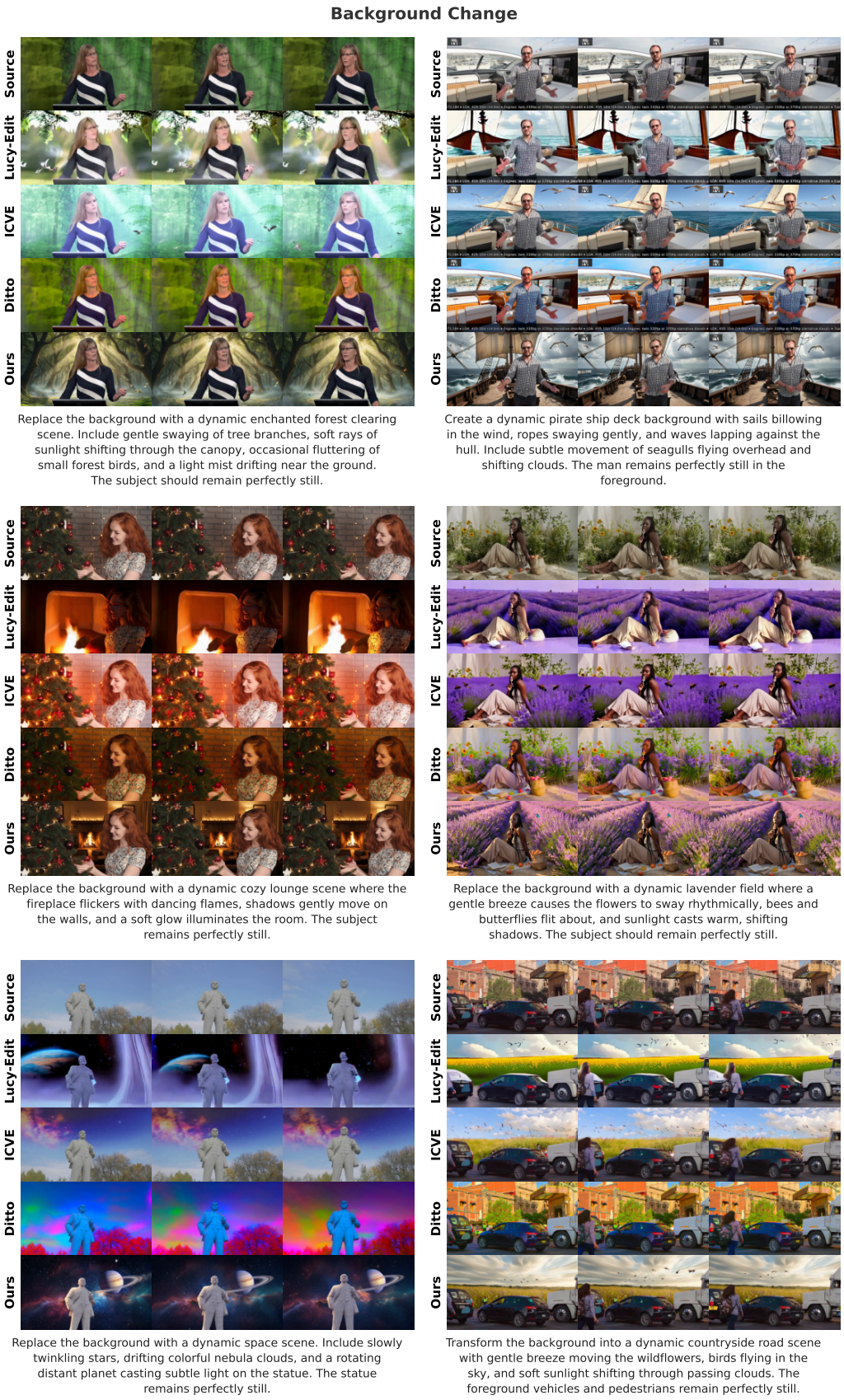


Figure 10: Qualitative comparison on **background change** editing.

