EDITEVOLVE: A SELF-EVOLVING AGENT FOR CONTEXT-AWARE VIDEO EDITING

Anonymous authors
Paper under double-blind review

ABSTRACT

Modern video editing models have acquired increasingly powerful content manipulation capabilities, yet high-quality video editing is not simply an instruction-to-video mapping. The same editing instruction can impose different contextual requirements on different source videos: changes to a subject must coordinate its spatial relations, interactions, and associated effects while preserving unrelated content and temporal continuity. The key challenge therefore lies in understanding the source-video context and organizing an appropriate editing strategy accordingly. Identifying an effective strategy, however, requires understanding not only the scene but also how the available tools and generators behave in practice. To this end, we introduce **EditEvolve**, a self-evolving framework that learns context-aware video editing strategies from execution experience without updating model parameters. To accommodate diverse contextual requirements, EditEvolve constructs a composable editing workflow space that enables the agent to flexibly organize editing operations. To learn how to effectively use tools and generators, the framework performs offline workflow exploration and feedback-driven revision, turning failures in instruction following, contextual consistency, and content preservation into signals for strategy improvement. EditEvolve further distills execution trajectories into reusable experience that connects contextual conditions with effective editing strategies, allowing prior execution outcomes to guide workflow composition for unseen tasks. Experimental results show that EditEvolve achieves state-of-the-art performance and consistently improves editing performance across different video editing backbones.

1 INTRODUCTION

Modern video editing models can perform increasingly diverse modifications to subjects, appearance, and visual style from natural-language instructions. However, reliably grounding a language-level editing goal in a specific video remains challenging: text instructions and the model's parametric knowledge may not sufficiently specify the target appearance, intended instance, local geometry, or action–object relationship. We refer to this source- and task-dependent information as the **editing context**. As illustrated by the left MiniMax-H3(MiniMax, 2026) result in Fig. 1, the generator may fail to reproduce an iguana's appearance when replacing a chameleon without a target reference; localize the intended bird when similar birds and their reflections are present; preserve the desired details and shape when replacing a horse with a camel; or clearly bind the action and object changes when "picking up a blue trash bag" is changed to "lifting a red barbell." These cases show that different tasks and source videos require different editing contexts, motivating auxiliary workflows that adaptively provide the conditions needed for reliable editing.

Achieving this gives rise to two core challenges. **First, existing video editing systems lack a unified auxiliary strategy space that can adapt to diverse editing contexts**. Context-aware planning is meaningful only when the system is equipped with corresponding auxiliary capabilities. Different tasks may require target appearance references, instance localization, local appearance examples, or more explicit relationship descriptions. Fixed auxiliary forms or predefined pipelines cannot accommodate such variations. Furthermore, a single task often involves multiple contextual requirements simultaneously, and information dependencies may exist between different auxiliary operations. These operations therefore need to be flexibly composable according to the task. **Second, a reliable mechanism is needed to plan effective workflows and achieve self-evolving through**

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

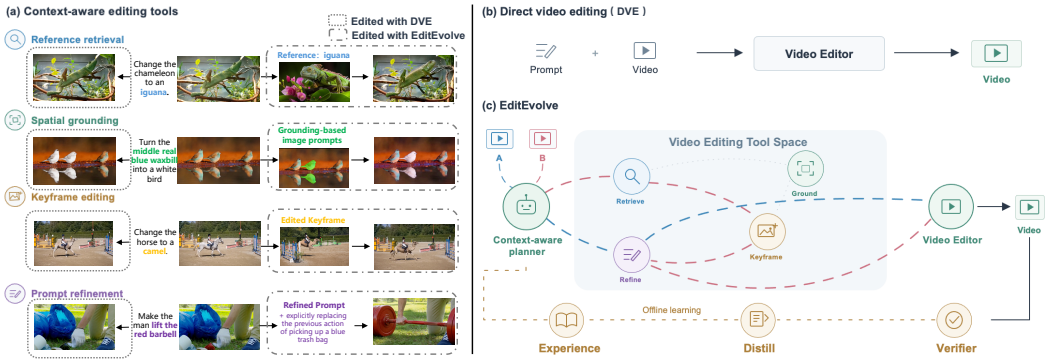

Figure 1: **Overview of EditEvolve.** (a) Reference retrieval, spatial grounding, keyframe editing, and prompt refinement provide complementary guidance for challenging video edits. (b) Direct video editing uses only the source video and input prompt. (c) EditEvolve composes task-specific tool workflows through a context-aware planner and distills offline verification feedback into reusable experience. The colored paths illustrate different workflows selected for different tasks.

execution experience. The same workflow may be effective, redundant, or even harmful to content preservation across different editing contexts, making its utility hard to predict via static rules or model priors. Since historical execution outcomes reveal empirical links among editing contexts, workflow choices, and generation quality, planning strategies should evolve through cross-instance experience to continually calibrate future workflow decisions. Moreover, to remain compatible with closed-source and rapidly evolving video generators, such improvement should ideally not require updating the underlying generator parameters.

To address these challenges, we propose **EditEvolve**,as illustrated in Figure 1(c), a self-evolving framework for context-aware video editing that keeps all model parameters fixed. To provide the task-specific editing context required by different samples, EditEvolve organizes four complementary tools into a composable auxiliary strategy space: Reference retrieval provides evidence of the target appearance; spatial grounding identifies the intended target and edit scope; keyframe editing demonstrates fine-grained changes while preserving the surrounding scene; and prompt refinement clarifies editing constraints and visual bindings. As shown in Figure 1(a) right, these tools can be used independently or composed through their intermediate outputs to construct suitable generation conditions. To account for the context-dependent effectiveness of different workflows, EditEvolve plans each workflow from the source video, editing instruction, and experience retrieved from prior executions. A verifier then compares the source and edited videos for edit correctness, content preservation, and temporal consistency, providing feedback for bounded revision during offline self-evolution. Decision-relevant signals from verified trajectories are distilled into experience cards indexed by context keywords and stored in an external memory. Retrieving these cards for later tasks progressively improves workflow decisions while keeping all model parameters fixed. At test time, the memory is frozen, and each unseen task undergoes a single planning and execution pass without revision or experience updates.

