VicEdit: Learning to Edit Videos from Visual In-Context Examples

Yuji Wang[1*] **Teng Hu**[1*] **Yuheng Chen**[1] **Ran Yi**[1†] **Han Feng**[2] **Weijian Cao**[2]
Chengjie Wang[2] **Lizhuang Ma**[1‡] **Jiangning Zhang**[3]
[1]Shanghai Jiao Tong University [2]Tencent Youtu Lab [3]Zhejiang University
Project Page: https://rain152.github.io/VicEdit/

Abstract

Despite remarkable progress in instruction-based video editing, textual descriptions remain limited in expressing fine-grained appearance, spatial transformations, and temporal dynamics. We introduce Visual In-context Editing, a new paradigm that shifts video editing from text-only instructions to visual demonstrations, where users specify desired edits through heterogeneous visual references including single images, image pairs, and video pairs. To establish this paradigm, we curate VicEdit-400K, the first large-scale dataset for visual in-context video editing, containing 400K high-quality samples across ten editing tasks with comprehensive semantic and visual consistency filtering. Building upon this foundation, we propose VicEdit, a unified framework that enables effective reasoning over diverse visual contexts and textual instructions. Specifically, we introduce Modality-Adaptive Semantic Distillation to extract modality-aware visual editing semantics and Dual-Context Injection to align visual contexts with textual editing intents. Extensive experiments on VicEdit-Bench demonstrate that VicEdit achieves state-of-the-art performance on both conventional instruction-based editing and visual in-context editing. Further controlled studies and human evaluations verify that VicEdit effectively leverages heterogeneous visual references, establishing visual in-context editing as a promising paradigm for controllable video editing.

1 Introduction

Instruction-based video editing [2–6] has made remarkable progress, enabling users to modify video content through natural language with increasing controllability. Built upon advances in large language models and diffusion architectures [7–9], recent methods support a broad range of editing operations, from localized object manipulation to global style transfer. However, textual instructions remain limited in expressing the fine-grained visual details, precise spatial transformations, and complex temporal dynamics required for high-fidelity editing. Describing a desired appearance, an object's exact transformation, or a complex motion pattern in words can be challenging, even when the intended result is visually clear.

As illustrated in Fig. 1 (Top), heterogeneous visual references offer complementary forms of editing guidance. A *Single Image* provides direct cues about the desired appearance and style, an *Image Pair* illustrates the visual and spatial changes between two states, and a *Video Pair* conveys temporal patterns that are difficult to specify precisely through text alone. Although existing methods have explored text-only and reference-guided editing, their ability to exploit heterogeneous visual references remains limited. Moreover, existing benchmarks [10, 11, 1] provide insufficient coverage of

*Equal contributions.
†Project lead.
‡Corresponding author.

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

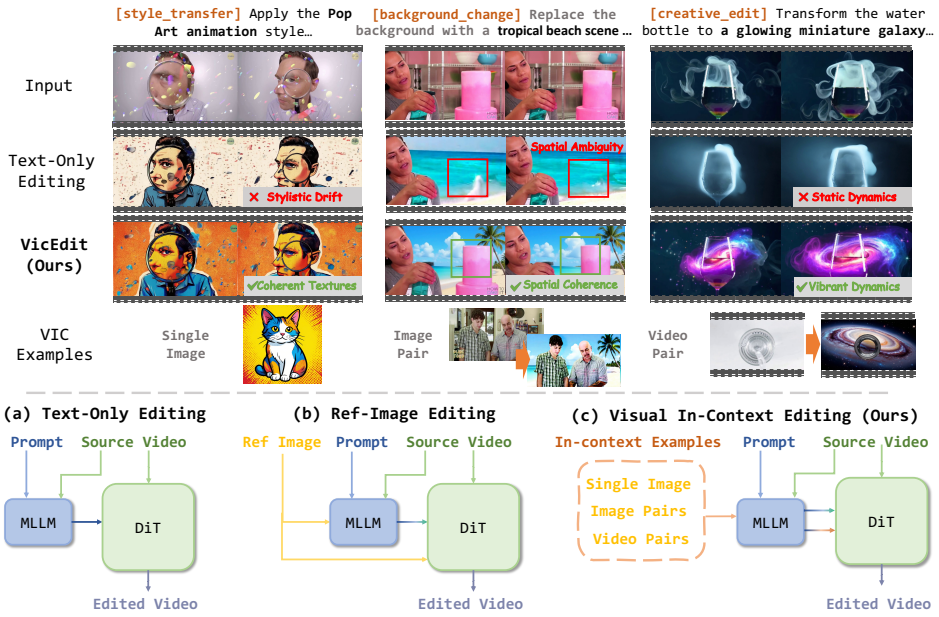

Figure 1: **Top:** Qualitative comparison of VicEdit with text-only baselines. Single-image, image-pair, and video-pair references help mitigate stylistic drift, spatial inaccuracies, and insufficient motion guidance, respectively. **Bottom:** Video editing paradigms: (a) text-only guidance; (b) visual reference conditioning via auxiliary branches; and (c) VicEdit, which distills modality-aware editing semantics from heterogeneous references and fuses them with text instructions within the video DiT. All examples are from OpenVE [1].

diverse reference modalities for systematically studying this setting. These limitations motivate a shift from describing edits through text alone to demonstrating them through visual examples, which we formulate as **Visual In-context Editing**.

Visual in-context editing is a unified video editing paradigm in which visual demonstrations complement textual instructions to communicate the desired edit. Rather than restricting users to a single reference modality, it accommodates Single Image, Image Pair, and Video Pair inputs, enabling different forms of visual evidence to guide the editing process. Realizing this paradigm presents two key challenges: extracting relevant editing semantics from heterogeneous references, whose information varies across modalities, and integrating these visual semantics with textual instructions while preserving their complementary roles.

To address these challenges, we introduce **VicEdit-400K** (Sec. 3.2), the first large-scale dataset specifically designed for visual in-context video editing. It comprises 400K high-quality samples across three reference modalities and ten editing tasks, curated through an automated pipeline with structured instruction generation and quality filtering. Building upon this dataset, we propose **VicEdit** (Sec. 3.3), a unified framework for editing with heterogeneous visual references. Specifically, **Modality-Adaptive Semantic Distillation (MASD)** uses modality-conditioned learnable queries to extract modality-aware editing semantics from visual references. **Dual-Context Injection (DCI)** then integrates the resulting visual context tokens with textual instructions into the video diffusion transformer (DiT), enabling the generation process to leverage both sources of editing information.

We evaluate VicEdit on **VicEdit-Bench**, which assesses visual in-context editing across the three reference modalities, as well as on conventional instruction-based video editing tasks. Extensive experiments demonstrate that VicEdit achieves state-of-the-art performance in both settings. Controlled studies and human evaluations further examine its ability to leverage heterogeneous visual references, while ablation studies validate the contributions of its key components.

Our contributions are summarized as follows:

1) We formulate a unified visual in-context video editing paradigm, transitioning from descriptions to visual demonstrations. To support this, we introduce **VicEdit-400K**, the first large-scale dataset

featuring 400K high-quality samples across three reference modalities and ten tasks, providing a systematic foundation for visual in-context editing.

2) We propose the **VicEdit** framework designed for heterogeneous visual references. Specifically, we introduce **Modality-Adaptive Semantic Distillation** to adaptively extract editing semantics via modality-conditioned queries, and **Dual-Context Injection** to synergistically fuse distilled visual tokens with textual instructions into DiT via cross-attention.

3) Experiments demonstrate that VicEdit not only excels in base text-only editing but also shows significant superiority on our proposed **VicEdit-Bench**, which comprehensively evaluates visual in-context editing across three reference modalities. Extensive ablation studies further validate the efficacy of each architectural component.

2 Related Work

Instruction-driven Video Editing. Instruction-driven video editing has evolved from training-free propagation [12–14] to large-scale trained systems [2–4, 11]. Recently, several works have begun exploring visual references: Kiwi-Edit [10] introduces single reference images via fixed queries, OmniTransfer [15] utilizes video references for generation, and EditVerse [16] encodes task types through in-context tokens. Despite these efforts, a unified framework supporting heterogeneous reference modalities remains absent. VicEdit addresses this gap by proposing a visual in-context paradigm that extends editing from text-only instructions to multi-modal guidance across single images, image pairs, and video pairs. To support this paradigm, we construct VicEdit-400K, the first large-scale dataset covering three reference modalities and ten task types with 400K samples.

In-Context Learning in Video Generation. In-context learning (ICL) [17] has emerged as a powerful paradigm where models learn to perform tasks by conditioning on demonstrations rather than explicit specifications [17]. In image generation, methods like Painter [18] and ImageBrush [19] demonstrate that visual exemplars enable more precise control than text alone. This paradigm has since extended to video: early works [20] explored prompt-level ICL for multi-scene tasks, while VICL [21] utilized an autoregressive DiT for viewpoint and motion transfer. Recent methods like EditVerse [16, 5] leverage global attention for context interaction, yet struggle with long-sequence references. Alternatively, OmniTransfer [15] employs an auxiliary DiT to learn from reference examples. In summary, existing methods are typically confined to video generation and support only limited reference modalities. VicEdit advances this field by introducing Modality-Adaptive Semantic Distillation to adaptively extract editing semantics from diverse references, and Dual-Context Injection to synergistically integrate these visual cues with textual instructions within the DiT latent space.

3 Method

3.1 Formulation: Visual In-Context Editing

We define the task of visual in-context video editing as a conditional generation process [16, 20]. Given a source video $V_{src} \in \mathbb{R}^{T\times 3\times H\times W}$ and a textual instruction P, the goal is to synthesize an edited video V_{tgt} that manifests the editing intentions through an optional visual reference R. Formally, this transformation is represented as:

$$V_{tgt} = \mathcal{G}(V_{src}, P, \langle R\rangle), \tag{1}$$

where $\mathcal{G}$ denotes the generative model and $\langle\cdot\rangle$ represents an optional conditioning input [20, 22]. To encompass diverse editing scenarios, R can take one of three reference modalities, each conveying complementary editing semantics:

- **Single Image** ($R_{si} = I_{ref}$): Provides fine-grained visual anchors, including textures and appearances, to guide stylistic or identity-consistent synthesis.
- **Image Pair** ($R_{ip} = \{I_s^{ref}, I_t^{ref}\}$): Characterizes the transformation between reference domains, capturing shifts in spatial layouts and localized details.
- **Video Pair** ($R_{vp} = \{V_s^{ref}, V_t^{ref}\}$): Provides dense temporal priors and dynamic semantic correspondences for motion-intensive editing.

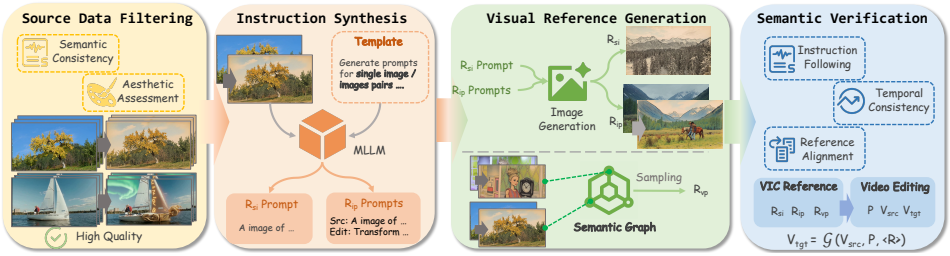

Figure 2: **Data construction pipeline** that comprises four stages (Sec. 3.2), yielding a final dataset of 400K samples that covers conventional instruction-based editing and visual in-context editing tasks. All examples presented in the figure are collected from the open-source OpenVE [1].

