Unicorn: Controllable Multi-Shot Video Editing Framework via Shot-wise Keyframe Anchors

Anonymous submission

Abstract

Video editing has made rapid progress on videos captured as a single continuous shot. However, real-world videos often consist of multiple shots with abrupt scene transitions and various camera viewpoints. In this setting, existing methods face two key limitations: they often fail to accurately localize the target shot for editing and struggle to preserve consistent object identity across multiple shots. In this paper, we present Unicorn, the first controllable multi-shot video editing framework that addresses these challenges through consistent shot-level keyframe anchors. Concretely, Unicorn extracts a keyframe from each shot, identifies the shots that require editing, and edits their corresponding keyframes with an image editing model. The edited keyframes are then injected into the video editing process as precise shot-level anchors. To maintain consistency across multiple edited shots, Unicorn introduces in-context image editing to propagate the edit from an anchor keyframe to other keyframes, thereby maintaining the consistency of the key subject across edited keyframes from multiple shots. Furthermore, to address the lack of a systematic benchmark in the field, we introduce MultiShotBench, a real-world multi-shot video editing benchmark covering diverse scenarios and edit types with multidimensional evaluation protocols. On MultiShotBench, Unicorn achieves state-of-the-art performance, demonstrating strong shot-level edit accuracy and cross-shot object consistency. We further validate its generalization to single-shot video editing on OpenVEBench and IVEBench, where it achieves state-of-the-art performance among open-source methods.

Introduction

Video editing (Qi et al. 2023; Geyer et al. 2023; Ku et al. 2024; Jiang et al. 2025; Bai et al. 2025a; Wei et al. 2025; Zhang et al. 2025a; Wang et al. 2026) aims to modify visual content in a source video according to user intent while preserving unrelated content. Recent methods have made rapid progress on single-shot videos, where the camera motion is continuous and the target object usually remains visible throughout the video. However, real-world videos often contain multiple shots with abrupt scene transitions, changing viewpoints and different backgrounds. For example, advertisements may switch between product and usage shots; films may alternate between characters in a conversation. Despite its practical importance, the field of multi-shot video editing is still underexplored.



Figure 1: Demonstration of two challenges by applying existing video editing models to multi-shot videos.

Applying existing video editing models to multi-shot videos faces two major challenges, as illustrated in Fig. 1. The first is **inaccurate edit localization**: in the top example, replacing the brown cowboy hat should affect only shots containing it, yet the current method also adds a white cap to irrelevant shots. The second is **cross-shot identity preservation**: in the bottom example, though the existing method changes the gray top to a red sports jacket in each shot, the appearance of the red sports jacket remains inconsistent across shots.

These failures arise because existing video editors implicitly assume that the target object is expected to remain present along the entire timeline. In multi-shot videos, however, shot transitions break this assumption, and the target may be absent from some shots. Without explicit awareness of shot structure, current editors may still apply the editing instruction to every shot, leading to edits in irrelevant shots. Moreover, multi-shot videos often exhibit substantial changes in the target object's scale and viewpoint across shots, making it difficult for current methods to preserve a consistent object appearance across shots.

This is an anonymized submission for review purposes only. Distribution, citation, or public sharing of this manuscript is strictly prohibited. Copyright and publication details will appear in the final version if accepted.

To address these challenges, we present **Unicorn**. To the best of our knowledge, this is the first multi-shot video editing framework. Our key insight is to explicitly model its shot structure with a sequence of keyframes and perform the desired edits on these keyframes. Enabled by current powerful image editing models, such as Qwen-Image-Edit (Wu et al. 2025a) and FLUX.2 (Black Forest Labs 2025), the resulting edited keyframes provide high-quality visual anchors that reduce the difficulty of subsequent video editing. Based on this representation, Unicorn addresses inaccurate edit localization via keyframe planner module, and improves cross-shot identity consistency via in-context keyframe editing module.

Concretely, Unicorn represents each shot with a keyframe and uses a VLM-based planner to identify the shots relevant to the editing request. Only their keyframes are edited, while the others remain unchanged. These shot-level anchors explicitly indicate where the edit should and should not occur, preventing unintended changes to irrelevant shots.