We evaluate EditEvolve on a diverse evaluation set spanning both general video editing and context-dependent scenarios with varying editing requirements. The evaluation considers four complementary aspects of editing quality: Edit Accuracy, Temporal Consistency, Background Preservation, and Aesthetic Quality, with additional analysis of Edit Accuracy across general and context-dependent editing tasks. Experimental results show that EditEvolve effectively handles diverse editing requirements while maintaining strong overall video quality.

Our contributions are threefold:

- We develop a unified, composable workflow space, enabling an agent to organize editing operations according to the source video and instruction.
- We propose **EditEvolve**, which combines context-aware planning with video-based verification and offline experience learning to improve workflow composition without updating model parameters.

- We evaluate EditEvolve across multiple benchmarks and video editing backbones, demonstrating its effectiveness and analyzing the contributions of the tools, adaptive composition, and accumulated execution experience.

2 Related Work

2.1 Instruction-based Video Editing

Instruction-based video editing has progressed through unified generation–editing models, specialized editors, and controllable methods. Unified models such as VACE, VEGGIE, InstructX, UniVideo, VideoCoF, and VIVA (Jiang et al., 2025; Yu et al., 2025; Mou et al., 2025; Wei et al., 2025; Yang et al., 2025; Cong et al., 2025) integrate multimodal instruction understanding with video generation or editing backbones. Other systems, including InsViE-1M, Ditto, and OpenVE (Wu et al., 2025; Bai et al., 2025a; He et al., 2025), improve instruction following and temporal consistency through grounded generation, data construction, or flexible control. QK-Edit, NOVA, EditCtrl, and FlowDirector (Shen et al., 2025; Pan et al., 2026; Litman et al., 2026; Li et al., 2025a) further explore attention manipulation, regional editing, local control, and flow-based guidance.Despite these advances, most existing systems rely on editing policies and control interfaces fixed during training. Their model-specific designs make adaptation to a new video editing backbone costly, often requiring additional training or substantial re-engineering. They also generally lack the ability to adapt editing decisions to different source-video contexts based on previously observed execution outcomes. EditEvolve addresses these limitations with a context-aware and composable workflow that can be transferred across backbones through lightweight interface adaptation and improved across tasks through experience-guided planning, without updating model parameters.

2.2 Agentic and Self-Evolving Visual Generation

Recent studies have begun to explore agentic workflows for visual generation and editing. Preacher applies structured planning and tool coordination to paper-to-video generation, while VISTA, VideoAgent, and SPIRAL improve video generation through critic feedback, environmental interaction, iterative refinement, or long-horizon memory (Liu et al., 2025; Long et al., 2026; Soni et al., 2024; Yang et al., 2026). Agent Workflow Memory and AFlow further investigate reusable workflow induction and execution-driven workflow optimization for general-purpose agents (Wang et al., 2025; Zhang et al., 2025). Most of these methods focus on improving an individual generation process or optimizing workflows in general-purpose settings, rather than learning reusable editing procedures for diverse video contexts. Aurora is more directly related to video editing and uses a tool-using agent to construct model-ready editing conditions (Yu et al., 2026). However, it does not implement an execution-driven, cross-task self-evolution mechanism that distills verified editing trajectories into reusable experience. EditEvolve addresses this gap by combining context-aware planning, comparative video verification, and offline experience distillation without updating model parameters.

3 Method

3.1 Overview

As shown in Fig. 3.1,EditEvolve is a self-evolving workflow for video editing. Given a source video V and an editing instruction I, it generates an edited video $\hat{V}$ that follows the requested edit. Specifically, EditEvolve consists of 3 main parts: (1) to support video editing under diverse context requirements, we construct a composable tool space that provides complementary assistance through reference retrieval, spatial grounding, keyframe editing, and prompt refinement; (2)to adapt tool use to different editing contexts, we introduce a context-aware planning and verification stage, in which the planner selects and composes tools based on the video's and instruction's context, retrieved experience, the verifier diagnoses generation errors and provides feedback for offline self-evolution and (3) to enable self-evolution through accumulated experience without updating model parameters, we introduce offline experience learning, which distills verifier-accepted trajectories into context-indexed experience cards and retrieves relevant cards to guide subsequent planning.

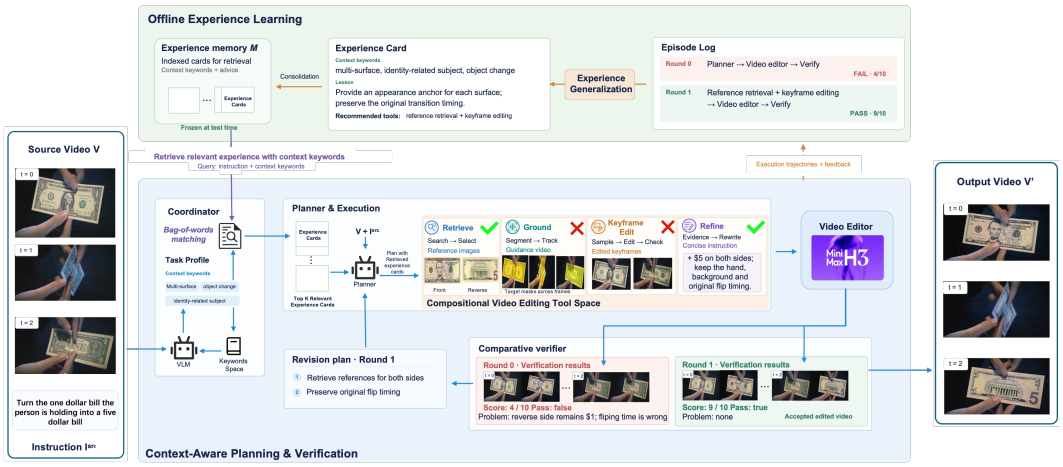

Figure 2: Overview of EditEvolve. (1) A context-aware planner selects and composes optional editing tools. (2) Tool outputs guide a frozen video editor; a comparative verifier diagnoses results and supports bounded revision during offline learning. (3) Execution trajectories are logged, and accepted episodes are distilled into experience cards stored in memory $\mathcal{M}$.