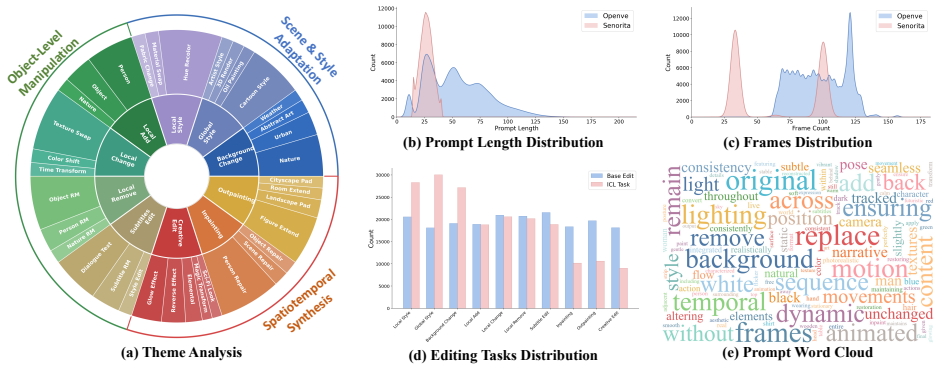

Figure 3: Statistics of VicEdit-400K.

This paradigm faces two challenges: the scarcity of large-scale unified datasets across these modalities and the structural heterogeneity that complicates semantic extraction and multimodal integration.

3.2 Data Curation for Visual In-Context Editing

In this section, we present the construction pipeline of VicEdit-400K. We curate 400K high-quality samples by synthesizing editing instructions, generating corresponding visual references, and applying multi-stage quality filtering. Based on this dataset, we further establish VicEdit-Bench to rigorously evaluate performance across diverse visual in-context editing tasks. We provide further details regarding the dataset and benchmark in Appendices B and C.

Data Source. We build VicEdit-400K upon two recent large-scale datasets: Señorita-2M [23], which covers 18 editing sub-categories, and OpenVE-3M [1], containing approximately 3 million samples across 8 spatial and non-spatial categories. While extensive, both datasets rely solely on textual instructions and lack visual references. To address this, we systematically consolidate their editing tasks and curate a subset to construct multi-modal visual references.

Data Construction Pipeline. As shown in Fig. 2, our construction pipeline comprises four stages:

1) Source Data Filtering. We filter raw data for semantic consistency and aesthetic quality [24, 25]. By leveraging the LAION-Aesthetic [26] predictor and Qwen3-VL [27], we implement a dual-model filtering mechanism that selects only those samples possessing both superior visual appeal and high semantic editability for the downstream pipeline.

2) Instruction Synthesis. Leveraging MLLMs [28, 27], we synthesize modality-aware instructions tailored to each reference type. By employing structured prompting, we ensure the instructions maintain strict semantic alignment with the visual references while explicitly prescribing non-trivial edits that differentiate the target from the source content. This approach yields a diverse prompt set spanning 10 core tasks, ensuring both consistency and editability.

3) Visual Reference Generation. We employ modality-specific strategies for reference synthesis. For single-image and image-pair references, we utilize state-of-the-art generative models [29, 30] to synthesize target content guided by the generated instructions. For video-pair references, to circumvent temporal artifacts in generative editing, we construct a semantic knowledge graph and

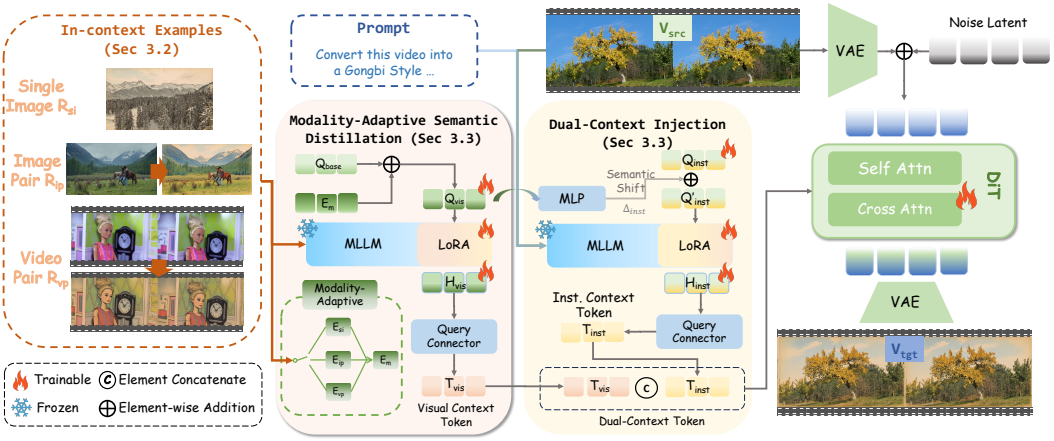

Figure 4: **Overview of VicEdit.** ***1)*** **MASD** modulates base queries Q_{base} with modality embeddings E_m to distill visual context tokens T_{vis}. ***2)*** **DCI** applies a modality-conditioned shift to instruction queries. The resulting visual and instruction context tokens are concatenated and injected into the video DiT to guide generation.

partition the task space via spectral clustering [31]. By sampling semantically-linked pairs from these clusters, we provide authentic references that maintain high visual fidelity.

4) Semantic Verification. Final samples undergo MLLM-based multi-dimensional verification. By employing pre-defined scoring templates, we prompt MLLMs to rigorously assess instruction following, temporal consistency [10, 1], and reference alignment. This standardized evaluation allows us to filter out inconsistent samples and ensure the high reliability of the final dataset.

Data Statistics and Analysis. VicEdit-400K covers three categories and 10 fine-grained tasks as shown in Fig. 3. Statistical analysis reveals significant diversity, with video lengths primarily concentrated in the 30∼120 frame intervals. This bimodal distribution, coupled with a broad semantic prompt space (Fig. 3e), provides a robust foundation for advancing visual in-context editing.

Benchmark and Evaluation. We evaluate VicEdit on two paradigms: *1)* Instruction-based Editing, where we introduce OpenVE-Ext by augmenting OpenVE-Bench [1] with 3 additional tasks from Señorita [23]; and *2)* Visual In-Context Editing, using our proposed VicEdit-Bench to assess heterogeneous reference following. VicEdit-Bench consists of 200 representative instances (10 per task-modality combination) strictly isolated from the training set for held-out evaluation. Following [1], we leverage VLMs [27, 32] for automated scoring, supplemented by fidelity and temporal metrics from VBench [24, 25] and IVE-bench [33].

3.3 Methodology: Visual In-context Editing (VicEdit)

Existing text-only video editing frameworks are inherently task-limited and fail to generalize to complex editing scenarios requiring fine-grained spatial and semantic constraints. They cannot adapt to emerging in-context editing paradigms that rely on multi-modal visual guidance. Therefore, we introduce VicEdit, a unified framework for visual in-context video editing that extends video editing to accommodate multi-modal visual guidance.

Overview of VicEdit. As illustrated in Figure 4, VicEdit employs an MLLM for semantic understanding and a DiT for generation, connected by two lightweight modules: **MASD**, which extracts modality-aware visual context tokens $\mathbf{T}_{\text{vis}}$ from the in-context reference, and **DCI**, which produces instruction context tokens $\mathbf{T}_{\text{inst}}$ using modality-conditioned instruction queries. $\mathbf{T}_{\text{vis}}$ and $\mathbf{T}_{\text{inst}}$ encode complementary perceptual editing semantics and task-level edit directions. These token sequences are concatenated as cross-attention keys and values for the DiT to generate $\mathbf{V}_e$.

Modality-Adaptive Semantic Distillation. The three reference modalities capture structurally distinct editing semantics: image pairs reflect spatial transformations, video pairs provide temporal motion priors, and single images anchor global appearance attributes. Existing methods [22, 34, 10] often extract semantic tokens using a fixed set of learnable queries, implicitly assuming a universal

pattern for all visual cues. However, this assumption fails under heterogeneous modalities, as a shared query cannot adaptively distinguish between the localized spatial shifts in an image pair versus the global stylistic textures in a single reference image.

To explicitly account for reference modality, MASD introduces an additive, modality-conditioned offset:

$$\mathbf{Q}_{\text{vis}} = \mathbf{Q}_{\text{base}} + \mathbf{E}_m, \quad m \in \{\text{ip}, \text{vp}, \text{si}\}, \tag{2}$$

where $\mathbf{Q}_{\text{base}} \in \mathbb{R}^{N \times d}$ denotes learnable queries shared across reference modalities, and $\mathbf{E}_m \in \mathbb{R}^{N \times d}$ is a learnable modality-specific embedding. This parameterization combines shared queries with modality-dependent offsets, allowing the extraction to adapt to heterogeneous reference types.

We append $\mathbf{Q}_{\text{vis}}$ to the reference input and process the sequence through the MLLM. The output hidden states at the query positions, denoted by $\mathbf{H}_{\text{vis}}$, are projected into N visual context tokens through an MLP Query Connector:

$$\mathbf{T}_{\text{vis}} = \text{MLP}_{\text{vis}}(\mathbf{H}_{\text{vis}}). \tag{3}$$

The connector maps the query hidden states into the DiT conditioning space. While the query embeddings depend on the reference modality, the resulting tokens also capture instance-specific visual content. The shared queries and modality embeddings are learned under the editing objective.

Dual-Context Injection. DCI couples visual and instruction extraction through their query parameterizations. Specifically, it maps the modality-conditioned visual queries to an instruction-query shift:

$$\mathbf{\Delta}_{\text{inst}} = \text{MLP}_{\text{shift}}(\mathbf{Q}_{\text{vis}}) \in \mathbb{R}^{N \times d}, \tag{4}$$

which is added to the learnable instruction queries:

$$\mathbf{Q}'_{\text{inst}} = \mathbf{Q}_{\text{inst}} + \mathbf{\Delta}_{\text{inst}}. \tag{5}$$

Since $\mathbf{Q}_{\text{vis}} = \mathbf{Q}_{\text{base}} + \mathbf{E}_m$, the shift depends on the reference modality and is shared across instances of the same modality at inference. Without MASD, it reduces to a global learned shift. During training, the shared query parameters receive gradients through both visual extraction and instruction-query modulation, coupling the two branches.

We append $\mathbf{Q}'_{\text{inst}}$ to the text prompt and process the sequence through the MLLM. The output hidden states at the query positions, denoted by $\mathbf{H}_{\text{inst}}$, are projected into instruction context tokens through an MLP Query Connector:

$$\mathbf{T}_{\text{inst}} = \text{MLP}_{\text{inst}}(\mathbf{H}_{\text{inst}}). \tag{6}$$

Finally, we concatenate the two token sequences as $\mathbf{T}_{\text{dual}} = [\mathbf{T}_{\text{vis}}; \mathbf{T}_{\text{inst}}]$ and inject them as keys and values into the video DiT cross-attention layers. This supplies the generator with instance-specific visual evidence and modality-conditioned instruction representations.

Training Strategy. VicEdit builds upon Wan-TI2V-5B [8] and inherits pretrained weights from Kiwi-Edit. VicEdit is trained in three stages to progressively incorporate in-context editing capability.

1) Base Editing. We first fine-tune the pretrained video DiT on instruction-based editing tasks to expand its editing capabilities while promoting instruction following and temporal consistency. The MLLM is initialized from pretrained weights and kept frozen.

2) In-Context Branch. With the DiT frozen, we train the in-context modules, including MASD, the semantic shift MLP in DCI, and the instruction queries $\mathbf{Q}_{\text{inst}}$, together with rank-64 LoRA adapters in the MLLM language component. Each training batch mixes samples from all three modalities to ensure balanced learning of the modality embeddings.