To further address cross-shot consistency, Unicorn introduces in-context image editing to propagate the edit from an editing example to other keyframes. Instead of editing each keyframe independently, the model uses an edited anchor example to guide the edits of other keyframes, thereby maintaining the consistency of the key subject across edited keyframes from multiple shots. To incorporate the edited keyframes into the video editing process, we provide them as conditions to the video editing model, enabling localized edit control and coherent appearance across shots.

Due to the lack of systematic benchmarks for multi-shot video editing, we introduce **MultiShotBench**, the first multi-shot video editing benchmark with expert shot-level annotations. We design a multi-dimensional evaluation protocol to assess shot-level edit accuracy, cross-shot consistency, and overall video quality.

We evaluate Unicorn on MultiShotBench to verify its effectiveness in multi-shot video editing, achieving a state-of-the-art overall score above all methods. To further assess its generalizability to single-shot scenarios, we conduct experiments on OpenVEBench (He et al. 2025) and IVEBench (Chen et al. 2025b), where Unicorn also achieves state-of-the-art overall performance. Ablation studies further validate the effectiveness of each module. We anticipate that this work could inspire future research in controllable multi-shot video editing.

Our contributions are as follows:

- A novel, controllable multi-shot video editing framework, **Unicorn**, that uses shot-level edited keyframe anchors to achieve accurate localization.
- An in-context keyframe propagation mechanism to maintain subject identity across different shots.
- A state-of-the-art video editing model that demonstrates superior performance on both multi-shot and single-shot video editing benchmarks.
- **MultiShotBench**, the first real-world multi-shot video editing benchmark with expert shot-level annotations and multi-dimensional evaluation protocols.

Related Work

Instruction-Guided Video Editing

Early diffusion-based methods extend image editing priors to videos through attention control or feature propagation, as exemplified by FateZero (Qi et al. 2023), TokenFlow (Geyer et al. 2023), and Rerender-A-Video (Yang et al. 2023). More recent end-to-end methods, including InsViE (Wu et al. 2025b), Ditto (Bai et al. 2025a), and Kiwi-Edit (Lin et al. 2026), improve instruction following and visual controllability through large-scale training data and reference guidance. Lance (Fu et al. 2026) and Bernini (Liu et al. 2026) are unified models that support both video generation and editing.

Reference-Image-Guided Video Editing

Reference-image-guided methods use edited frames or external images for more precise visual control. AnyV2V (Ku et al. 2024) edits the first frame externally and reconstructs the video with an image-to-video model, while VACE (Jiang et al. 2025) supports reference-to-video and video-to-video editing through a unified interface. NOVA (Pan et al. 2026b) adopts pair-free sparse-to-dense control: user-edited keyframes provide semantic guidance, while dense motion and texture cues from the source video preserve fidelity and temporal coherence.

Multi-Shot Video Modeling

Recent multi-shot research primarily focuses on video generation. Video Storyboarding (Atzmon et al. 2024) shares diffusion features to preserve character identity across independently generated shots, while MultiShotMaster (Wang et al. 2025b) and InfinityStory (Elmoghany et al. 2026) explicitly model shot arrangement, narrative continuity, and cross-shot entity consistency. These works demonstrate that shot boundaries are meaningful structural events rather than ordinary temporal transitions.

Method

Unicorn Framework Overview

Given a multi-shot source video $\mathcal{V} = \{v_t\}_{t=1}^{T}$, where T denotes the total number of frames, and a video-level editing request p, our goal is to generate an edited video $\hat{\mathcal{V}}$ that fulfills p while preserving irrelevant content.

Unicorn consists of three modules. First, to enable accurate shot-level editing localization, the **shot-aware keyframe planning module** extracts one keyframe per shot, and generates image editing prompts for these keyframes. Second, to preserve the consistency across shots, the **cross-shot consistency module** incorporates in-context image editing (Zhang et al. 2025b) across the affected keyframes. Third, to incorporate the localized and consistent keyframe edits into video editing, the **keyframe-anchored video editing module** conditions the video editor on the resulting edited keyframes to produce a controllable and cross-shot consistent edited video.