3.2 Composable Video Editing Tool Space

Different editing contexts may require different forms of auxiliary guidance. Since a fixed assistance strategy cannot cover these heterogeneous requirements, we organize four complementary operations into a composable tool space. Each operation can access the source video, the editing instruction, and outputs from preceding operations. These four tools provide modular capabilities for external appearance retrieval, target localization, scene-consistent visual guidance, and textual coordination. All four operations are optional: the planner may invoke one independently or compose multiple according to the task. Specifically:

Reference retrieval. To clarify target appearance or capture distinctive visual details(e.g. five dollar bill in Figure 3.1), the retrieval tool searches for external images based on the editing intent. Relevant references guide subsequent keyframe editing or video generation.

Spatial grounding. To disambiguate the editing target and its spatial extent, the grounding tool localizes instruction-relevant regions through segmentation or tracking. The resulting masks or annotated frames guide subsequent local editing and video generation.

Keyframe editing. To provide desired changes, such as fine-grained appearance and target shape, while preserving the surrounding scene, the keyframe tool edits selected source-video frames, optionally using retrieved references and spatial guidance. The edited frames provide sparse visual examples for video generation.

Prompt refinement. To further coordinate the editing intent with visual evidence and preservation constraints, the refinement tool updates the textual instruction while maintaining the intended target, required constraints, and explicit reference or temporal bindings.

3.3 Context-Aware Planning and Verification

To provide suitable assistance for each source context and editing requirement, EditEvolve combines workflow planning with comparative video verification. The planner selects and organizes optional tools to construct generation guidance, while the verifier assesses editing results and provides feedback for workflow revision and subsequent offline self-evolution.

Task analysis and workflow planning. To identify the auxiliary guidance needed for a task, a dedicated VLM call analyzes the source video V and original instruction I^{src} to construct an

instance-level task profile c. The profile summarizes scene content, distinctive visual characteristics, relevant spatial and semantic relations, editing type, and preservation requirements, and maps them to normalized, sample-agnostic context keywords $K(c)$.

The keyword space $\mathcal{K}$ is initialized offline from recurring editing conditions in the source datasets. For example, `appearance_replace` describes appearance substitution in FiveBench-style tasks, while `subject_with_shadow` describes a scene containing a subject and its cast shadow. When existing keywords are insufficient, the VLM may propose a normalized descriptor without instance-specific names, such as `smoke_color_shift` for recoloring smoke, preferably reusing existing terms. Accepted descriptors are added to $\mathcal{K}$.

To reuse prior executions, the planner queries the experience memory $\mathcal{M}$. Each card is indexed by context keywords and contains tool-use and workflow-revision suggestions. A lightweight bag-of-words matcher ranks cards by their overlap with $K(c)$ and I^{src}, returning the top-k cards as m. The planner then combines V, I^{src}, c, and m to construct a task-specific workflow consisting of an ordered sequence of selected tools and their inputs. Tools are executed according to their information dependencies, and the sequence may be empty when the available information is sufficient. Valid tool outputs and editing requirements are then organized into the generation prompt p and visual guidance, which are provided to the video editor together with V to produce a candidate video $\hat{V}$.

Comparative verification. To assess whether the generated edit follows the requested change while preserving the source content, the source video V and the edited result $\hat{V}$ are concatenated into a comparative visual input. The verifier asks targeted questions about instruction following, contextual consistency, content preservation, and temporal correspondence. It aggregates the answers into an overall score, a pass decision, and a diagnostic description of observed problems. The verification output is recorded in the execution trajectory and used by the offline experience-learning stage.

3.4 Offline Experience Learning

To transfer useful feedback across tasks without updating model parameters, EditEvolve performs offline self-evolution on training tasks. Within each training task, the planner may revise tool calls and generation guidance for a bounded number of rounds based on verification feedback. When an accepted candidate passes verification, its trajectory is summarized into an experience card and added to the experience memory $\mathcal{M}$. These cards guide subsequent tasks and allow experience to accumulate across tasks.

Trajectory logging and experience generalization. For each training task, the system maintains an Episode Log recording the task profile c, the expanded editing instruction, tool calls, intermediate outputs, generation conditions, candidate results, and verification feedback from each round. Full artifacts remain in the log for analysis, while distillation uses the information relevant to tool selection and workflow revision.

After a training task, only a trajectory with an accepted, verification-passing candidate is summarized into an experience card. The card keywords are derived from the task-profile keywords $K(c_i)$ and normalized against the shared keyword space $\mathcal{K}$. Instance-specific names and details are removed, while applicable scene characteristics, editing types, and preservation requirements are retained. Each card stores these keywords together with transferable suggestions for tool use and workflow revision. The resulting cards are added to $\mathcal{M}$. Trajectories that fail verification remain in the Episode Logs for diagnosis but are not stored as successful experience.