3) Joint Fine-Tuning. We unfreeze the DiT and jointly fine-tune the entire pipeline (except the MLLM) to align the visual conditioning signals with the generative process. All stages are optimized with the standard denoising loss:

$$\mathcal{L} = \mathbb{E}_{\mathbf{V}_e, \boldsymbol{\epsilon}, t}\left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{V}_e^{(t)}, t, \mathbf{T}_{\text{dual}})\|^2\right], \tag{7}$$

where $\mathbf{V}_e^{(t)}$ is the noised edited video at timestep t, $\boldsymbol{\epsilon}$ is the ground-truth noise, and $\boldsymbol{\epsilon}_\theta$ denotes the DiT noise predictor conditioned on $\mathbf{T}_{\text{dual}}$.

Table 1: Quantitative Comparison on OpenVE-Ext with Gemini 2.5 Pro [32].

Methods	Overall	Scene & Style Adaptation			Object-Level Manipulation				Spatiotemporal Synthesis		
		Global Style	Local Style	Background Change	Local Change	Local Remove	Local Add	Subtitle Edit	Creative Edit	Inpainting	Outpainting
VACE [35]	1.68	1.49	1.39	1.55	2.07	1.46	1.26	1.48	1.47	2.31	2.35
Lucy-Edit [36]	2.35	2.27	2.02	1.57	3.20	1.75	2.30	1.61	2.86	3.01	2.92
DITTO [37]	2.31	**4.01**	**3.67**	1.68	2.03	1.53	1.41	2.81	1.23	2.05	2.67
NovaEdit [3]	2.76	2.24	1.98	2.40	3.62	3.43	2.64	2.93	2.59	2.78	3.01
VideoCoF [4]	2.79	2.57	2.24	2.32	3.37	3.61	**2.75**	3.10	2.20	2.87	2.88
Kiwi-Edit [10]	3.08	3.64	2.85	2.64	3.83	2.63	2.36	3.20	**3.39**	3.35	2.86
VicEdit (Ours)	**3.28**	3.48	2.90	**3.05**	**3.85**	**3.68**	2.64	**3.32**	3.25	**3.38**	**3.20**

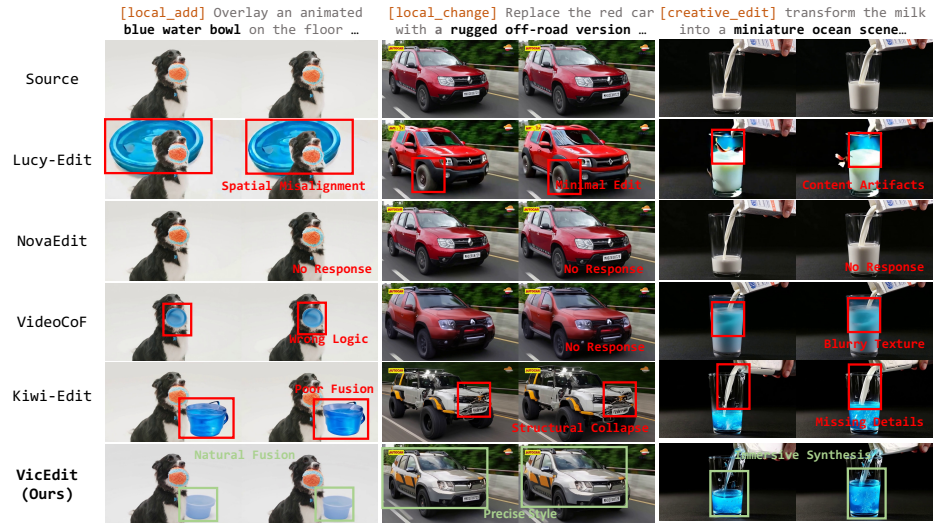

Figure 5: Comparison with baselines on instruction-based editing tasks.

4 Experiments

4.1 Base Video Editing

Quantitative Analysis. We compare VicEdit with six video editing methods [35–37, 3, 4, 10] on OpenVE-Ext. Table 1 reports an Overall score of 3.28 for VicEdit and 3.08 for Kiwi-Edit. Performance varies by task: for example, DITTO is strong on global and local style, while VicEdit achieves the highest reported scores on background change, local change, local removal, subtitle editing, inpainting, and outpainting. These text-only results characterize the trained system's editing ability; they do not independently isolate the contributions of the visual context modules. Component ablations are reported in Table 4.

Qualitative Analysis. Fig. 5 presents several qualitative examples. As shown in the comparisons, our method achieves superior performance in spatial layout preservation, semantic comprehension, and visual quality. For instance, in the "local_add" task, baselines struggle to handle the spatial relationships between the newly added object and existing objects in terms of position and scale, whereas VicEdit produces well-structured editing results. In the "local_change" task, most baselines fail to understand the semantic transformation, and Kiwi-Edit, while demonstrating semantic understanding, introduces noticeable artifacts. In the "creative_edit" example, videos generated by our method are more semantically coherent and logically consistent with the given prompts.

4.2 Visual In-Context Video Editing

Quantitative Analysis. To comprehensively evaluate the effectiveness of our proposed task and model, we conduct experiments on the VicEdit-Bench benchmark. Crucially, unlike prior works that rely solely on textual instructions, our evaluation setting incorporates visual in-context examples as additional references, which is the key innovation of our VIC task. For a fair comparison, we equip all

Table 2: Quantitative comparison on VicEdit-Bench with Gemini 2.5 Pro [32].

Methods	Overall	Scene & Style Adaptation			Object-Level Manipulation				Spatiotemporal Synthesis		
		Global Style	Local Style	Background Change	Local Change	Local Remove	Local Add	Subtitle Edit	Creative Edit	Inpainting	Outpainting
VACE [35]	1.85	1.45	2.03	1.50	2.25	1.49	1.33	1.43	2.17	2.23	2.60
Lucy-Edit [36]	2.25	1.94	1.67	1.89	3.30	2.05	2.15	1.43	2.45	2.63	2.97
DITTO [37]	2.33	3.03	**3.60**	2.29	2.45	1.64	1.43	2.61	1.46	2.48	2.32
NovaEdit [3]	2.61	2.08	2.23	2.62	3.07	3.65	2.05	2.46	2.13	2.65	3.18
VideoCoF [4]	2.90	2.66	2.43	2.37	3.67	3.60	2.44	**3.65**	1.83	3.06	3.33
Kiwi-Edit [10]	2.95	3.19	2.39	2.53	**4.29**	3.24	2.28	3.16	2.77	3.18	2.56
VicEdit (Ours)	**3.45**	**3.68**	3.03	**3.05**	4.17	**3.86**	**2.85**	3.62	**3.30**	**3.45**	**3.47**

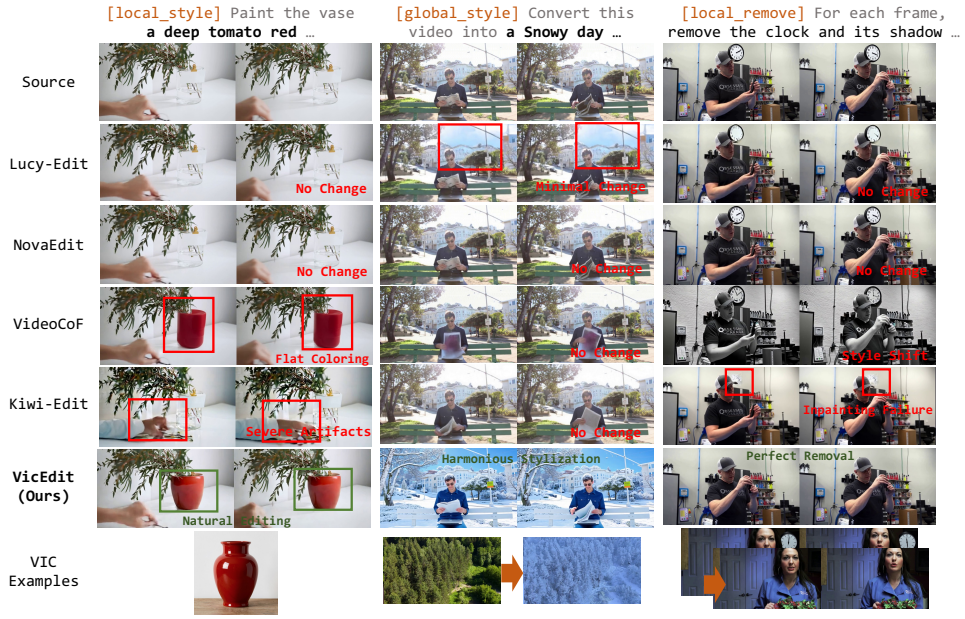

Figure 6: Comparison with baselines on visual in-context editing tasks. All examples are from OpenVE [1].

baselines with an instruction enhancement module to augment their text-only instructions. As shown in Table 2, VicEdit achieves a remarkable Overall score of 3.45. This performance surpasses the second-best method, Kiwi-Edit, by a substantial margin of 0.50 points (a 17% relative improvement). The significant gap underscores VicEdit's superior capability in understanding and executing complex editing instructions. In contrast, previous text-only approaches often struggle to capture precise semantics from text descriptions alone, leading to suboptimal editing results.

Qualitative Analysis. Qualitative results on the visual in-context task are shown in Figure 6. A key observation is that our model, conditioned on diverse visual contexts (single images, image pair, video pair), effectively captures the instruction semantics. Taking the **Global Style** task as an example, converting the scene to a snowy day proves challenging for existing text-only baselines, which struggle to synthesize a visually plausible snowy style. However, equipped with visual in-context example, our model seamlessly adapts to the desired style, highlighting its robust editing capability.

4.3 Human Evaluation

We conduct pairwise human preference studies on base editing and visual in-context editing, comparing VicEdit with the strongest baseline for each selected task, with ten annotators each evaluating 20 examples per setting. As shown in Table 3, VicEdit achieves tie-adjusted preference scores of 76% and 72%, respectively, with each tie contributing half a point to each method. Human judgments agree with Gemini 2.5 Pro at rates of 80% and 76%, while inter-annotator agreement reaches Fleiss' κ values of 0.70 and 0.66, respectively. Separately, Gemini 2.5 Pro and Qwen-VL scores have Pearson correlations of 0.76 and 0.74. These results provide complementary human evidence for VicEdit's

Table 3: Human preferences and evaluator agreement. Ten annotators each evaluate 20 examples per setting. Preference scores compare against the strongest task-specific baseline, awarding 1 point for wins and 0.5 for ties. Gemini–Qwen correlation measures judge–judge agreement separately.

Metric	Base Editing	VIC Editing
Human Preference Score (%) ↑	76	72
Human–Gemini Agreement (%) ↑	80	76
Gemini–Qwen-VL Pearson's r ↑	0.76	0.74
Fleiss' κ ↑	0.70	0.66

Table 4: Component ablation on VicEdit-Bench. VLM scores are mean ± standard deviation over three video-generation runs, using one fixed trained checkpoint per configuration. Shared-Query Base uses source-video and text conditioning without visual-reference tokens. The other configurations incorporate visual-reference tokens; Naive Concatenation combines them with instruction tokens by direct concatenation without MASD or DCI.

	VLM Evaluation				**VBench Metrics**				
Variant	Overall↑	Instr. Comp.↑	Detail Fid.↑	Visual Qual.↑	BC↑	MS↑	TF↑	IQ↑	AQ↑
Shared-Query Base	3.02 ± 0.05	3.12 ± 0.06	2.98 ± 0.06	2.96 ± 0.05	0.968	0.995	0.986	0.711	0.564
Naive Concatenation	3.14 ± 0.05	3.24 ± 0.05	3.09 ± 0.05	3.09 ± 0.06	0.970	0.993	0.985	0.712	0.566
MASD only	3.25 ± 0.04	3.40 ± 0.04	3.20 ± 0.05	3.14 ± 0.05	0.969	0.994	0.986	0.712	0.566
DCI only	3.34 ± 0.04	3.56 ± 0.03	3.22 ± 0.04	3.24 ± 0.04	0.971	0.995	0.986	0.715	0.567
MASD + DCI	$\mathbf{3.45 \pm 0.03}$	3.65 ± 0.03	3.34 ± 0.04	3.35 ± 0.03	**0.975**	0.994	**0.987**	**0.718**	**0.568**

effectiveness; the small study does not establish general reliability or judge independence of VLM scoring. Appendix D.3 describes the evaluation protocol.