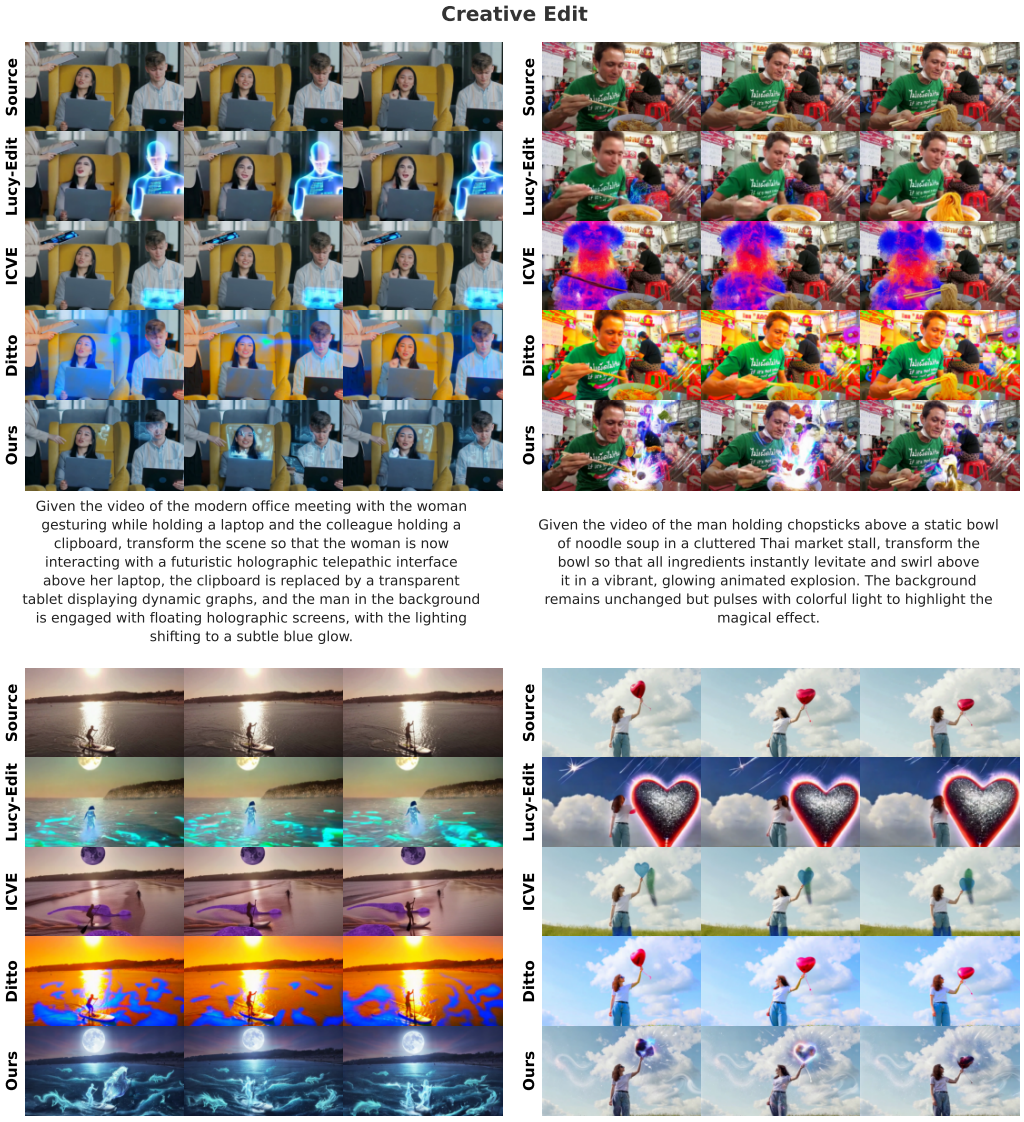


Figure 11: Qualitative comparison on **creative edit** editing.