4 Benchmark and Evaluation

4.1 Evaluation Data

Our evaluation set comprises 274 editing instances, including 165 samples from FiveBench (Li et al., 2025b) and 109 context-dependent editing samples collected by us. The FiveBench samples cover general editing operations such as subject replacement, attribute modification, addition, and removal. To complement these general editing tasks, our collected samples focus on scenarios involving visually or physically coupled content, such as smoke, flames, shadows, reflections, and human–

object interactions, where modifying the target subject may require coordinated changes to associated regions or relations. Together, these samples provide diverse editing tasks with substantially different contextual requirements.

4.2 Evaluation Metrics

We evaluate generated videos along four complementary dimensions: Edit Accuracy, Temporal Consistency, Background Preservation, and Aesthetic Quality. Edit Accuracy measures whether the requested modification is correctly applied to the target subject. Temporal Consistency evaluates the temporal stability and coherence of the edited content, including instance consistency, flickering, and motion smoothness across frames. Background Preservation measures whether regions irrelevant to the requested edit remain faithful to the source video. Aesthetic Quality assesses the perceptual quality of the edited video. We report each metric separately and aggregate the normalized scores for overall comparison. Detailed evaluation protocols are provided in Appendix A.

5 Experiments

5.1 Implementation Detail

Our framework is implemented as a fully training-free pipeline over a frozen video editor. Qwen3.8-27B is used as the VLM planner to select optional evidence tools under a discrete cost budget, and the same model family drives prompt refinement, official H3 Skill compilation, and side-by-side verification Bai et al. (2025b). When spatial localization is required, SAM3 performs text-prompted segmentation and mask tracking on the source video; rejected or sparse masks are discarded rather than forced into generation For geometry-preserving appearance anchors, we uniformly sample up to 8 frames from the source, select the clearest distinct views, and edit them with GPT Image 2; failed edits are dropped and do not alter the instruction (OpenAI, 2026b). Finally, MiniMax-H3 MiniMax (2026) generates the edited video using the Skill-compiled prompt (produced by GPT-5.6-lunaOpenAI (2026a) for fair comparison) and any accepted reference evidence. For offline experience learning, we construct a separate experience-learning split whose task distribution matches that of the held-out test set. For each task in this split, the planner–verifier loop performs at most three rounds of iterative generation and revision, from which preliminary experience cards are distilled. During testing, each sample is processed only once to avoid the additional cost of repeated video generation; however, experience accumulated from earlier test samples is still used to guide subsequent samples.

5.2 Quantitative Evaluation

Table 1 reports the quantitative comparison on our evaluation set. EditEvolve achieves the best overall score of 0.6615 and the highest Edit Accuracy of 0.6685, outperforming the second-best Omni-Video2 (0.5978). On FiveBench and the context-dependent subset, EditEvolve achieves Edit Accuracy scores of 0.6515 and 0.6944, respectively, ranking first on both. The consistent gains across both subsets suggest that EditEvolve adapts effectively to heterogeneous editing requirements, rather than benefiting only a particular type of editing context.

The remaining dimensions further show that the improvement in edit fulfillment does not come at the cost of video quality. EditEvolve achieves the second-best Temporal Consistency of 0.8941, closely approaching MiniMax-H3 (0.8981), while maintaining competitive Background Preservation (0.5380) and Aesthetic Quality (0.5454). Although individual baselines show strengths in particular quality dimensions, they do not achieve the same Edit Accuracy across both evaluation subsets. Overall, EditEvolve improves edit fulfillment across diverse editing requirements while maintaining strong temporal and perceptual quality.

5.3 Qualitative Evaluation

Figure 3 presents qualitative comparisons across diverse video editing tasks. The examples highlight why different instructions require different forms of editing guidance. For instance, changing only the middle real blue waxbill into a white bird requires distinguishing the intended instance from other birds in the scene. Replacing the one-dollar bill held by the person with a five-dollar bill requires

Table 1: Performance comparison of different video editing methods on our evaluation set. We report Edit Accuracy, Temporal Consistency, Background Preservation, and Aesthetic Quality, together with Edit Accuracy on FiveBench and our context-dependent editing samples.

Method	Dimension Performance					Detailed Performance	
	Total Score ↑	Edit Accuracy ↑	Temporal Consistency ↑	Background Preservation ↑	Aesthetic Quality ↑	FiveBench Edit Acc. ↑	Context-Dep. Edit Acc. ↑
Ditto (Bai et al., 2025a)	0.5199	0.1264	0.8521	**0.5655**	0.5356	0.1000	0.1667
Aurora (Yu et al., 2026)	0.5821	0.4635	0.8519	0.4872	0.5256	0.4333	0.5092
Kiwi-Edit (Lin et al., 2026)	0.5858	0.4615	0.8567	0.4952	0.5296	0.4421	0.4908
Omni-Video2 (Anonymous, 2026)	0.6160	<u>0.5978</u>	0.8543	0.4391	**0.5727**	<u>0.5741</u>	<u>0.6330</u>
MiniMax-H3 (MiniMax, 2026)	<u>0.6170</u>	0.4760	**0.8981**	<u>0.5556</u>	0.5381	0.5576	0.3491
Our	**0.6615**	**0.6685**	<u>0.8941</u>	0.5380	<u>0.5454</u>	**0.6515**	**0.6944**

Table 2: Ablation study of EditEvolve. All variants are evaluated using the same evaluation protocol as the main comparison.

Method	Total Score ↑	Edit Accuracy ↑	Temporal Consistency ↑	Background Preservation ↑	Aesthetic Quality ↑
MiniMax-H3	0.6170	0.4760	**0.8981**	**0.5556**	0.5381
w/o Prompt Refinement	0.6349	0.5680	0.8879	0.5368	0.5469
w/o Keyframe Editing	0.6236	0.5221	0.8888	0.5406	0.5430
w/o Spatial Grounding	0.6227	0.5111	0.8922	0.5470	0.5405
w/o Reference Retrieval	0.6254	0.5239	0.8884	0.5426	0.5467
w/o Memory	0.6254	0.5641	0.8753	0.5243	0.5380
w/o Online Memory	0.6441	0.6113	0.8899	0.5359	0.5391
EditEvolve	**0.6615**	**0.6685**	<u>0.8941</u>	<u>0.5380</u>	0.5454

both precise target localization and appearance guidance across the visible sides of the bill. Similarly, changing a chameleon into an iguana or a dog into a seal clapping its flippers involves substantial changes in object appearance, shape, and action. Other examples require localized effect editing, such as recoloring the mist from a whale's blowhole or the exhaust behind a race car, while preserving the surrounding scene and original motion.