4.4 Ablation Study

Ablation configurations. Table 4 compares five configurations. Shared-Query Base is a Kiwi-Edit-style fixed-query model fine-tuned on VicEdit-400K, conditioned on the source video and instruction without visual-reference tokens. Naive Concatenation adds visual-reference tokens using separate visual and instruction queries, concatenating $[\mathbf{T}_{\text{vis}}; \mathbf{T}_{\text{inst}}]$ for DiT conditioning without modality embeddings or DCI. MASD only adds modality embeddings to this baseline; DCI only adds query-shift coupling with $\mathbf{Q}_{\text{vis}} = \mathbf{Q}_{\text{base}}$. The full model combines both modules.

Ablation on MASD. Table 4 evaluates modality-conditioned queries under two configurations. Without DCI, adding MASD increases Overall from 3.14 to 3.25. With DCI enabled, adding MASD increases Overall from 3.34 to 3.45 and detail fidelity from 3.22 to 3.34. These results support the effectiveness of modality-specific query offsets for heterogeneous reference extraction under the evaluated setup. They do not establish an explicitly enforced decomposition of semantic and spatial information.

Ablation on DCI. DCI improves Overall from 3.14 to 3.34 without MASD, and from 3.25 to 3.45 with MASD. Without MASD, the shift $\mathrm{MLP}(\mathbf{Q}_{\text{base}})$ is reference-independent at inference and can be absorbed into freely learned instruction queries. During training, however, DCI couples visual extraction and instruction-query modulation through the shared query parameters. With MASD, the shift additionally depends on the reference modality, while instance-specific visual content is conveyed through $\mathbf{T}_{\text{vis}}$. The results support the effectiveness of this parameterization, although the precise optimization mechanism responsible for the gains is not isolated by these ablations.

The synergistic integration of both modules achieves the best overall performance, with detailed visual comparisons provided in Appendix E.5.

Ablation on Training Strategy. As shown in Table 5, Stage 1 broadens the range of editing tasks, improving both base and visual in-context editing across all metrics. Stage 2 focuses on visual in-context learning, further improving VIC performance while reducing base editing performance. Finally, Stage 3 jointly trains both tasks, recovering base editing performance and further improving VIC editing, yielding the highest overall scores in both settings.

Table 5: Quantitative Ablation on Training Stages. We evaluate the performance evolution across different training phases on the OpenVE-Ext and VicEdit-Bench benchmarks.

Training Stage	Base Editing				Visual In-Context Editing			
	Overall↑	Instr. Comp.↑	Detail Fid.↑	Visual Qual.↑	Overall↑	Instr. Comp.↑	Detail Fid.↑	Visual Qual.↑
Base (Kiwi-Edit)	3.08	3.24	3.01	2.98	2.95	3.10	2.98	2.76
Stage 1	3.23	3.36	3.11	3.20	3.10	3.25	3.05	3.01
Stage 2	3.10	3.22	3.02	3.07	3.40	3.62	3.30	3.27
Stage 3	**3.28**	3.47	3.27	3.10	**3.45**	3.65	3.34	3.35

Table 6: Separating the Contributions of Training Data and Model Architecture.

Method	Overall ↑	Instr. Comp. ↑	Detail Fid. ↑	Visual Qual. ↑
Kiwi-Edit (original)	2.95	3.10	2.98	2.76
Kiwi-style + VicEdit-400K	3.14	3.24	3.09	3.09
VicEdit (full)	**3.45**	**3.65**	**3.34**	**3.35**

Table 7: Ablation on Visual Reference Effectiveness. We compare correct, mismatched, and absent visual references to evaluate their impact on editing performance and reference adherence.

Setting	Instr. Comp. ↑	Detail Fid. ↑	Visual Qual. ↑	Ref. Adherence ↑
Text-only (no reference)	3.12	3.08	3.10	–
Mismatched reference	2.90	2.82	2.68	1.68
Correct reference	**3.58**	**3.28**	**3.28**	**3.42**

Ablation on Dataset and Architecture Contributions. To disentangle the contributions of VicEdit-400K and our proposed architecture, we compare the original Kiwi-Edit, a Kiwi-style variant trained on VicEdit-400K, and the full VicEdit model. Replacing the original training data with VicEdit-400K improves the overall score from 2.95 to 3.14, demonstrating the benefit of the proposed dataset. With the training data and experimental setup held constant, incorporating MASD and DCI further increases the score to 3.45. The full model also achieves consistent gains in instruction compliance, detail fidelity, and visual quality. These results show that the improvement comes from both the dataset and the proposed architecture, with their combination yielding the strongest overall performance.

Ablation on Visual Reference Effectiveness. To assess whether VicEdit uses visual references beyond textual instructions, we compare correct, mismatched, and no-reference settings. Correct references improve instruction compliance, detail fidelity, and visual quality over text-only conditioning, whereas mismatched references reduce all three metrics below the text-only baseline. Reference adherence also drops substantially with mismatched references. This pattern indicates that visual references actively influence the editing process: relevant references provide useful guidance, while misleading ones can interfere with instruction execution. The benefits of visual conditioning therefore depend on reference relevance.

5 Conclusion

We present VicEdit, a unified framework extending text-only video editing to multi-modal visual guidance. Modality-Adaptive Semantic Distillation (MASD) extracts editing semantics through modality-conditioned learnable queries, while Dual-Context Injection (DCI) couples visual and textual contexts through a semantic shift, enabling fine-grained control across single images, image pairs, and video pairs. We construct VicEdit-400K, the first large-scale dataset for visual in-context video editing, with 400K samples across ten tasks, and introduce VicEdit-Bench for standardized evaluation. Extensive experiments demonstrate state-of-the-art performance on both instruction-based and visual in-context editing benchmarks, establishing a scalable paradigm for video editing. Our current design targets semantic-level editing; future work may explore pixel-level control and larger-scale datasets.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 72192821, 62302297, 62472282, 62272447, 62472285, 625B2115).

References

[1] 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.

[2] Jinjie Mai, Chaoyang Wang, Guocheng Gordon Qian, Willi Menapace, Sergey Tulyakov, Bernard Ghanem, Peter Wonka, and Ashkan Mirzaei. Easyv2v: A high-quality instruction-based video editing framework. *arXiv preprint arXiv:2512.16920*, 2025.

[3] Tianlin Pan, Jiayi Dai, Chenpu Yuan, Zhengyao Lv, Binxin Yang, Hubery Yin, Chen Li, Jing Lyu, Caifeng Shan, and Chenyang Si. Nova: Sparse control, dense synthesis for pair-free video editing. *arXiv preprint arXiv:2603.02802*, 2026.

[4] Xiangpeng Yang, Ji Xie, Yiyuan Yang, Yan Huang, Min Xu, and Qiang Wu. Unified video editing with temporal reasoner. *arXiv preprint arXiv:2512.07469*, 2025.

[5] Kinam Kim, Junha Hyung, and Jaegul Choo. Temporal in-context fine-tuning with temporal reasoning for versatile control of video diffusion models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[6] Wenhao Sun, Rong-Cheng Tu, Jingyi Liao, and Dacheng Tao. Diffusion model-based video editing: A survey. *arXiv preprint arXiv:2407.07111*, 2024.

[7] 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.

[8] 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.

[9] 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.

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

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

[12] 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.

[13] Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Tokenflow: Consistent diffusion features for consistent video editing. *arXiv preprint arXiv:2307.10373*, 2023.

[14] 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.

[15] Pengze Zhang, Yanze Wu, Mengtian Li, Xu Bai, Songtao Zhao, Fulong Ye, Chong Mou, Xinghui Li, Zhuowei Chen, Qian He, et al. Omnitransfer: All-in-one framework for spatio-temporal video transfer. *arXiv preprint arXiv:2601.14250*, 2026.

[16] 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.

[17] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, et al. A survey on in-context learning. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pages 1107–1128, 2024.

[18] Xinlong Wang, Wen Wang, Yue Cao, Chunhua Shen, and Tiejun Huang. Images speak in images: A generalist painter for in-context visual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6830–6839, 2023.

[19] Yifan Yang, Houwen Peng, Yifei Shen, Yuqing Yang, Han Hu, Lili Qiu, Hideki Koike, et al. Imagebrush: Learning visual in-context instructions for exemplar-based image manipulation. *Advances in Neural Information Processing Systems*, 36:48723–48743, 2023.

[20] Zhengcong Fei, Di Qiu, Changqian Yu, Debang Li, Mingyuan Fan, and Xiang Wen. Video diffusion transformers are in-context learners. *arXiv e-prints*, pages arXiv–2412, 2024.

[21] Wentao Zhang, Junliang Guo, Tianyu He, Li Zhao, Linli Xu, and Jiang Bian. Video in-context learning: Autoregressive transformers are zero-shot video imitators. In *The Thirteenth International Conference on Learning Representations*, 2025.

[22] 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.

[23] Bojia Zi, Penghui Ruan, Marco Chen, Xianbiao Qi, Shaozhe Hao, Shihao Zhao, Youze Huang, Bin Liang, Rong Xiao, and Kam-Fai Wong. Señorita-2m: A high-quality instruction-based dataset for general video editing by video specialists. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025.

[24] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21807–21818, 2024.

[25] Dian Zheng, Ziqi Huang, Hongbo Liu, Kai Zou, Yinan He, Fan Zhang, Lulu Gu, Yuanhan Zhang, Jingwen He, Wei-Shi Zheng, et al. Vbench-2.0: Advancing video generation benchmark suite for intrinsic faithfulness. *arXiv preprint arXiv:2503.21755*, 2025.

[26] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294, 2022.

[27] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

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

[29] Black Forest Labs. FLUX.2: Frontier Visual Intelligence. `https://bfl.ai/blog/flux-2`, 2025.

[30] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report, 2025.

[31] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

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

[33] Yinan Chen, Jiangning Zhang, Teng Hu, Yuxiang Zeng, Zhucun Xue, Qingdong He, Chengjie Wang, Yong Liu, Xiaobin Hu, and Shuicheng Yan. Ivebench: Modern benchmark suite for instruction-guided video editing assessment. *arXiv preprint arXiv:2510.11647*, 2025.

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

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

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

[37] Qingyan Bai, Qiuyu Wang, Hao Ouyang, Yue Yu, Hanlin Wang, Wen Wang, Ka Leong Cheng, Shuailei Ma, Yanhong Zeng, Zichen Liu, Yinghao Xu, Yujun Shen, and Qifeng Chen. Scaling instruction-based video editing with a high-quality synthetic dataset. *arXiv preprint arXiv:2510.15742*, 2025.

[38] Teng Hu, Zhentao Yu, Zhengguang Zhou, Sen Liang, Yuan Zhou, Qin Lin, and Qinglin Lu. Hunyuancustom: A multimodal-driven architecture for customized video generation. *arXiv preprint arXiv:2505.04512*, 2025.

[39] Zhendong Wang, Yifan Jiang, Yadong Lu, Pengcheng He, Weizhu Chen, Zhangyang Wang, Mingyuan Zhou, et al. In-context learning unlocked for diffusion models. *Advances in Neural Information Processing Systems*, 36:8542–8562, 2023.

[40] 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.

[41] Lan Chen, Qi Mao, Yuchao Gu, and Mike Zheng Shou. Edit transfer: Learning image editing via vision in-context relations. *arXiv preprint arXiv:2503.13327*, 2025.