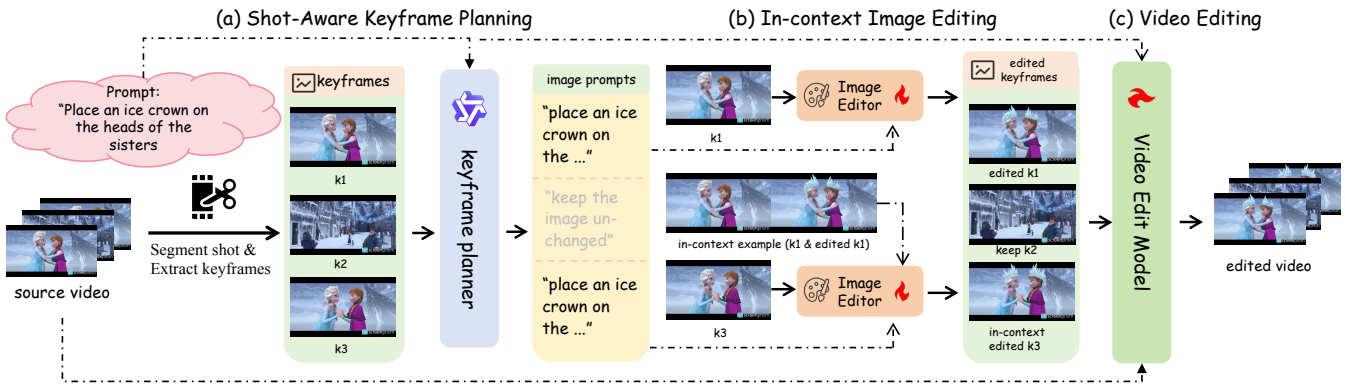


Figure 2: Overview of Unicorn. (a) The shot-aware keyframe planner segments the source video, extracts shot-wise keyframes, and localizes the video-level editing request. (b) In-context image editing transfers a shared visual transformation across affected keyframes while preserving irrelevant shots. (c) The edited keyframes and source video condition the video editing model to produce a controllable and cross-shot consistent output.

Shot-Aware Keyframe Planning

As shown in Fig. 2(a), we first apply a shot detector to partition the input video into S shots:

$$\mathcal{D}_{\text{shot}}(\mathcal{V}) = \{\mathcal{V}^{(1)}, \ldots, \mathcal{V}^{(S)}\}. \tag{1}$$

For each shot $\mathcal{V}^{(s)}$, we select its first frame as the representative keyframe k_s.

To accurately identify which shots should be edited, we leverage a vision-language model (VLM) (Bai et al. 2025b) as a keyframe planner to determine whether each shot-level keyframe is relevant to the video editing instruction. For the shots which are relevant to the editing instruction, the planner generates a shot-specific image editing instruction p_s for subsequent keyframe editing. For irrelevant shots, it explicitly outputs a preservation directive p_{keep}: "Keep the image unchanged.", ensuring that the corresponding keyframes remain unedited during the subsequent image editing stage.

Cross-Shot Consistency via In-Context Image Editing

Through keyframe planning, we can accurately identify the shots that should be edited. However, directly applying image editing to each keyframe may produce inconsistent edited objects across shots, as shown in Fig. 1. This inconsistency arises because textual descriptions often underspecify fine-grained visual appearance, while the editing process is inherently stochastic, leading to diverse and inconsistent edits across shots.

Inspired by recent in-context image editing methods (Zhang et al. 2025b; Chen et al. 2025a), we use a source–edited keyframe pair as an example that guides the remaining affected keyframes toward a consistent edit.

Let

$$\mathcal{A} = \{s \mid p_s \neq p_{keep}\}. \tag{2}$$

denote the affected shots, and let the first affected shot $a \in \mathcal{A}$ be the anchor shot. We first edit keyframe k_a using an image editor $\mathcal{E}_{\text{img}}$ to obtain $\hat{k}_a$. For each remaining affected shot, the pair $(k_a, \hat{k}_a)$ serves as an example that guides the subsequent edit.