Existing methods often fail to fully realize these edits, modify unintended regions, introduce shape or appearance distortions, or lose the original motion. In contrast, EditEvolve adapts its editing strategy to the task. It localizes the intended subject in ambiguous scenes, provides visual guidance for appearance-specific changes, and coordinates object, appearance, and action modifications. As shown in the figure, it correctly edits the selected bird while leaving the other birds unchanged, changes the held bill while preserving the hand and its motion, and performs local effect recoloring without substantially affecting the surrounding scene. It also handles larger transformations, such as removing a backpack or changing a person into a young boy, while maintaining the source video's motion and scene structure. These examples show how context-aware workflow composition enables one editing agent to adapt its problem-solving strategy to diverse video editing tasks.

5.4 Ablation Study

We conduct two sets of ablations to evaluate the conditional contributions of the composable tools and the experience-based planning mechanism. For the tool-space analysis, we remove one tool at a time while keeping the planner, video editor, remaining tools, and evaluation protocol unchanged. This one-at-a-time design measures the marginal contribution of each tool within the complete workflow, rather than assuming that their effects are independent or additive. As shown in Table 2, removing prompt refinement, keyframe editing, spatial grounding, and reference retrieval decreases the total score by 2.66, 3.79, 3.88, and 3.61 percentage points, respectively. The corresponding edit-accuracy drops are 10.05, 14.64, 15.74, and 14.46 percentage points, with spatial grounding causing the largest degradation. This result suggests that grounding and visual guidance are particularly important for correctly identifying and executing the requested edit, while prompt refinement also provides a substantial contribution. The effects on temporal consistency, background preservation, and aesthetic quality are smaller and non-monotonic: for example, removing spatial grounding