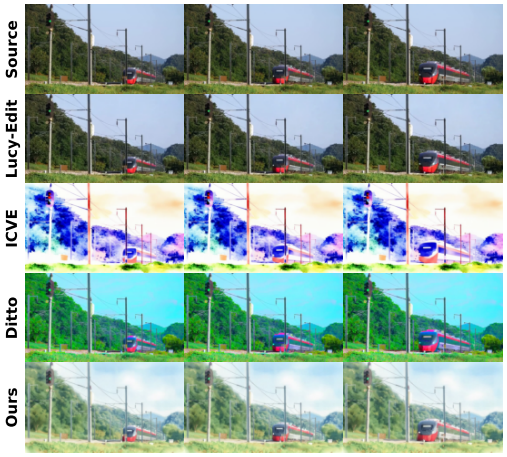
Global Style



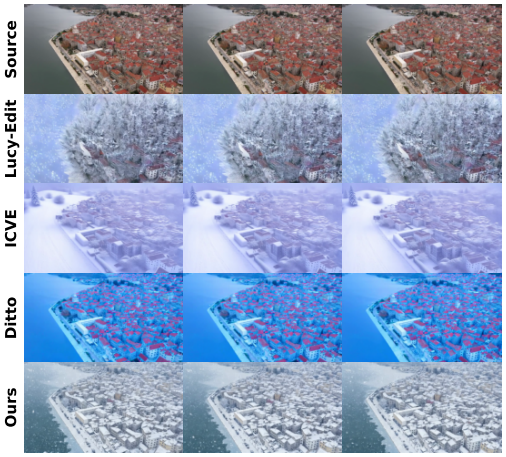
Apply the sunny style to this video, ensuring seamless, frame-by-frame consistency throughout. The final output should feature warm, golden-hour lighting, soft shadows, and vivid color typical of a bright, sunny day. Maintain all original motion, character actions, and camera movements, preserving the video's temporal flow, narrative integrity, and visual coherence.



Apply the traditional Gongbi painting style to this video, ensuring seamless temporal consistency across all frames. The final output should exhibit the signature characteristics of Gongbi—fine linework, harmonious color transitions, and layered ink effects—while preserving original motion sequences, character animations, and camera movements. Motion blur, lighting shifts, and narrative pacing must remain unchanged to maintain the video's natural flow.



Apply the Watercolor animation style to this video, ensuring seamless frame-by-frame consistency. The final output should exhibit the characteristic softness of watercolor, with gentle color gradients and subtle, ethereal blending across all frames, while perfectly maintaining the original motion, character actions, camera movements, and narrative flow—no flickering or jarring style changes allowed.



Apply the immersive Snowy style to this video, ensuring seamless temporal consistency across all frames. The final output should evoke a winter wonderland, complete with dynamic snowfall, frosty textures, and soft, diffused lighting. All original motion, character actions, and camera movements must be precisely preserved.



Apply the Pop Art animation style to this video, ensuring seamless temporal consistency across all frames. The result should mirror the aesthetic of classic Pop Art, with vivid colors, repetitive patterns, and Ben-Day dots integrated smoothly into each frame. Preserve the original motion, character movements, camera angles, and narrative flow without any abrupt or jarring style changes.