[42] Teng Hu, Jiangning Zhang, Hongrui Huang, Ran Yi, Zihan Su, Jieyu Weng, Zhucun Xue, Lizhuang Ma, Ming-Hsuan Yang, and Dacheng Tao. Evolution of video generative foundations. *arXiv preprint arXiv:2604.06339*, 2026.

[43] Guojun Lei, Rong Zhang, Chi Wang, Tianhang Liu, Hong Li, Zhiyuan Ma, and Weiwei Xu. Unitransfer: Video concept transfer via progressive spatial and timestep decomposition. *arXiv preprint arXiv:2509.21086*, 2025.

[44] Yuheng Chen, Qingdong He, Teng Hu, Yuji Wang, Yabiao Wang, Lizhuang Ma, and Jiangning Zhang. Omni-customizer: End-to-end multimodal customization for joint audio-video generation. *arXiv preprint arXiv:2605.17488*, 2026.

[45] Yuheng Chen, Teng Hu, Yuji Wang, Qingdong He, Zhucun Xue, Qianyu Zhou, Jason Li, Lizhuang Ma, Jiangning Zhang, and Dacheng Tao. Cinedance: Towards next-generation multi-shot long-form cinematic audio-video generation. *arXiv preprint arXiv:2606.09639*, 2026.

[46] Paul Couairon, Clément Rambour, Jean-Emmanuel Haugeard, and Nicolas Thome. Videdit: Zero-shot and spatially aware text-driven video editing. *Transactions on Machine Learning Research*, 2023.

[47] Yuji Wang, Yuheng Chen, Teng Hu, Ran Yi, Yijia Hong, Han Feng, Weijian Cao, Chengjie Wang, Lizhuang Ma, and Jiangning Zhang. Personashot: Benchmarking person-centric narrative continuity in multi-shot video generation. *arXiv preprint arXiv:2608.16717*, 2026.

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

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

[50] Zhen Li, Zuo-Liang Zhu, Ling-Hao Han, Qibin Hou, Chun-Le Guo, and Ming-Ming Cheng. Amt: All-pairs multi-field transforms for efficient frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9801–9810, 2023.

[51] Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. Musiq: Multi-scale image quality transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5148–5157, 2021.

Appendix Overview

This appendix provides additional details and results supporting the main paper.

• Appendix A reviews related work on video editing, visual in-context learning, and datasets.

• Appendix B describes the task taxonomy, data curation pipeline, dataset composition, and prompt templates.

• Appendix C presents the composition and examples of VicEdit-Bench.

• Appendix D documents training configurations, baseline input settings, and evaluation protocols, including human evaluation and Reference Adherence.

• Appendix E reports additional quantitative results, controlled analyses, computational efficiency, and qualitative comparisons.

A Detailed Related Work

A.1 Instruction-driven Video Editing

Training-free methods adapt pre-trained image diffusion models for video editing without task-specific training. Tune-A-Video [12] fine-tunes a text-to-video backbone on a single video for personalized editing. TokenFlow [13] enforces temporal coherence via attention-space nearest-neighbor field matching, while AnyV2V [14] provides a plug-and-play tuning-free framework guided by the first edited frame.

End-to-end trained systems learn editing policies directly from paired datasets to robustly handle complex instructions. EasyV2V [2] highlights training data quality as the primary performance driver across the design space. NOVA [3] decouples editing into sparse control prediction and dense synthesis. VideoCoF [4] adopts chain-of-thought observe-reason-edit stages for temporal consistency, and ReCo [11] incorporates bounding-box constraints for localized modifications. However, these methods rely exclusively on text instructions, lacking visual reference integration.

Visual reference integration has recently emerged to bridge this gap. Kiwi-Edit [10] introduces a single reference image via fixed learnable queries, which lack content adaptivity. While reference-conditioned or customized video generation architectures [38, 15] utilize visual exemplars, they primarily focus on synthesis from noise rather than the direct manipulation of an existing source video. EditVerse [16] tokenizes heterogeneous tasks through interleaved sequence self-attention, but treats distinct modalities homogeneously. In contrast, VicEdit uniquely supports heterogeneous reference inputs (single images, image pairs, and video pairs) via adaptive semantic extraction.

A.2 In-Context Learning for Visual Generation

In-context learning (ICL) originated in large language models [17] and has expanded to visual synthesis, steering generation via contextual demonstrations without gradient updates.

Image generation. Painter [18] and PromptDiffusion [39] formulate visual exemplars as unified sequence prompting. Paint by Example [40], ImageBrush [19], and Edit Transfer [41] further adapt ICL to exemplar-guided image editing using source-target demonstration pairs.

Video generation. Extending ICL to videos involves severe temporal consistency and computational challenges [42]. Early efforts explored prompt-level demonstrations [20] or autoregressive zero-shot clip imitation (VICL [21]). Recent works branch into several technical paradigms: EditVerse [16] uses interleaved full self-attention across modalities; TIC-FT [5] concatenates frames along the temporal axis with learned buffer frames for few-shot adaptation; OmniTransfer [15] and UniTransfer [43] decompose reference and target spaces for concept transfer. Related advances also explore customized and long-sequence synthesis across modalities [44, 45].

Despite these advancements, existing video ICL methods primarily focus on conditional generation from scratch under homogeneous conditions. VicEdit instead addresses heterogeneous multimodal reference integration in an instruction-driven video editing setting, combining Modality-Adaptive Semantic Distillation (MASD) with Dual-Context Injection (DCI).

A.3 Video Editing Datasets

Video editing benchmarks have evolved alongside model paradigms. Early datasets primarily target text-only instruction editing: VidEdit [46] provides web-crawled pairs with spatial masks, and OpenVE [1] introduces diverse operations with preference alignments, alongside benchmarks assessing consistency in long-form generation [47]. For reference-augmented setups, ReCo [11] annotates spatial boxes for region editing, while Kiwi-Edit [10] augments training with single reference images.

Unlike previous benchmarks constrained by text-only directives or single-modality exemplars, VicEdit-400K spans ten editing task categories across heterogeneous single-image, image-pair, and video-pair references, totaling approximately 400K raw curated instances before consolidation.

B More Details on Data Curation

B.1 Task Taxonomy and Definitions

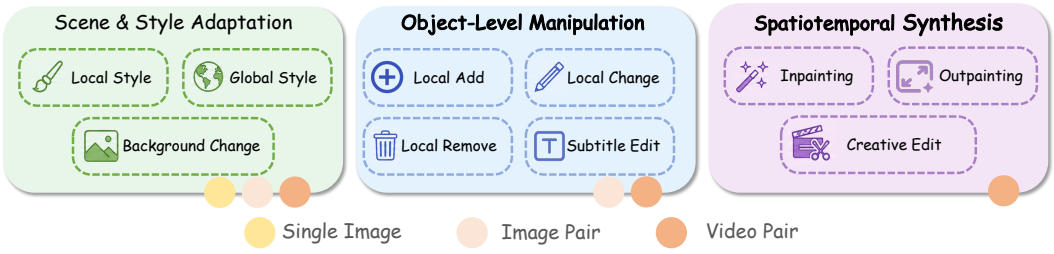

Figure 7: Taxonomy of Editing Tasks and Modalities.

We categorize VicEdit-400K into three task groups aligned with increasing editing complexity along visual, spatial, and temporal dimensions (see Fig. 7). This taxonomy determines which reference modalities are supported for each task, ensuring that the model receives the minimal sufficient visual context for the target editing granularity.

Scene & Style Adaptation. This category focuses on global or regional visual transformation, including tasks such as *Global Style*, *Local Style*, and *Background Change*. All three reference modalities are supported. **Single Image** provides sufficient color, texture, and material priors to serve as an effective anchor for style transfer. **Image Pair** and **Video Pair** further clarify transformation boundaries via explicit before/after comparisons, which is particularly beneficial for background replacement or complex lighting transfer where global consistency must be preserved.

Object-Level Manipulation. This category targets localized object addition, removal, or attribute modification, covering tasks such as *Local Add*, *Local Change*, *Local Remove*, and *Subtitle Edit*. Only Image Pair and Video Pair are supported. Accurate spatial grounding requires source-target correspondence, which Single Image cannot provide because it lacks the "before" context to disambiguate between background preservation and object introduction. **Image Pair** explicitly defines the spatial displacement and local deformation through source-target mapping. **Video Pair** further supplies motion priors (e.g., deformation during object motion), which are critical for handling occlusion and temporal interaction in dynamic scenes.

Spatiotemporal Synthesis. This category covers structurally complex tasks including *Inpainting*, *Outpainting*, and *Creative Edit*. Only **Video Pair** is supported. These tasks elevate editing from pixel-level modification to spatiotemporal logic reconstruction. Static references (Single Image or Image Pair) cannot convey action continuity or object evolution. Video Pair provides dense temporal semantic correspondences that constrain the generation process, enabling the model to learn temporal regularities from reference videos and produce physically plausible completion and scene extrapolation.

Table 8 summarizes the supported modalities for each task type.

Table 8: Supported reference modalities for each task type in VicEdit-400K. SI = Single Image; IP = Image Pair; VP = Video Pair.

Category	Task Type	SI	IP	VP
Scene & Style Adaptation	Global Style	✓	✓	✓
	Local Style	✓	✓	✓
	Background Change	✓	✓	✓
Object-Level Manipulation	Local Add	–	✓	✓
	Local Change	–	✓	✓
	Local Remove	–	✓	✓
	Subtitle Edit	–	✓	✓
Spatiotemporal Synthesis	Inpainting	–	–	✓
	Outpainting	–	–	✓
	Creative Edit	–	–	✓

B.2 Dataset Comparison

Table 9 compares VicEdit-400K with existing video editing datasets. Prior datasets are either text-only or support only a single visual reference modality, limiting their coverage of diverse editing scenarios. In contrast, VicEdit-400K is the first large-scale dataset that supports three reference modalities (Single Image, Image Pair, and Video Pair) across 10 distinct editing task types, providing a systematic foundation for visual in-context video editing research.

B.3 More Details on Data Pipeline

The construction of VicEdit-400K follows a rigorous four-stage pipeline designed to ensure both large-scale diversity and fine-grained semantic alignment.

Source Data Filtering. We begin by curating a high-quality pool of approximately 1.0M video segments, applying an aesthetic cut-off of 0.5 to filter out low-quality raw footage. Given the scale of the source data, we employ the **Qwen3-VL-8B** model as an efficient initial rater to assess semantic consistency (see B.4.1). The filtered corpus is then split: a primary subset serves as the foundation for base editing tasks, while the remainder is reserved for synthesizing heterogeneous in-context exemplars.

Instruction Synthesis. To derive precise editing instructions, we utilize the more powerful **Qwen3-VL-32B** model. As detailed in B.4.2 and B.4.3, our synthesis strategy prioritizes two core objectives: (i) *Semantic Adherence*, ensuring reference content strictly aligns with the editing intent, and (ii) *Visual Divergence*, enforcing a clear distinction between references and source videos to robustly evaluate the model's structural generalization rather than simple memorization.

Table 9: Comparison of video editing datasets. VicEdit-400K provides the most comprehensive support for multi-modal visual in-context references, including Single Image (SI), Image Pair (IP), and Video Pair (VP).

Dataset	Scale	Reference Modality				Task Types	Frames	Resolution
		Text	SI	IP	VP			
InsViE [48]	1.0M	✓	–	–	–	-	25	1024×576
Ditto [37]	1.0M	✓	–	–	–	3	101	1280×720
Señorita [23]	2.0M	✓	–	–	–	9	33–64	336×592–1120×1984
OpenVE [1]	3.0M	✓	–	–	–	8	65–129	1280×720
ReCo [11]	40K	✓	–	–	–	5	81	480×832
RefVIE [10]	477K	✓	✓	–	–	2	65–129	1280×720
VicEdit-400K (Ours)	**400K**	✓	✓	✓	✓	**10**	**33–129**	**480×832–1120×1984**

Visual Reference Generation. For *Single Image* and *Image Pair* modalities, we feed the synthesized prompts into **FLUX.2** [29], which achieves high-fidelity generation in approximately 4s per GPU, facilitating large-scale production. For *Video Pair* synthesis, we employ a hybrid matching strategy. For common tasks, we use *Identity-based Grouping* followed by intra-group random sampling. For specialized tasks like *Creative Edit*, we perform semantic clustering using **BGE-large** embeddings and **HDBSCAN**. This allows us to construct a **Semantic Knowledge Graph** from which relevant in-context pairs are strategically sampled.