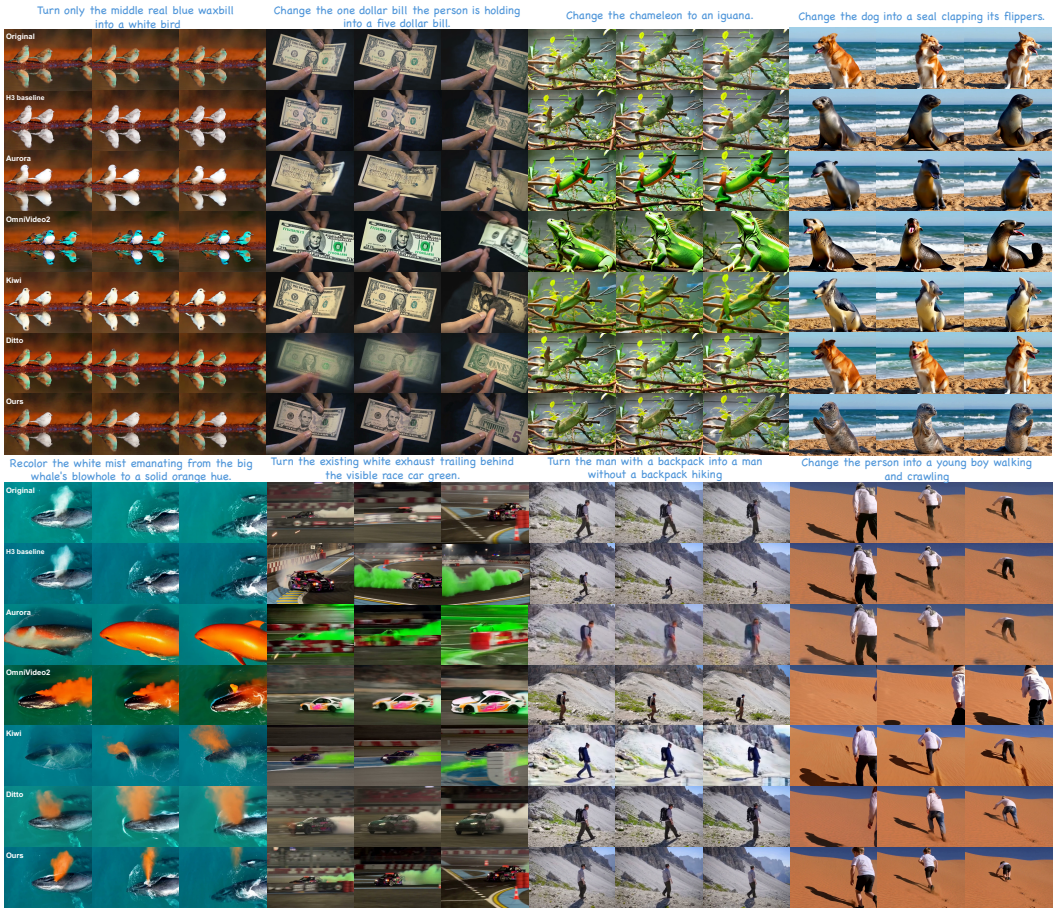

Figure 3: **Qualitative results.**

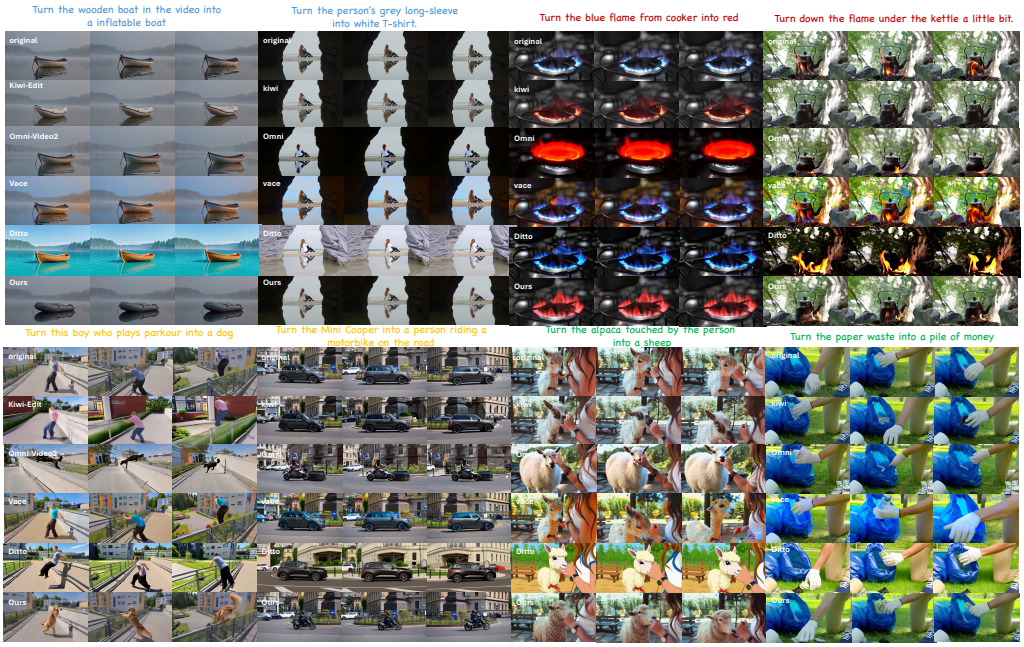

Figure 4: **Qualitative results of VACE enhanced by compositional tool space.**

Table 3: Performance comparison of tool space enhanced VACE and other video editing methods.

Method	Total Score ↑	Edit Accuracy ↑	Temporal Consistency ↑	Background Preservation ↑
VACE (Jiang et al., 2025)	0.5058	0.1759	0.8479	0.4937
Ditto (Bai et al., 2025a)	0.5546	0.3243	0.8200	<u>0.5195</u>
Kiwi-Edit (Lin et al., 2026)	<u>0.6489</u>	0.5937	**0.8858**	0.4672
Omni-Video2 (Anonymous, 2026)	0.6212	<u>0.6481</u>	0.8600	0.3556
Our	**0.7319**	**0.7925**	<u>0.8660</u>	**0.5371**

slightly improves background preservation from 0.5380 to 0.5470, while removing prompt refinement increases aesthetic quality from 0.5454 to 0.5469. Thus, the main benefit of the complete tool space lies in improving edit correctness and the aggregate score, rather than uniformly maximizing every individual metric.

Table 2's bottom 3 rows show three settings: *EditEvolve w/o Memory*, which removes both training-time and test-time experience; *EditEvolve w/o Online Memory*, which retains the offline experience memory but disables experience accumulation during testing; and the full *EditEvolve* model. Removing all experience leads to the lowest performance, with an overall score of 0.6254 and an edit accuracy of 0.5641, indicating that relying solely on the existing VLM itself gives it a certain workflow planning capability, but it does not fully understand the editing context. Retaining the offline memory improves the overall score to 0.6441 and the edit accuracy to 0.6113, demonstrating the value of previously accumulated execution knowledge. The full model further improves the overall score to 0.6615 and achieves the best performance across all reported metrics, especially in edit accuracy, which reaches 0.6685. These results suggest that offline experience provides useful initial workflow priors, while test-time experience accumulation offers additional cross-task improvements. Together, the two forms of experience enable more effective workflow decisions without increasing the per-task generation budget.

5.5 Tool-Space Transfer

To assess the portability of the composable tool space, we apply the same workflow to VACE (Jiang et al., 2025) on our collected context-dependent editing set. As shown in Table 3, the tool-augmented VACE achieves the best total score, edit accuracy, and background preservation, reaching 0.7319, 0.7925, and 0.5371, respectively, while remaining competitive in temporal consistency. The qualitative results in Figure 4 show that the workflow supports diverse appearance and shape replacements, localized effect changes, and interaction-related edits while preserving scene motion and surrounding content. These results demonstrate that the composable tool space provides effective auxiliary guidance for frozen video editors and transfers beyond a single backbone.

6 Conclusion

In this work, we present **EditEvolve**, a training-free, self-evolving agent for context-aware video editing. Rather than relying on fixed pipelines or solving each request from scratch, EditEvolve organizes reference retrieval, spatial grounding, keyframe editing, and prompt refinement into a composable tool space. A context-aware planner selects task-specific workflows according to the source video and editing instruction, while comparative video verification evaluates the generated results and supports bounded revision during offline learning. Verified execution trajectories are distilled into reusable experience cards that used for improving subsequent workflow decisions without updating model parameters. Experiments on general and context-dependent editing tasks show that EditEvolve improves editing quality across different video editing backbones while maintaining temporal consistency and background preservation. Ablation studies further confirm the complementary contributions of the editing tools and accumulated experience. Online analysis additionally demonstrates the potential of verified experience to support continual cross-task workflow improvement under a one-generation-per-task budget.

AI USE STATEMENT

Generative AI tools were used to assist with the interpretation and organization of experimental results and qualitative analysis. All AI-assisted analyses were independently reviewed and verified by the authors. The authors take full responsibility for the final content of this work.

REFERENCES

Anonymous. Omni-Video 2: Scaling MLLM-conditioned diffusion for unified video generation and editing, 2026.