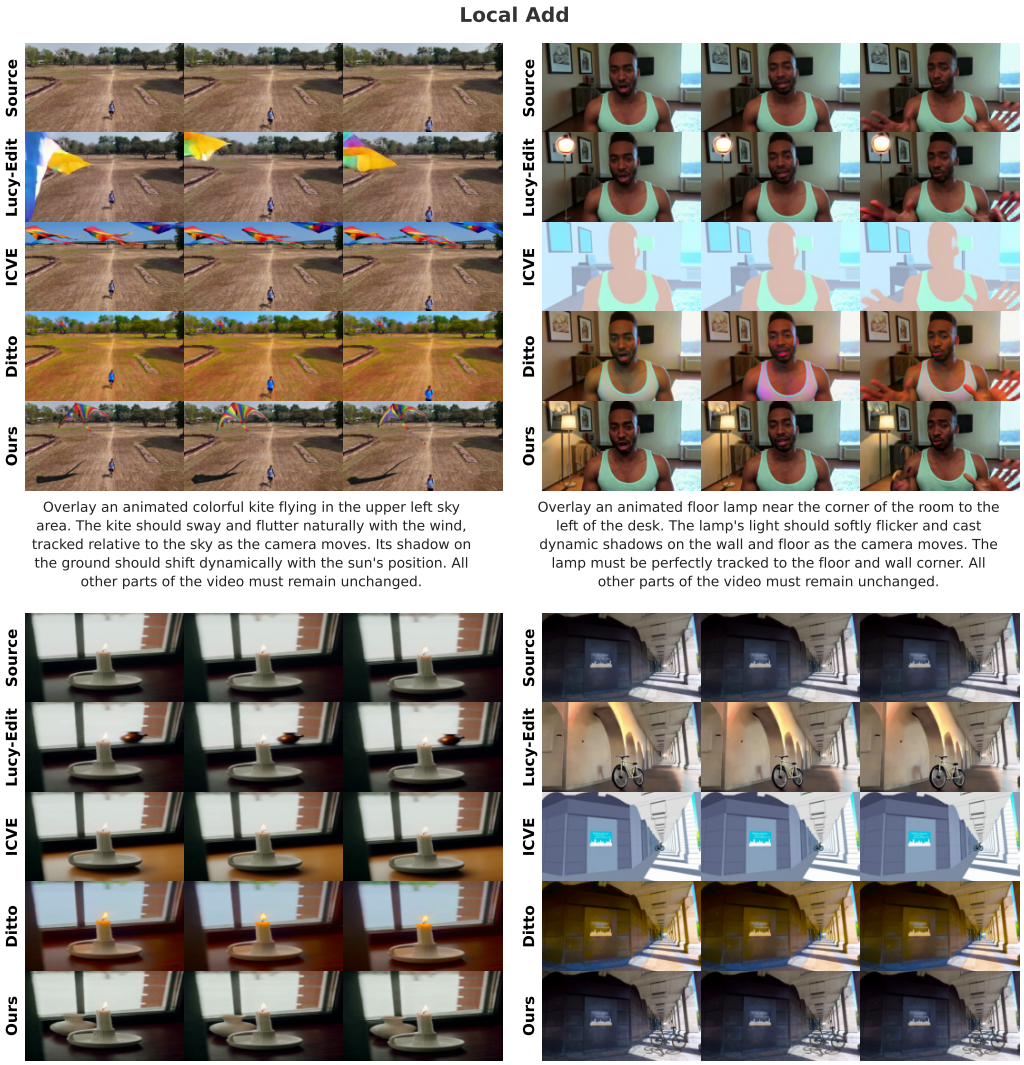


Figure 13: Qualitative comparison on **local add** editing.

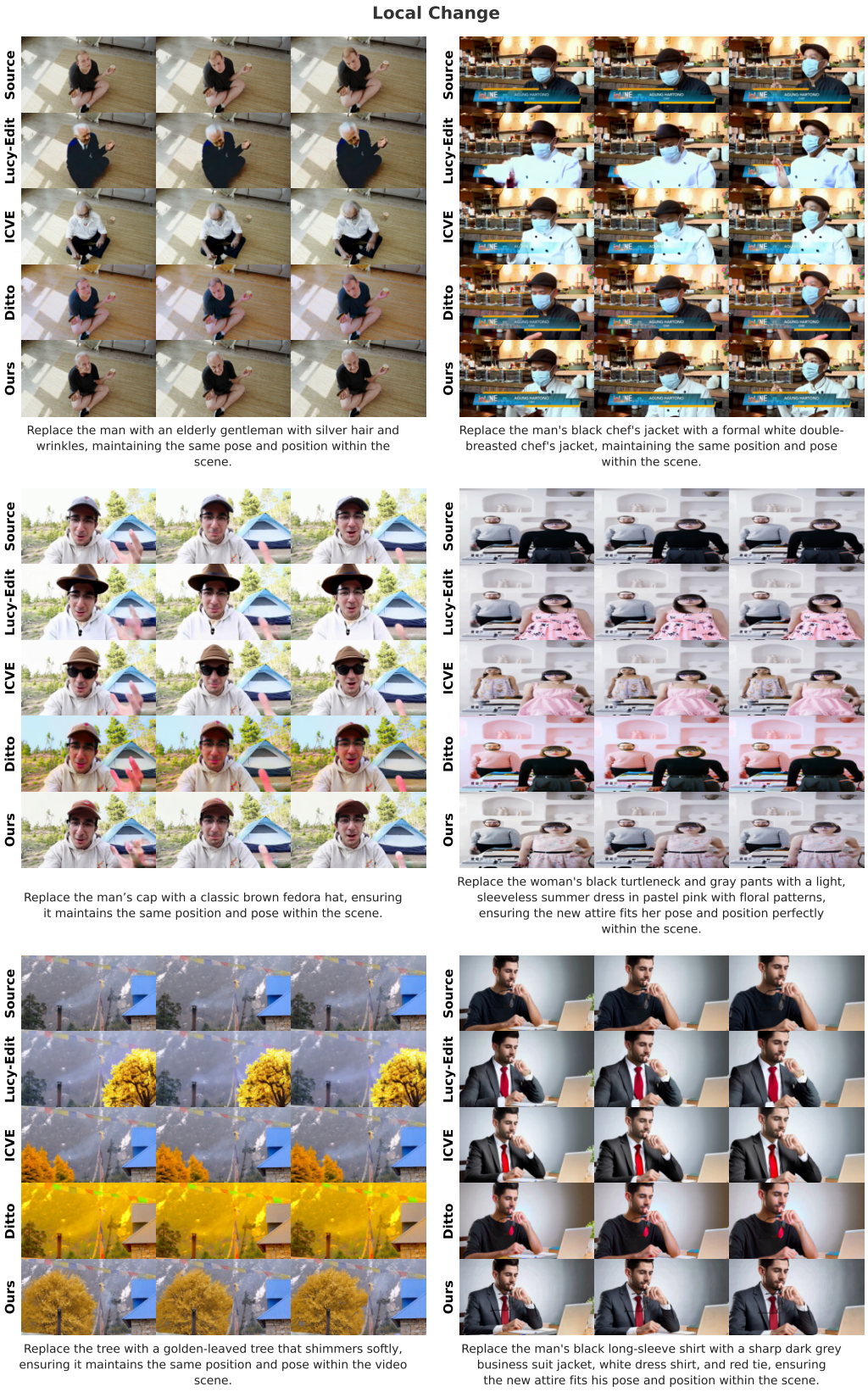
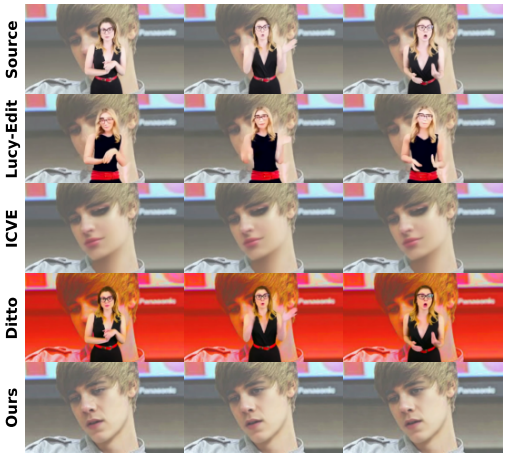