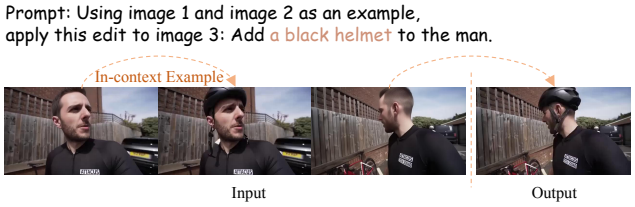


Figure 3: An in-context editing example.

The complete edited keyframe set is:

$$\hat{k}_s = \begin{cases} \mathcal{E}_{\text{img}}(k_s, p_s), & s = a, \\ \mathcal{E}_{\text{img}}(k_a, \hat{k}_a, k_s, p_s), & s \in \mathcal{A} \setminus \{a\}, \\ k_s, & s \notin \mathcal{A}. \end{cases} \tag{3}$$

As shown in Fig. 3, the source–edited example from the first shot establishes the specific appearance of a black helmet. Based on this in-context example, the image editor adds the same helmet to another keyframe, preserving its color, shape, and head-relative placement across shots.

Building on the above mechanism, we train an image editing model based on Qwen-Image-Edit-2511 (Wu et al. 2025a) that supports both single-image editing and in-context image editing. In Figure 2(b), the first keyframe is edited using single-image editing to obtain the in-context example; conditioned on this example, the third keyframe is then edited using in-context editing, preserving the consistency of the ice crown across the two keyframes. By incorporating the in-context editing mechanism, the model improves the consistency of edited objects and backgrounds across multiple affected keyframes.

Keyframe-Anchored Multi-Shot Video Editing

After obtaining the shot-level keyframes from the preceding stages, we design a keyframe-anchored video editing model to perform the video editing process, as illustrated in Fig. 2(c). These edited keyframes provide shot-level visual guidance into the video editing model to improve the edit accuracy and cross-shot consistency.

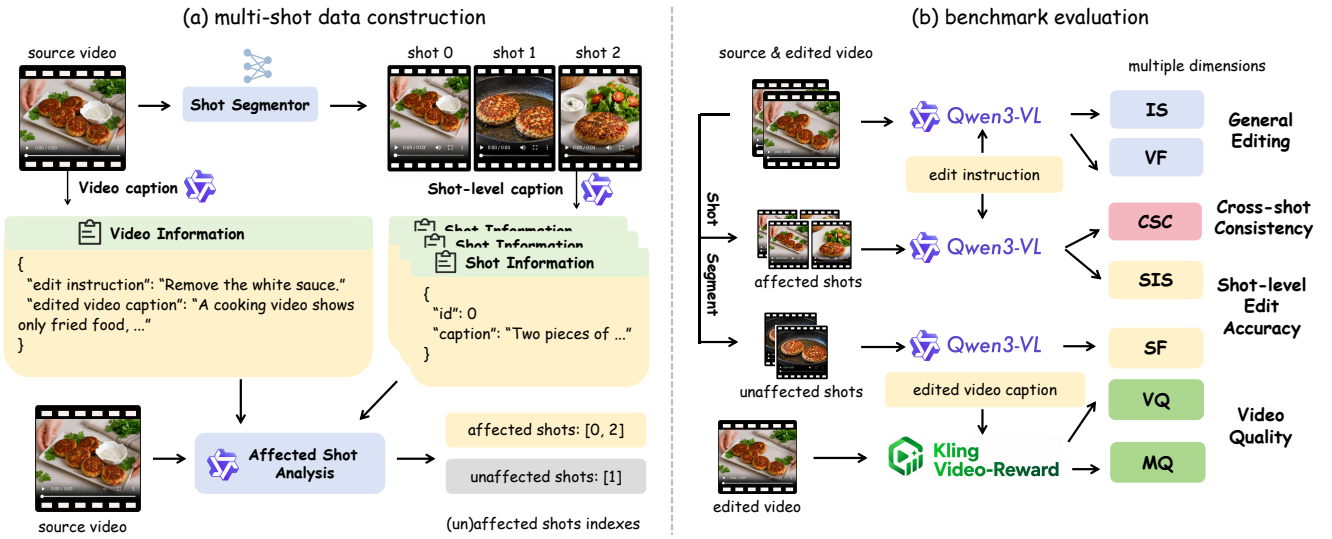


Figure 4: Overview of the MultiShotBench construction pipeline and benchmark evaluation.