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, 2025a.

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

Xiaoyan Cong, Haotian Yang, Angtian Wang, Yizhi Wang, Yiding Yang, Canyu Zhang, and Chongyang Ma. VIVA: VLM-guided instruction-based video editing with reward optimization, 2025.

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, 2025.

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

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

Guangzhao Li, Yanming Yang, Chenxi Song, and Chi Zhang. FlowDirector: Training-free flow steering for precise text-to-video editing, 2025a.

Minghan Li, Chenxi Xie, Yichen Wu, Lei Zhang, and Mengyu Wang. FiVE: A fine-grained video editing benchmark for evaluating emerging diffusion and rectified flow models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025b.

Yiqi Lin et al. Kiwi-Edit: Versatile video editing via instruction and reference guidance, 2026.

Yehonathan Litman, Shikun Liu, Dario Seyb, Nicholas Milef, Yang Zhou, Carl Marshall, Shubham Tulsiani, and Caleb Leak. EditCtrl: Disentangled local and global control for real-time generative video editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Jingwei Liu, Ling Yang, Hao Luo, Fan Wang, Hongyan Li, and Mengdi Wang. Preacher: Paper-to-video agentic system. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17129–17139, 2025.

Do Xuan Long, Xingchen Wan, Hootan Nakhost, Chen-Yu Lee, Tomas Pfister, and Sercan O. Arik. VISTA: A test-time self-improving video generation agent. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6021–6032, 2026.

MiniMax. MiniMax H3: An open model breaking the boundaries between tasks and modalities. `https://www.minimax.io/blog/minimax-h3`, 2026. Blog post. Accessed: 2026-09-24.

Chong Mou, Qichao Sun, Yanze Wu, Pengze Zhang, Xinghui Li, Fulong Ye, Songtao Zhao, and Qian He. InstructX: Towards unified visual editing with MLLM guidance, 2025.

OpenAI. GPT-5.6 Sol. `https://developers.openai.com/api/docs/models/gpt-5.6-sol`, 2026a. OpenAI API model documentation. Accessed: 2026-09-24.

OpenAI. GPT-Image-2. `https://developers.openai.com/api/docs/models/gpt-image-2`, April 2026b. OpenAI API model documentation. Accessed: 2026-09-24.

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, 2026.

Tiancheng Shen, Zilong Huang, Xiangtai Li, Zhijie Lin, Jiyang Liu, Yitong Wang, Jiashi Feng, Ming-Hsuan Yang, and Jun Hao Liew. QK-Edit: Revisiting attention-based injection in MM-DiT for image and video editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19043–19053, 2025.

Achint Soni, Sreyas Venkataraman, Abhranil Chandra, Sebastian Fischmeister, Percy Liang, Bo Dai, and Sherry Yang. Videoagent: Self-improving video generation, 2024.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 63897–63911, 2025.

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, 2025.

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 (ICCV)*, pp. 16692–16701, 2025.

Zitong Xu, Huiyu Duan, Bingnan Liu, Guangji Ma, Jiarui Wang, Liu Yang, Shiqi Gao, Xiaoyu Wang, Jia Wang, Xiongkuo Min, Guangtao Zhai, and Weisi Lin. LMM4Edit: Benchmarking and evaluating multimodal image editing with LMMs. In *Proceedings of the ACM International Conference on Multimedia*, pp. 6908–6917, 2025.

Xiangpeng Yang, Ji Xie, Yiyuan Yang, Yue Ma, Yan Huang, Min Xu, and Qiang Wu. VideoCoF: Unified video editing with temporal reasoner, 2025.

Yu Yang, Yue Liao, Jianbiao Mei, Baisen Wang, Xuemeng Yang, Licheng Wen, Jiangning Zhang, Xiangtai Li, Liang Lv, Hanlin Chen, Botian Shi, Yong Liu, Shuicheng Yan, and Gim Hee Lee. SPIRAL: Self-evolving action-conditioned video generation via reflective planning agents, 2026.

Shoubin Yu, Difan Liu, Ziqiao Ma, Yicong Hong, Yang Zhou, Hao Tan, Joyce Chai, and Mohit Bansal. VEGGIE: Instructional editing and reasoning video concepts with grounded generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15147–15158, 2025.

Yongsheng Yu, Ziyun Zeng, Zhiyuan Xiao, Zhenghong Zhou, Hang Hua, Wei Xiong, and Jiebo Luo. Aurora: Unified video editing with a tool-using agent, 2026.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xiong-Hui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, Bingnan Zheng, Bang Liu, Yuyu Luo, and Chenglin Wu. AFlow: Automating agentic workflow generation. In *International Conference on Learning Representations*, 2025.

A Detailed Evaluation Protocol

We provide implementation details for the four evaluation dimensions used in our experiments: Edit Accuracy, Temporal Consistency, Background Preservation, and Aesthetic Quality. All metric scores are normalized to $[0, 1]$. The overall score reported in the main paper is computed by equally averaging the four dimensions:

$$S_{\text{overall}} = \frac{1}{4} \left(S_{\text{edit}} + S_{\text{temp}} + S_{\text{bg}} + S_{\text{aes}} \right). \tag{1}$$

A.1 Edit Accuracy

Edit Accuracy measures whether the requested modification is correctly performed in the edited video. We follow the FiVE-Acc protocol from FiveBench (Li et al., 2025b) and formulate edit correctness as a VQA-based verification task. For each sample, binary or multiple-choice questions are constructed according to the editing instruction and source video and manually verified before evaluation.

We use Qwen3.8-Flash as the VQA evaluator. For each edited video, we uniformly sample 8 frames and ask the evaluator to answer the verified questions. Edit Accuracy is computed from the accuracy of these responses. We apply the same evaluation procedure to both the FiveBench samples and our context-dependent samples, and additionally report Edit Accuracy on the two subsets separately.