Figure 14: Qualitative comparison on **local change** editing.

Local Remove



Remove the woman with long, wavy blonde hair, glasses, a sleeveless black dress cinched at the waist with a vibrant red belt and prominent buckle from the entire video sequence. The background must be reconstructed with temporal consistency to match the original context, and all other video content must remain unchanged.



Remove the woven wicker basket with a sturdy, arched handle, delicate white lace trim, and a small decorative bow tied at the front from the entire video sequence. The background must be reconstructed with temporal consistency, and all other video content must remain unchanged.



Remove the young boy wearing a vibrant red beanie, a colorful jacket adorned with green and yellow circles, standing with a relaxed yet attentive posture, displaying curiosity through a slight head turn, with the hood resting against his back, steady gaze, slightly open mouth, neatly styled hair, and maintaining a consistent stance from the entire video sequence. The background must be reconstructed with temporal consistency, and all other video content must remain unchanged.



Remove the person with long, flowing red hair wearing a light brown, textured coat with a wide collar and tied waist belt from the entire video sequence. The background must be reconstructed with temporal consistency, and all other video content must remain unchanged.



Remove the off-road vehicle (dune buggy) from the entire video. This includes its robust design, large rugged tires, dark matte body, protective roll bar, red taillights, driver's seat with protective cage, visible suspension system, and shock absorbers. The background must be reconstructed with temporal consistency, ensuring all other video content remains unchanged.



Remove the man with short, neatly styled hair, a beard, wearing a black t-shirt with a prominent logo on the front, standing behind a desk with his right arm resting on the surface gesturing slightly as he speaks, left hand placed on the desk with fingers slightly curled, an attentive expression with eyes focused forward, occasionally pointing with his index finger, smooth and deliberate movements, and a calm and confident presence from the entire video sequence. The background must be reconstructed with temporal consistency, and all other video content must remain unchanged.



Figure 16: Qualitative comparison on **subtitle edit** editing.



Figure 17: Qualitative comparison of three inference modes of VINCIE-NExT on subject replacement (**left**) and subject removal (**right**): V2V direct inference, CoE with self-editing, and CoE with an external image editor.