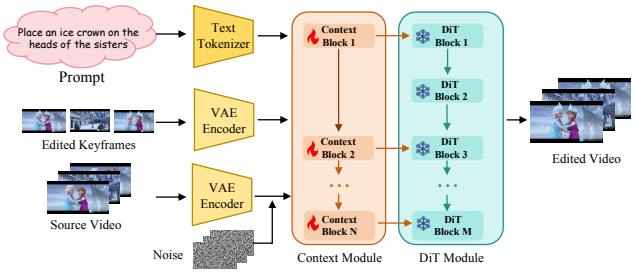


Figure 5: Multi-reference conditioning in the keyframe-anchored video editing module.

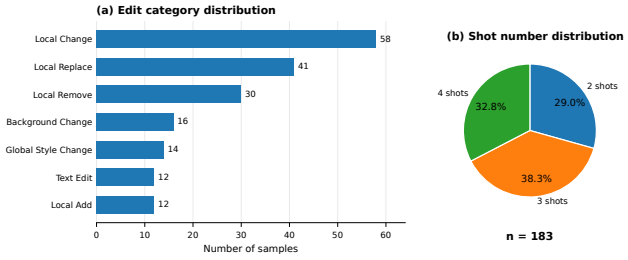


Figure 6: Distribution statistics of MultiShotBench.

As shown in Fig. 5, we build a multi-reference video editing model based on VACE (Jiang et al. 2025). The model is conditioned on the source video $\mathcal{V}$, the editing instruction p, and the ordered edited keyframes $\hat{\mathcal{K}} = \{\hat{k}_1, \ldots, \hat{k}_S\}$. Let z^{src} and $z^{\text{key}}_{1:S}$ denote the latent representations of the source video and edited keyframes, respectively, and let c denote the text embedding of p. We train the video editor using flow matching (Lipman et al. 2022) with target latent z_0, Gaussian noise ϵ, timestep t, and interpolated latent $z_t = (1-t)\epsilon + tz_0$:

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{t,z_0,\epsilon}\left[\left\| v_\theta(z_t, t, c, z^{\text{src}}, z^{\text{key}}_{1:S}) - (z_0 - \epsilon)\right\|_2^2\right], \tag{4}$$

where v_θ denotes the predicted velocity field.

MultiShotBench

Benchmark Construction

Figure 4 provides an overview of MultiShotBench. We collect candidate multi-shot videos from Shot2Story (Han et al. 2023), MMBench-Video (Fang et al. 2024), and Uni-VBench (Wei et al. 2026), and segment them into shots using TransNetV2 (Soucek and Lokoc 2020). A VLM verifies the shot boundaries, captions each valid shot, and generates a video-level editing request, expected edited captions, and affected-shot labels. VLM-based filtering and expert review then remove samples with ambiguous requests or unreliable annotations. Following this construction pipeline, MultiShotBench comprises 183 real-world multi-shot videos. Each sample includes an edit instruction, a video-level caption, verified shot boundaries, a caption for each shot, an expected post-edit video caption, and labels identifying the shots affected by the edit instruction. The benchmark distribution is summarized in Fig. 6. We will publicly release MultiShotBench to facilitate future research in multi-shot video editing.

Evaluation Metrics

Since multi-shot video editing introduces additional challenges beyond single-shot editing, existing video editing metrics are insufficient for a comprehensive evaluation. To capture the unique challenges posed by multi-shot video editing, we introduce the following metrics. **Shot-level Instruction Satisfaction (SIS)** and **Shot-level Fidelity (SF)** jointly evaluate whether an edit is accurately localized to the intended shots, while **Cross-shot Consistency (CSC)** measures whether the edited content maintains a consistent appearance across different affected shots. We further adopt the general video editing metrics **Instruction Satis-**

Table 1: Quantitative results on MultiShotBench. Higher is better. Best results are in **bold**, and second-best results are underlined.