Semantic Verification. In the final stage, we perform a multi-dimensional audit covering *Instruction Compliance*, *Detail Fidelity*, and *Visual Quality*. To ensure maximum judgment reliability for the final dataset, we upgrade the evaluator to **Qwen3-VL-32B**, retaining only samples that exhibit superior average scores across all dimensions. This closed-loop verification guarantees that every triplet in VicEdit-400K provides a meaningful and accurate signal for in-context learning.

B.4 Prompt Template in Data Pipeline

To support diverse in-context editing granularities, our data construction pipeline employs a structured, modality-aware prompting strategy using Qwen3-VL. We design distinct templates tailored to the specific requirements of heterogeneous in-context modalities, with a particular emphasis on ensuring that synthesized references remain *thematically consistent yet visually distinct* from the source video content to enhance model generalization.

B.4.1 VLM-as-a-Judge: Multi-Dimensional Quality Assessment

To ensure the high-quality synthesis of VicEdit-400K, we employ a VLM-as-a-Judge mechanism. Unlike simple heuristic-based filtering, our approach leverages the advanced reasoning capabilities of Qwen3-VL to perform a holistic evaluation of the triplet: (Source Video, Edited Video, Instruction). This stage acts as a critical quality gate in our data curation pipeline, filtering out low-fidelity generations and ensuring precise instruction-image-video alignment.

The assessment logic is grounded in three orthogonal dimensions: Instruction Compliance (semantic accuracy), Consistency & Detail Fidelity (spatial-temporal identity preservation), and Visual Quality & Stability (generative aesthetics). We specifically enforce a "bottleneck constraint" where structural or aesthetic scores cannot exceed the compliance score, preventing the inclusion of visually pleasing but semantically irrelevant samples.

Video Score Template

You are a data rater specializing in grading video editing quality. You will be given two videos (before and after editing) and the editing instruction. Your task is to evaluate the video edit on a 5-point scale from three perspectives:
Instruction Compliance
1. No edit or completely unrelated change.
2. Partial or wrong edit; instruction largely unfollowed.
3. Instruction mostly followed but with significant errors or omissions.

4. Correct edit with only minor inaccuracies.
5. Perfect edit that fully matches the instruction.
Consistency & Detail Fidelity
1. Major content lost, distorted, or hallucinated; original scene barely recognisable.
2. Main subject recognisable but key details wrong/missing.
3. Overall structure correct; some local artifacts, warping, or minor omissions.
4. Nearly all geometry and motion intact; only slight, non-distracting deformation.
5. All objects, spatial relations, and motion perfectly preserved outside the edit region.
Visual Quality & Stability
1. Severe flickering, artifacts, or temporal inconsistency making the video unwatchable.
2. Significant and distracting quality degradation or temporal instability.
3. Noticeable but tolerable flaws or flickering.
4. Largely stable and high quality with only minor, subtle issues.
5. Perfectly stable and temporally coherent; indistinguishable from real footage.

Note: The scores for Consistency & Detail Fidelity and Visual Quality & Stability should not be higher than the Instruction Compliance score.
Example Response Format Brief reasoning: A short explanation of the scores, no more than 30 words. Instruction Compliance: A number from 1 to 5. Consistency & Detail Fidelity: A number from 1 to 5. Visual Quality & Stability: A number from 1 to 5. Editing instruction is: {edit_prompt}.
Below are the videos before and after editing:

B.4.2 Cross-Scenario Reference Synthesis for Single-Image Modality

This part of the pipeline focuses on generating a standalone **Visual Exemplar** (R) to support the single-image reference modality. To prevent the model from overfitting to specific instances, we adopt a **"Category-Consistent but Content-Divergent"** strategy. The MLLM is strictly instructed to avoid reproducing the exact scene from the original video, instead generating objects or environments that share the same underlying concept but differ in specific visual instantiation.

Template for Single-Image Reference Synthesis

Core Constraint: The output must describe a *different* but *same-type* instance compared to the instruction.

- **Object-Level Manipulation:** Generate a detailed prompt for a DIFFERENT object of the same category (e.g., if the instruction adds a "brown fedora," describe a "red beret"). Focus strictly on the object's material, shape, and texture while ensuring no background context is included.
- **Style Adaptation:** Use a "Style-on-Random-Scene" approach. Pick a RANDOM everyday scene unrelated to the original video (e.g., "a cat on a windowsill") and render it in the target style to isolate stylistic characteristics from content.
- **Scene Adaptation:** Describe a THEMATICALLY SIMILAR but visually distinct background (e.g., a "retro cyberpunk alley" to represent a "neon futuristic city" theme), ensuring all foreground subjects are removed.

B.4.3 Visual Task Distillation for Image-Pair Modality

For the image-pair modality, the objective is to transform dynamic temporal editing logic into static, paired visual representations. By processing the original frame, the edited frame, and the instruction simultaneously, the MLLM distills the dynamic transition into a dense semantic mapping that captures the "before-and-after" logic.

Template for Image-Pair Task Distillation

System Prompt:
"You are an expert at converting video editing tasks into equivalent image editing tasks. Analyze the visual difference between the ORIGINAL and EDITED frames, understand the editing logic, and generate a representative image editing task."

User Inputs:

- *Visual Anchor:* ORIGINAL video frame (pre-edit).
- *Target Anchor:* EDITED video frame (post-edit).
- *Editing Logic:* Original video instruction: "P".

Synthesized Outputs:

- **IMAGE_PROMPT**: A detailed description of the source scene tailored for high-fidelity T2I generation.
- **EDIT_INSTRUCTION**: A clear, distilled instruction describing the spatial or stylistic transformation required to bridge the two frames.

Through these targeted prompting strategies, VicEdit-400K provides a systematic foundation for training models to interpret and execute edits across diverse visual contexts without relying on simple memorization.

C Comprehensive Specifications of VicEdit-Bench

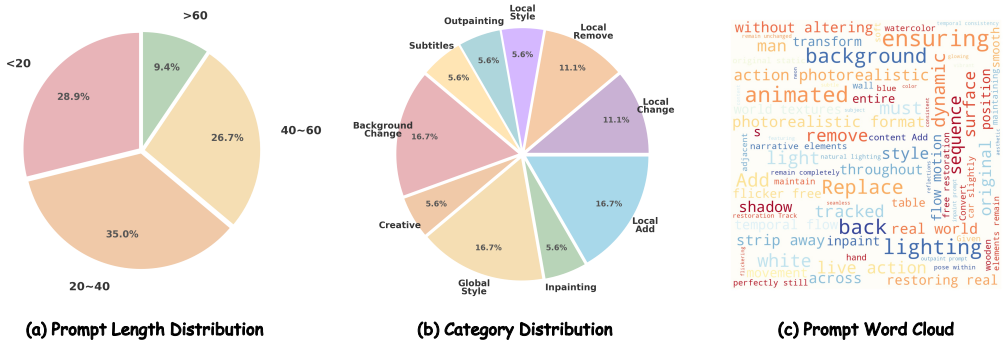

Figure 8: Statistics of VicEdit-Bench.

As illustrated in Figure 8, the design of VicEdit-Bench is strategically aligned with the requirements of visual in-context editing. The category distribution (Fig. 8b) spans a diverse spectrum from global stylistic adaptations to fine-grained local manipulations, necessitating the synergistic use of both textual instructions and visual exemplars. Furthermore, the prevalence of detailed, long-context prompts (Fig. 8a) and the rich semantic vocabulary (Fig. 8c) ensure a rigorous evaluation of the model's ability to resolve linguistic ambiguity through visual anchoring. Overall, VicEdit-Bench provides a comprehensive and challenging testbed that effectively validates the efficacy of our visual in-context editing paradigm.

Fig. 9 showcases the diverse range of video editing tasks covered by VicEdit-Bench.

D Implementation and Evaluation Details

D.1 Detailed Training Configurations

Following the three-stage training strategy described in Section 3.3, we progressively incorporate in-context editing capability into VicEdit. The visual-reference and instruction branches both use



Figure 9: Examples of VicEdit-Bench.

pretrained Qwen weights as their base; the language components are adapted with LoRA in Stages 2 and 3, while Stage 1 keeps the MLLM frozen.

Stage 1: Base Editing. We fully fine-tune the Video DiT on text-instructed editing tasks to establish fundamental instruction-following ability. The model is conditioned on the source video and text instruction, with the target video providing training supervision; no visual reference is used in this stage. Only the DiT parameters are updated, while the MLLM remains frozen and no MLLM LoRA adapters are trained. We train for 2 epochs with a learning rate of 1×10^{-5}. This run uses the 70,000-record instruction subset, comprising 30,000 OpenVE, 30,000 Señorita, and 10,000 OpenVE-derived Creative Edit examples. It is distinct from the balanced 200,000-record curation inventory. Each training sample contains 81 frames.

Stage 2: In-Context Branch. We freeze the Video DiT and train the newly introduced context modules, including MASD and DCI. The ICL manifest contains 200,070 records: 60,000 SI, 100,000 IP, and 40,070 VP examples. We also adapt the language component of the MLLM using rank-64 LoRA adapters, while keeping its underlying pretrained weights and vision encoder frozen. Each training batch mixes samples from the Single Image, Image Pair, and Video Pair modalities to train the modality-specific embeddings jointly. We train the context modules and MLLM LoRA adapters for 4 epochs with a learning rate of 1×10^{-5}.

Stage 3: Joint Fine-Tuning. We jointly optimize the context modules, rank-64 LoRA adapters in the MLLM language component, and rank-32 LoRA adapters in the Video DiT. The underlying backbone weights of both models and the MLLM vision encoder remain frozen during this stage. We train for 4 epochs with a learning rate of 1×10^{-5}.

All stages use 16 GPUs, DeepSpeed ZeRO-3, and BF16 mixed precision, with a reported total training cost of approximately 32 GPU-days.

D.2 Baselines

We compare VicEdit against six state-of-the-art video editing methods. For fair comparison, all methods are evaluated at their recommended resolution and frame length following their official configurations.

VACE [35] is a unified video creation and editing framework based on a 14B Wan2.1 DiT backbone. It formulates diverse editing tasks as sequence completion problems via a task-agnostic Video Condition Unit (VCU), supporting both text-only and reference-guided editing at 720p resolution. However,

its single cross-attention mechanism lacks modality-adaptive semantic extraction for heterogeneous references.

Lucy-Edit [36] is an open-weight text-guided video editing model built on the Wan2.2-5B backbone. It achieves strong instruction-following fidelity via attention-based spatial control, but supports only text-only instructions without visual reference conditioning.

DITTO [37] builds upon a VACE-based diffusion transformer (Wan2.2) fine-tuned on the large-scale Ditto-1M dataset ($\sim$1M paired triplets). Training uses AdamW at 1×10^{-4} for 16K steps on 64 GPUs, processing 101-frame videos at 1280$\times$720 resolution. It operates exclusively in a text-only paradigm.

NovaEdit [3] (CVPR 2026) proposes a sparse-control, dense-synthesis strategy built on the 1.3B Wan2.1 VACE backbone, with cross-attention modules connecting a sparse VACE branch and a dense DiT branch. Training uses only 5K curated video clips at 832$\times$480 resolution, 81 frames, for $\sim$8K AdamW steps. Visual reference guidance is not supported.