A.2 Temporal Consistency

Temporal Consistency measures the temporal stability and coherence of the edited video. It consists of three complementary components: strict instance consistency, temporal flickering, and motion smoothness. The final score is computed as

$$S_{\text{temp}} = 0.4 S_{\text{ins}} + 0.35 S_{\text{flicker}} + 0.25 S_{\text{motion}}. \tag{2}$$

Strict instance consistency. We divide each edited video into non-overlapping 32-frame clips and uniformly sample 8 frames from each clip. Qwen3.8-Flash is used to determine whether the edited instance remains consistently preserved within each clip. We adopt a strict video-level criterion: a video receives a score of 1 only when all clips pass the consistency check, and 0 otherwise.

Temporal flickering. Temporal flickering evaluates frame-to-frame appearance instability. Since pixel-wise differences can be dominated by legitimate object or camera motion, we use optical-flow-based warping consistency to compare temporally corresponding content between adjacent frames. Higher scores indicate lower appearance instability after accounting for motion.

Motion smoothness. Motion smoothness evaluates whether motion evolves naturally over time. We use the motion smoothness metric from VBench (Huang et al., 2024) and normalize the resulting score to $[0, 1]$.

A.3 Background Preservation

Background Preservation measures whether content irrelevant to the requested edit remains faithful to the source video. We adopt the attribute preservation metric from LMM4Edit (Xu et al., 2025) and extend it to video editing through frame-wise evaluation. For each sample, we uniformly select 8 temporally aligned frame pairs from the source and edited videos. Each pair is evaluated for the preservation of edit-irrelevant visual attributes, and the video-level score is obtained by averaging the scores across all sampled frame pairs.

A.4 Aesthetic Quality

Aesthetic Quality measures the perceptual quality of the edited video. We follow the `aesthetic_quality` metric in VBench (Huang et al., 2024), which uses the LAION Aesthetic Predictor v1 with an OpenAI CLIP ViT-L/14 backbone and a linear prediction head. For each video, we uniformly sample 16 frames and average their aesthetic scores. Following VBench, the averaged raw score is divided by 10 to obtain the final normalized score.

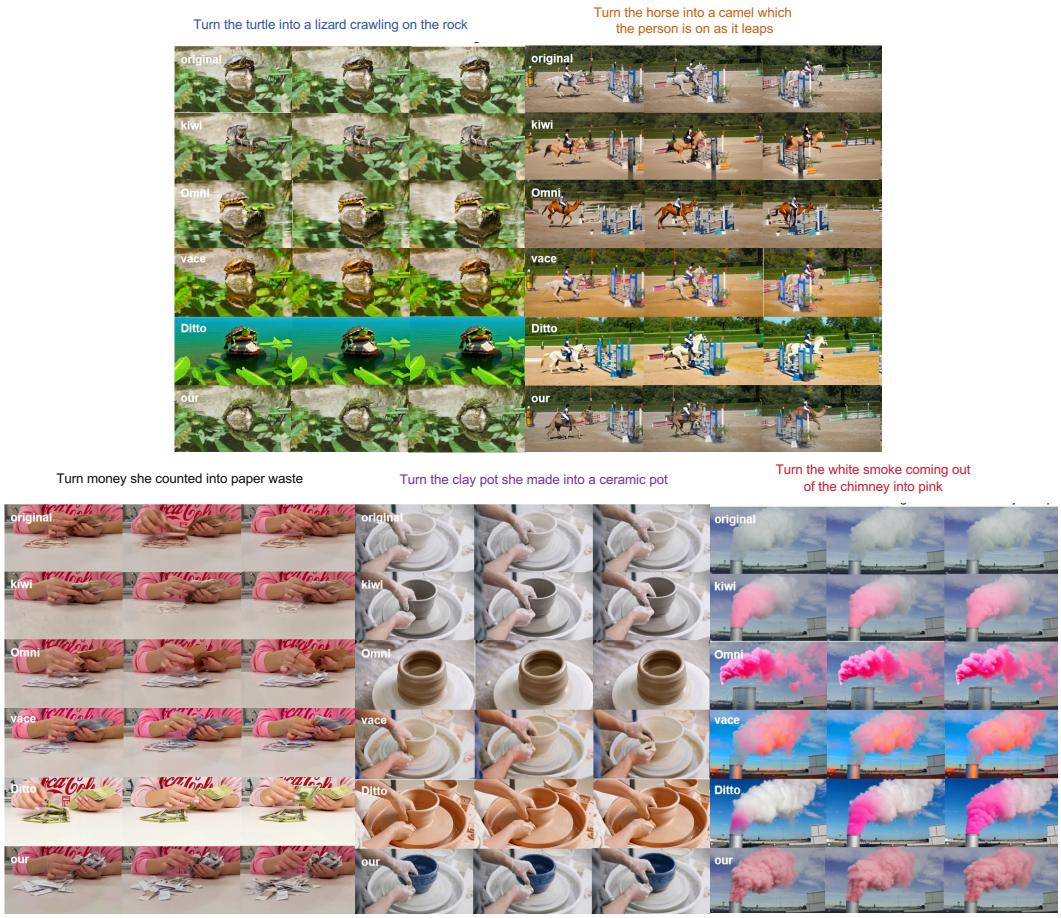

Figure 5: **Additional qualitative comparisons on diverse video editing tasks**. The examples cover both general editing and context-dependent scenarios with diverse editing requirements.

B Additional Qualitative Comparisons

Fig. 5 provides additional qualitative comparisons across diverse video editing tasks. Compared with existing baselines, EditEvolve more reliably fulfills the editing instructions while maintaining temporal coherence and preserving unrelated visual content. These results further demonstrate the effectiveness of adapting the editing workflow to different video contexts and editing requirements.