Method	Params	Overall↑	Shot-level Edit Accuracy		Cross-shot Consistency	General Editing		Video Quality	
			SIS↑	SF↑	CSC↑	IS↑	VF↑	VQ↑	MQ↑
VACE (Jiang et al. 2025)	14B	3.158	2.933	2.689	4.171	3.043	2.755	2.183	2.043
AnyV2V (Ku et al. 2024)	1.3B	2.610	2.574	1.760	3.469	2.859	2.163	2.440	2.248
LucyEdit (Decart AI Team 2025)	5B	3.135	2.643	3.367	3.760	2.864	3.500	1.834	1.887
PropFly (Seo et al. 2026)	14B	3.180	2.258	4.000	3.539	2.272	**4.462**	2.207	2.133
NOVA (Pan et al. 2026b)	1.3B	3.476	2.763	4.425	3.540	3.054	4.451	2.537	2.306
Ditto (Bai et al. 2025a)	14B	3.261	3.159	2.679	4.100	3.557	2.983	2.554	2.344
Lance (Fu et al. 2026)	3B	3.626	3.317	3.755	4.141	3.772	3.859	**2.703**	1.976
OmniWeave (Pan et al. 2026a)	8.3B	3.324	2.928	3.368	3.897	3.380	3.973	2.132	2.126
Kiwi (Lin et al. 2026)	5B	3.465	3.108	3.245	4.233	3.712	3.592	2.346	**2.482**
Bernini (Liu et al. 2026)	14B	3.905	3.813	4.160	4.267	4.342	4.152	2.123	2.019
Unicorn-Tiny (Ours)	**1.3B**	3.992	3.800	4.189	**4.359**	**4.592**	4.212	2.619	2.323
Unicorn-Base (Ours)	**14B**	**4.102**	**3.919**	**4.623**	4.197	4.554	4.418	2.701	2.467

faction (IS) and **Video Fidelity (VF)** to assess the overall instruction-following ability and content preservation ability. In addition, **Visual Quality (VQ)** and **Motion Quality (MQ)** jointly evaluate the perceptual quality of the generated video. The **Overall score** is computed as a linear combination of the aforementioned metrics, providing a comprehensive measure of multi-shot video editing performance.

Specifically, we first partition both the source and edited videos using a shot segmenter. For SIS, each affected shot is evaluated independently to determine whether it fulfills the editing instruction. For SF, each unaffected shot in the edited video is compared with its source counterpart to determine whether it remains unchanged. To compute CSC, the edited results of all affected shots are jointly evaluated to determine whether the target object or edited attribute remains visually consistent across different shots. IS and VF are instead computed on the complete source–edited video pair to assess global instruction following and content preservation. Finally, we feed the edited video with its corresponding edited video caption into VideoReward (Liu et al. 2025) to obtain the VQ and MQ scores, which assess the visual and motion quality of the edited video. All metric scores are normalized to a 0–5 scale.

Experiments

Experimental Setup

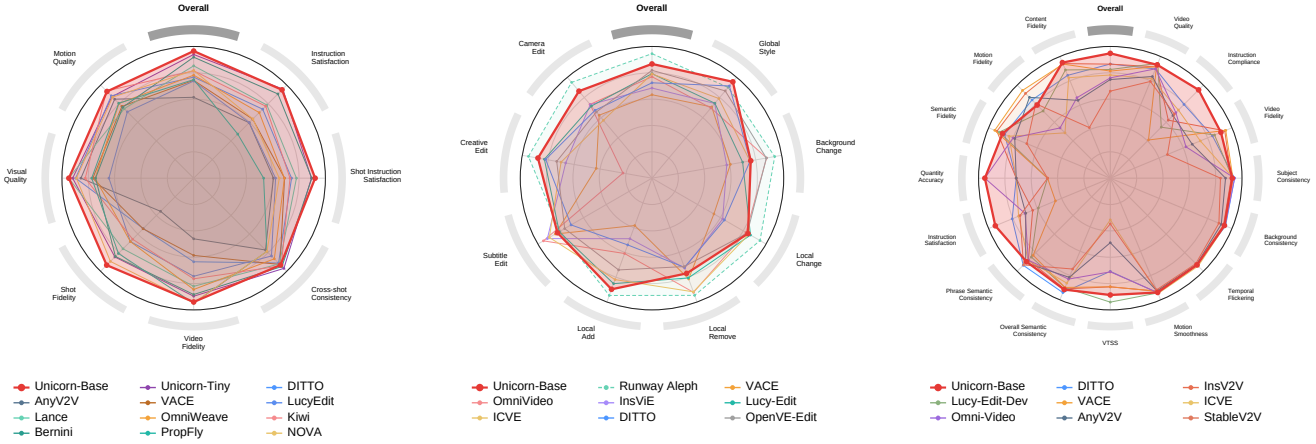
Implementation details. Our Unicorn-Base model is initialized from Wan2.2 14B (Wang et al. 2025a), while Unicorn-Tiny is initialized from Wan2.1 1.3B. We train the model on 800K multi-shot videos, each comprising 81 frames at 15 fps and 1–6 shots. Training is conducted for 50K steps on 32 GPUs with a learning rate of 5×10^{-5} and a batch size of 1. During inference, we solve the learned flow ODE with the UniPC multistep solver for 40 sampling steps, using a classifier-free guidance scale of 7.5. Our in-context image editing model is initialized from Qwen-Image-Edit-2511 (Wu et al. 2025a). We construct 100K in-context image editing training examples and train the model for 20K steps with a learning rate of 2×10^{-5} and a batch size of 1. During inference, we use the FlowMatch Euler solver for 40 sampling steps with a classifier-free guidance scale of 4.0. Both models are fine-tuned with LoRA using rank 32. The training data for both the video editing model and the in-context image editing model are constructed from publicly available datasets. We will release the data construction pipelines and the training data to facilitate future research in this area.