VideoCoF [4] formulates video editing as a temporal chain-of-frames reasoning problem on the 14B Wan-T2V backbone. It trains on only 50K video pairs at 33 frames across multi-bucket resolutions (336$\times$592 to 400$\times$944) for 8K steps on 16$\times$H100 GPUs, and generalizes to 500+ frames at inference via RoPE alignment. The method supports text-only instructions.

Kiwi-Edit [10] is the closest baseline, employing a dual-backbone architecture of Qwen2.5-VL-3B as the MLLM encoder and Wan2.2-TI2V-5B as the video DiT. It introduces learnable query tokens (256/512 for instruction, 768 for reference) that extract visual semantics from a single reference image. Training uses a three-stage curriculum (12K total steps at 2×10^{-5} LR) up to 1280$\times$704 at 81 frames. Unlike VicEdit, Kiwi-Edit is limited to single-image reference and does not support image-pair or video-pair modalities.

In contrast, VicEdit supports three reference modalities (Single Image, Image Pair, Video Pair) within a unified framework, and introduces modality-adaptive semantic distillation (MASD) and dual-context injection (DCI) to handle heterogeneous visual references.

Reference-to-Text Instruction Enhancement. For the main VIC comparison, we use Qwen3-VL [28] to enrich the editing instructions supplied to text-only baselines. It receives the original instruction and visual reference, without the source video, and adds editing-relevant visual details while preserving the original editing intent. For paired references, it describes the demonstrated before–after transformation. For illustration, "convert to Gongbi style" can be supplemented with attributes such as even coloring and fine brushwork. Text conversion can lose visual information and is not strictly equivalent to providing the raw reference. The enhancement request is summarized below.

Concise Instruction Enhancement Request

Refine the editing instruction using the visual example. Add concrete, editing-relevant details while preserving the original intent. For paired examples, describe the demonstrated change. Return only the refined instruction.

D.3 Evaluation Details

We employ five core metrics from the VBench [24] suite to assess the generative quality and temporal consistency.

Background Consistency (BC $\uparrow$) quantifies the temporal stability of static environments. It computes the average cosine similarity between consecutive frame-level features extracted by a **CLIP-ViT-B/32** [49] encoder, ensuring the background remains coherent despite camera or subject motion.

Motion Smoothness (MS $\uparrow$) evaluates the physical plausibility of object trajectories. By leveraging motion priors from the **AMT** [50] video frame interpolation model, it detects unnatural jitter or "teleportation" artifacts to ensure fluid pixel-level displacements.

Table 10: Quantitative comparison on VicEdit-Bench across VLM-based assessments and VBench [24] dimensions.

Method	VLM Evaluation				VBench Metrics				
	Overall↑	Instr. Comp.↑	Detail Fid.↑	Visual Qual.↑	BC↑	MS↑	TF↑	IQ↑	AQ↑
VACE [35]	1.85	1.98	1.79	1.80	0.954	0.994	0.983	0.692	0.543
Lucy-Edit [36]	2.25	2.40	2.20	2.15	0.962	0.993	0.983	0.701	0.544
DITTO [37]	2.33	2.44	2.32	2.20	0.966	0.995	0.984	0.703	0.560
NovaEdit [3]	2.61	2.78	2.49	2.54	0.969	0.994	0.985	0.707	0.562
VideoCoF [4]	2.90	3.02	2.85	2.85	0.972	0.995	0.986	0.705	0.561
Kiwi-Edit	2.95	3.10	2.98	2.76	**0.995**	0.986	0.711	0.564	
VicEdit (Ours)	**3.45**	3.65	3.34	3.35	**0.975**	0.994	**0.987**	**0.718**	**0.568**

Temporal Flickering (TF ↑) focuses on high-frequency luminance or texture instabilities. It calculates the pixel-wise standard deviation at identical spatial coordinates across the video duration, penalizing rapid, inconsistent fluctuations often caused by sampling instability.

Imaging Quality (IQ ↑) assesses low-level physical attributes such as sharpness and noise. It utilizes the **MUSIQ** [51] transformer, pre-trained on human-rated data, to objectively measure distortions including motion blur and compression artifacts.

Aesthetic Quality (AQ ↑) measures the artistic appeal and visual harmony (e.g., composition and lighting). Using a predictor aligned with **LAION-Aesthetics** [26], it quantifies how well the generated frames conform to professional photography and cinematic standards.

VLM-Based Editing Scores. For the main evaluation, the VLM judge compares sampled frames from the source video and edited output to assess instruction compliance (IC), detail fidelity (DF), and visual quality (VQ). The visual reference is not provided to the main judge. Overall is the arithmetic mean of IC, DF, and VQ. A separate reference-aware evaluation additionally supplies the intended visual reference and identifies its modality and role in specifying the edit; it reports Reference Adherence (RA) separately from Overall.

Human Evaluation Protocol. We report two pairwise preference studies, involving ten annotators who each evaluate 20 examples per setting: base editing on OpenVE-Ext and visual in-context editing on VicEdit-Bench. For each selected task, annotators compare VicEdit outputs with those of the strongest baseline for that task. The selected examples cover all task categories; the VIC examples are approximately balanced across SI, IP, and VP references. Base editing is assessed on instruction compliance, detail fidelity, and visual quality; VIC editing is assessed on instruction compliance, detail fidelity, and reference adherence.

Annotators choose one of three outcomes: VicEdit wins, VicEdit loses, or a tie. A win gives one point to the preferred method; a tie gives each method 0.5 points. VicEdit's tie-adjusted preference score is

$$S_{\text{human}} = 100 \times \frac{N_{\text{win}} + 0.5N_{\text{tie}}}{N_{\text{win}} + N_{\text{lose}} + N_{\text{tie}}}, \tag{8}$$

where the counts refer to the collected pairwise judgments. Table 3 reports preference scores of 76% and 72% for base editing and VIC editing, respectively. For the same ten-annotator study, human–Gemini agreement is 80% and 76%, and Fleiss' κ is 0.70 and 0.66, respectively. Human–Gemini agreement measures consistency between human preferences and the automated judge; Fleiss' κ measures agreement among annotators on commonly rated items.

E Supplementary Experiments

E.1 More Metrics of Main Experiments

In this section, we supplement more metrics of our main experiments on VicEdit-Bench. Table 10 presents the quantitative comparison, demonstrating that our VicEdit outperforms all baseline

Table 11: Quantitative Comparison on OpenVE-Ext with Qwen-VL-32B [28].

Methods	Overall	Scene & Style Adaptation			Object-Level Manipulation				Spatiotemporal Synthesis		
		Global Style	Local Style	Background Change	Local Change	Local Remove	Local Add	Subtitle Edit	Creative Edit	Inpainting	Outpainting
VACE [35]	2.45	2.11	1.94	2.38	2.87	2.41	2.13	2.24	2.06	2.72	3.09
Lucy-Edit [36]	3.56	3.04	2.82	4.59	4.52	4.44	4.00	2.41	3.16	3.48	3.15
DITTO [37]	3.83	**4.64**	**3.87**	4.66	4.31	4.65	3.99	3.61	1.94	2.92	3.75
NovaEdit [3]	3.23	1.70	2.90	4.50	3.70	4.80	1.95	3.74	2.48	3.25	3.23
VideoCoF [4]	3.41	1.75	3.20	4.55	4.60	4.45	2.90	3.82	2.31	3.15	3.35
Kiwi-Edit	4.14	3.90	3.10	4.80	4.95	4.80	4.45	4.12	3.98	3.75	3.58
VicEdit (Ours)	**4.40**	4.15	3.68	**4.90**	**4.98**	**4.92**	**4.65**	**4.38**	**4.05**	**4.02**	**4.23**

Table 12: Quantitative comparison on VicEdit-Bench with Qwen-VL-32B [28].

Methods	Overall	Scene & Style Adaptation			Object-Level Manipulation				Spatiotemporal Synthesis		
		Global Style	Local Style	Background Change	Local Change	Local Remove	Local Add	Subtitle Edit	Creative Edit	Inpainting	Outpainting
VACE [35]	2.62	2.15	2.44	2.30	3.10	2.45	2.18	2.32	2.85	3.05	3.40
Lucy-Edit [36]	3.15	2.80	2.35	2.92	4.25	3.12	3.30	2.28	3.20	3.55	3.75
DITTO [37]	3.32	3.72	**4.21**	3.35	3.25	2.68	2.35	3.42	2.10	3.30	3.12
NovaEdit [3]	3.48	2.95	3.08	3.84	3.92	4.58	3.12	3.36	2.95	3.48	3.52
VideoCoF [4]	3.81	3.52	3.31	3.75	4.45	4.38	3.61	**4.38**	2.64	3.92	4.10
Kiwi-Edit [10]	3.96	3.98	3.28	3.82	**4.82**	4.25	3.48	3.95	3.51	4.08	3.62
VicEdit (Ours)	**4.36**	**4.25**	3.85	**4.12**	4.75	**4.65**	**4.32**	4.30	**4.28**	**4.31**	**4.42**

methods across various VBench dimensions and VLM-based assessments, achieving state-of-the-art results.

E.2 VLM-based Evaluation Analysis

To verify the robustness of our evaluation protocol, we introduce Qwen-VL-32B as an alternative grader to compare with Gemini 2.5 Pro.

Cross-Model Stability. As shown in Tables 11 and 12, the results from Qwen-VL-32B maintain a high degree of correlation with the Gemini-based scores. While Qwen-VL exhibits a higher scoring baseline (typically $+1.0$ to $+1.5$), the relative performance ranking across all methods remains remarkably consistent. This stability suggests that the VLM-based metrics are model-agnostic and provide a reliable reference for video editing quality.

Quantitative Results. Under this cross-model validation, VicEdit (Ours) consistently achieves the highest overall scores (4.40 and 4.36). Our framework demonstrates a clear advantage in complex tasks such as *Local Remove*, *Creative Edit*, and *Outpainting*. While specialized baselines like DITTO and Kiwi-Edit show competitive results in specific dimensions (e.g., *Local Style* and *Local Change*), VicEdit provides the most balanced performance across all editing categories. These findings confirm the efficacy of our unified in-context learning paradigm in producing high-fidelity and temporally consistent video edits.

E.3 Additional Controlled Analyses from Rebuttal

The following tables collect additional modality-wise, prompt-format, and reference-aware analyses. Protocol distinctions and remaining verification items are described below.

Table 13: Overall scores by reference modality. The SI, IP, and VP subsets contain 30, 70, and 100 examples, respectively, with different task compositions. Comparison with the main Overall score requires the corresponding weights for each task and reference modality.

Reference modality	Kiwi-Edit	VicEdit
Single Image	3.02	3.56
Image Pair	2.92	3.42
Video Pair	2.96	3.30

Table 14: Text/visual input comparison reported during rebuttal. The Stage-1 text-only row uses a Kiwi-Edit-style architecture with our Stage-1 checkpoint, matching the architecture and weights of the original-instruction baseline but using enhanced instructions in a separate experiment. The original-instruction result is reported in Table 7. The full-system result is included for context; the comparison is not established as isolating visual access under matched generation and evaluation protocols.

Setting	Overall	Instr. Comp.	Detail Fid.	Visual Qual.
Kiwi-Edit, enhanced text	2.95	3.10	2.98	2.76
Stage-1 text-only, enhanced text	3.10	3.12	3.08	3.10
VicEdit, full	3.45	3.65	3.34	3.35

Table 15: VideoCoF prompt-format comparison reported during rebuttal. Both variants use reference-derived text and no raw visual exemplar. Verify the actual prompt templates and evaluation records.

Prompt setting	Overall	Instr. Comp.	Detail Fid.	Visual Qual.
Enriched instruction	2.90	3.02	2.85	2.85
Temporal Triptych formatting	3.15	3.25	3.09	3.12

Modality-Wise Results. Table 13 reports higher VicEdit scores on SI, IP, and VP. The subsets contain different task types, so their scores do not isolate modality difficulty or the benefit of changing modality on the same task.