Benchmark and evaluation metrics. We conduct the main evaluation on MultiShotBench, which focuses on controllable editing of real multi-shot videos, using the aforementioned metrics. Instruction-satisfaction and fidelity metrics are scored by Qwen3-VL-32B (Bai et al. 2025b). We further evaluate generalization to single-shot video editing on OpenVEBench (He et al. 2025) and IVEBench (Chen et al. 2025b), as detailed in the single-shot generalization analysis.

Compared methods. We compare Unicorn with three categories of representative video editing methods: instruction-guided methods, including LucyEdit (Decart AI Team 2025), Ditto (Bai et al. 2025a), OmniWeave (Pan et al. 2026a), and Kiwi (Lin et al. 2026); reference-image-guided methods, including VACE (Jiang et al. 2025), AnyV2V (Ku et al. 2024), PropFly (Seo et al. 2026), and NOVA (Pan et al. 2026b); and unified models, including Bernini (Liu et al. 2026) and Lance (Fu et al. 2026).

Quantitative Results

Table 1 reports the quantitative results on MultiShotBench, with the visualization in Fig. 7(a). Unicorn-Base achieves the highest overall score of 4.102, while Unicorn-Tiny ranks second with 3.992. Unicorn-Base obtains the best SIS and SF, showing accurate editing of affected shots and strong



(a) MultiShotBench **(b) OpenVEBench** **(c) IVEBench**

Figure 7: Radar-chart comparisons on MultiShotBench, OpenVEBench, and IVEBench. Unicorn-Base is highlighted in red.

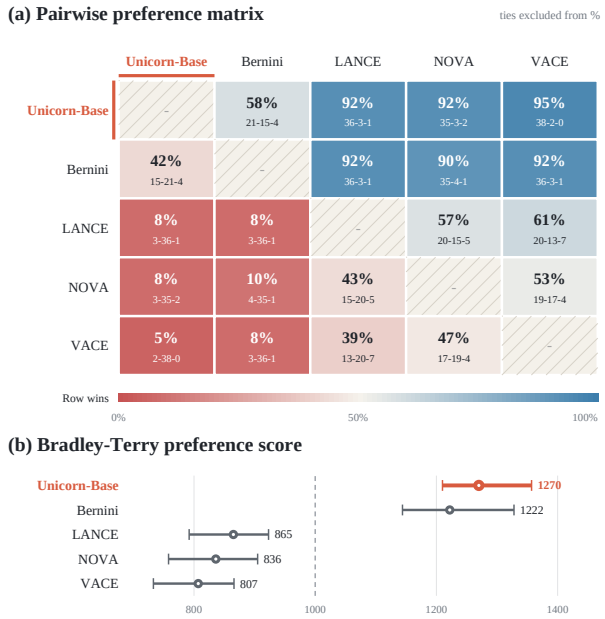


Figure 8: Human preference study with pairwise win rate.

preservation of irrelevant shots. Both model scales lead the comparison in complementary dimensions, validating the effectiveness and scalability of our method for multi-shot video editing.