Enhanced Text and Visual Access. Table 14 records the reported text-only and full-model results. A comparison isolates visual access only when source video, instruction text, checkpoint, and generation/evaluation settings are held fixed. The Stage-1 text-only row (Overall 3.10) is a separate experiment using enhanced instructions at generation time. It uses the same Kiwi-Edit-style architecture and our Stage-1 checkpoint as the original-instruction text-only baseline. It must be distinguished from the original-instruction Stage-1 Kiwi-style text-only baseline in Table 7, even though the reported component scores coincide. The table compares these text-only configurations with the reported full-system result; the score difference alone should not be assigned entirely to visual access without matching the generation and evaluation protocols.

Prompt Formatting. Table 15 records VideoCoF's improvement from 2.90 to 3.15 when using Temporal Triptych formatting. This shows that prompt format affects the comparison. It does not establish that all baseline prompt formats have been optimized or that enriched text preserves every visual cue.

Direct Reference Adherence. Table 16 records the reported RA scores of 2.75 for Kiwi-Edit and 3.42 for VicEdit. Both methods are evaluated on the same 200 benchmark samples using their main-evaluation generated videos, with no regeneration. The reference-aware judge receives sampled source and output frames, the enhanced instruction, and the correct reference with its modality identified. The main judge does not receive the visual reference, so these scores are reported separately, as described in Appendix D.3.

E.4 Computational Efficiency

As shown in Table 17, VicEdit achieves inference time comparable to text-only methods with the same backbone. The DiT denoising stage remains unchanged at $\sim$4 min, while the additional MLLM reference feature extraction incurs only 0.5–1.0 s across all three modalities. This overhead represents less than 0.5% of the total inference time, confirming that both the MASD and DCI modules are computationally lightweight and do not compromise inference efficiency.

E.5 Additional Ablation Results

Figure 10 compares component variants on global style transfer and local object replacement. The variants differ in texture, color, and object appearance, while the full model better matches the

Table 16: Reference-aware comparison reported during rebuttal, evaluated on the same complete set of 200 samples from VicEdit-Bench for both methods. The generated videos are reused from the main evaluation without regeneration; the judge inputs and scoring protocol differ. The reference-aware protocol and reconstructed scoring rubric are documented in Appendix D.3.

Method	Instr. Comp.	Detail Fid.	Visual Qual.	Ref. Adherence
Kiwi-Edit	3.02	2.90	2.68	2.75
VicEdit	3.58	3.28	3.28	3.42

Table 17: Quantitative Comparison of Computational Efficiency. All metrics are measured on a single GPU with a target video length of 81 frames (480×832). MLLM Time denotes the reference feature extraction time; DiT Time denotes the denoising inference time.

Method	#Ref. Modality	#Params	MLLM Time	DiT Time
VACE [35]	Text-Only	14B	—	∼35 min
Lucy-Edit [36]	Text-Only	5B	—	∼4 min
DITTO [37]	Text-Only	14B	—	∼35 min
NovaEdit [3]	Text-Only	1.3B	—	∼15 min
VideoCoF [4]	Text-Only	14B	—	∼35 min
Kiwi-Edit [10]	Text-Only	5B	∼0.4s	∼4 min
VicEdit (Ours)	Single Image	5B	∼0.5s	∼4 min
VicEdit (Ours)	Image Pair	5B	∼0.7s	∼4 min
VicEdit (Ours)	Video Pair	5B	∼1.0s	∼4 min

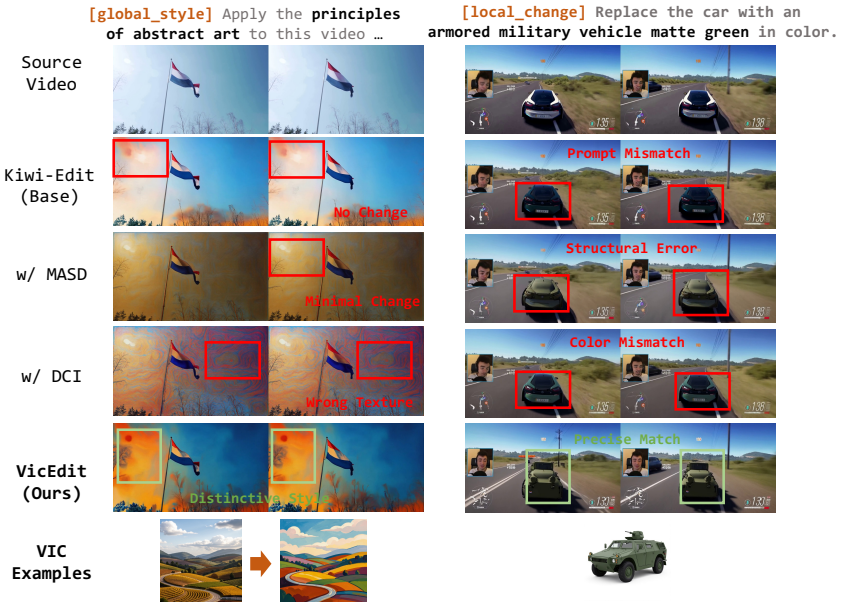

Figure 10: Qualitative ablation of MASD and DCI on (left) a global style editing task and (right) a local object replacement task. The full model better follows the reference style and editing instruction in the shown examples; the comparisons do not imply separately enforced roles for the two modules.

demonstrated edits in these examples. These qualitative observations complement the numerical ablation; they do not establish an exclusive assignment of texture, structure, or instruction-following roles to individual modules.

E.6 More Visualizations

We provide additional visual comparisons with baselines on both base editing and visual in-context editing tasks. As shown in Fig. 11 and Fig. 12, our method consistently outperforms existing approaches across a wider range of examples, further validating the effectiveness of VicEdit.

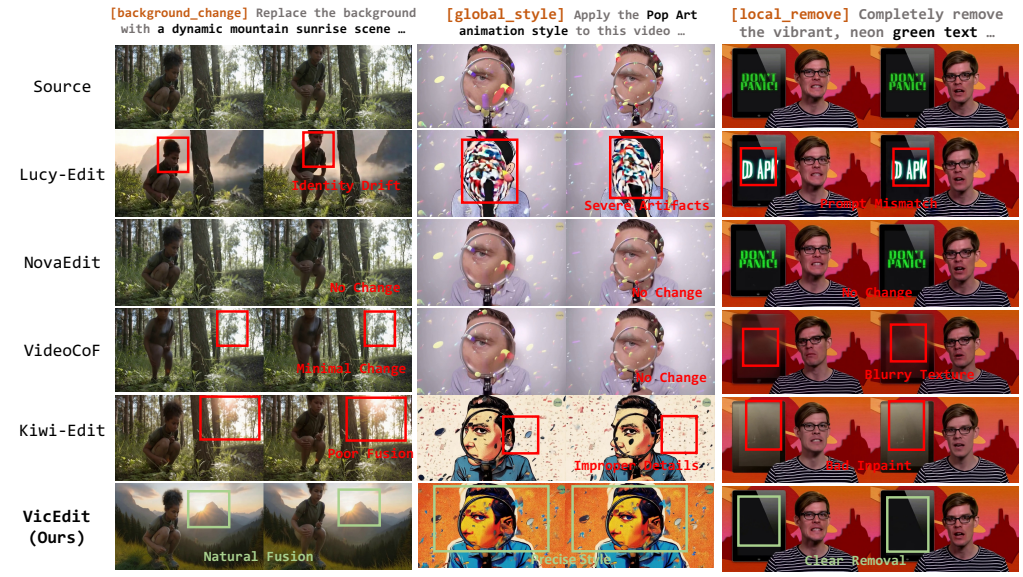

Figure 11: More comparisons with baselines on base editing tasks.

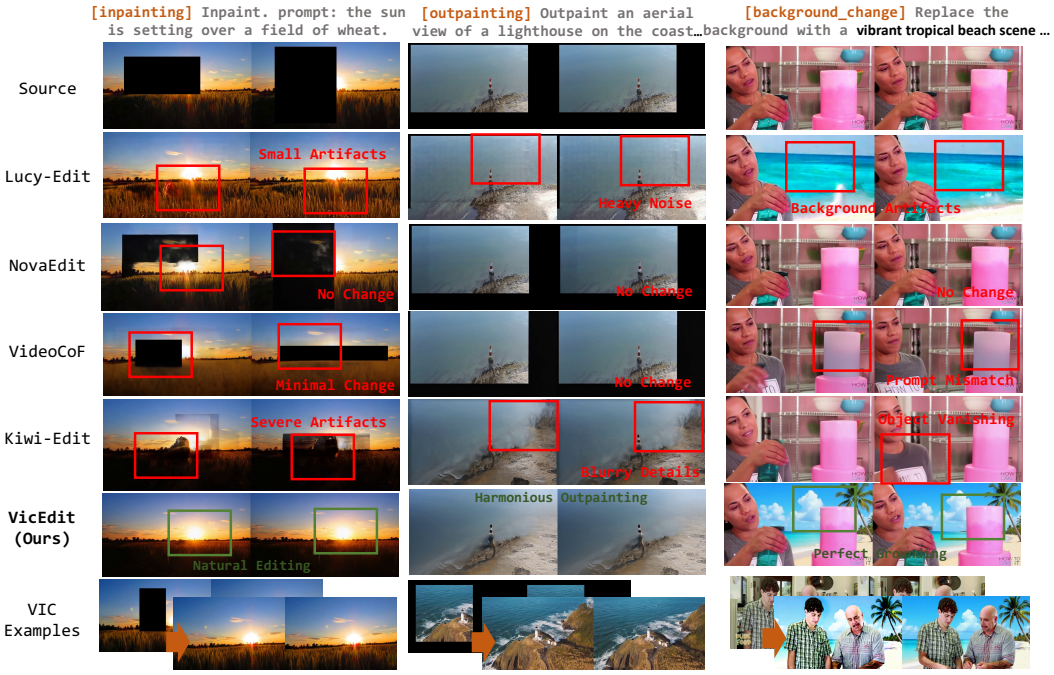

Figure 12: More comparisons with baselines on visual in-context editing tasks.

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 clearly outline our primary contributions, including the proposed framework and the new dataset.

   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: We discuss the limitations and future work in the Conclusion section.

   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 involve intensive theoretical analysis.

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: We have detailed the dataset specifications and training procedures in both the main paper and the appendix.

   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: [No]

Justification: Cleaning the large-scale dataset and codebase for public release is underway. We commit to making all code and data publicly available upon acceptance to ensure reproducibility.

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: We have detailed the dataset specifications and training procedures in both the main paper and the appendix.

   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: [Yes]

   Justification: We report the results of three runs for each experiment.

   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: [Yes]

   Justification: We provide the total computation hours and the number of devices used for training in the Appendix, ensuring reproducibility without disclosing specific infrastructure details.

   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: Our research follows the NeurIPS Code of Ethics.

   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: We discuss the potential positive impacts on video creation and the negative risks.

    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: [Yes]

    Justification: We have implemented semantic filtering during data curation to remove unsafe content and will include usage restrictions in our license.

    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: We properly credit and cite Senorita, OpenVE, and Wan-TI2V. All assets are used according to their respective licenses.

    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: We introduce VicEdit-400K and VicEdit-Bench, providing documentation and maintenance plans in the Appendix.

    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: [No]

    Justification: We conducted a human preference study with ten annotators. Appendix D.3 describes the evaluation protocol and scoring procedure.

    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: [No]

    Justification: The paper includes a human preference study, but does not report participant risk disclosures or the status of institutional ethics review or exemption.

    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: MLLMs (*e.g.*, Qwen-VL) are used as a core component in our data construction pipeline for automated instruction generation and verification.

    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.