User Study

Relying solely on VLM-based metrics may introduce biases and cannot fully capture human preferences. Therefore, we conduct a user study to complement and extend the VLM-based evaluation. Specifically, we randomly sample 40 cases from MultiShotBench and adopt an expert voting protocol: video evaluation experts pairwise compare anonymized outputs presented in random order and vote for the preferred result. Unicorn exceeds a 50% win rate against every competitor in Fig. 8(a) and ranks first by the Bradley–Terry score (Bradley and Terry 1952) in Fig. 8(b), showing that improvements measured by automatic metrics align with human preferences.

Table 2: Ablation study on MultiShotBench.

Variant	Shot Det.	VLM	ICEdit	Train Video Model	Overall↑
w/o Shot Det.	×	✓	✓	✓	3.848
w/o VLM	✓	×	✓	✓	4.027
w/o ICEdit	✓	✓	×	✓	4.039
w/o Video Model Training	✓	✓	✓	×	3.934
Unicorn-Base	✓	✓	✓	✓	**4.102**

Analysis

Ablation Study

We ablate four components to isolate their roles in the multi-shot editing pipeline. Table 2 reports the updated Overall results. Detailed results for each metric are provided in the supplementary material.

Shot detection. The w/o Shot Det. variant uses the video's first frame as its sole reference image. Its performance degradation shows that shot-level keyframe extraction effectively injects multi-shot structure into video editing.

VLM-based planning. The w/o VLM variant applies the video-level instruction to every keyframe. Without planning, SF score drops from 4.623 to 3.943, indicating that shots that should remain unchanged are unintentionally edited. This result demonstrates that keyframe planning improves edit localization and helps preserve irrelevant shots.

In-context keyframe editing. The w/o ICEdit variant edits each affected keyframe independently. Its CSC score drops from 4.197 to 4.049, the lowest among all ablation variants.



Figure 9: Qualitative comparison with representative methods on MultiShotBench.

This confirms that the in-context editing module effectively improves cross-shot consistency.

Video model training. The w/o Video Model Training variant uses the base video editing model. Its lower performance shows that training enables video editing models to better utilize the visual information provided by edited keyframes.

Generalization to Single-Shot Video Editing

Although Unicorn is designed for multi-shot video editing, we further evaluate it on OpenVEBench (He et al. 2025) and IVEBench (Chen et al. 2025b) to assess its performance on single-shot editing tasks. As shown in Fig. 7(b)–(c), Unicorn achieves the best performance among all compared open-source methods on both benchmarks. These results demonstrate that, while achieving state-of-the-art performance on multi-shot video editing, Unicorn also retains strong single-shot editing capability, highlighting the generalizability of our method.

Computational Cost Analysis

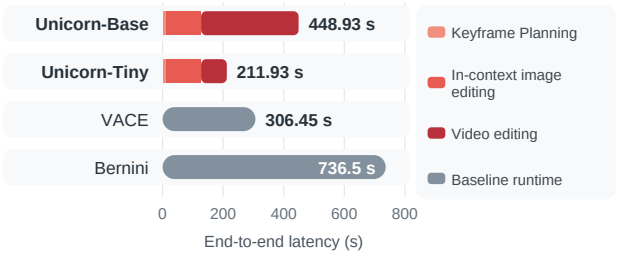


Figure 10: End-to-end latency comparison.

We evaluated the runtime of Unicorn and the competing methods on a single NVIDIA H800 GPU. For a fair comparison, all methods used the same input resolution of 640 × 360 and 40 inference steps. As shown in Fig. 10, Unicorn does not introduce excessive additional time overhead compared with representative video editing methods, despite incorporating shot-aware planning and cross-shot consistent editing.

Qualitative Analysis

Figure 9 presents three representative editing scenarios as examples. For watercolor stylization, NOVA, Lance, and VACE treat the two shots unevenly, while Bernini introduces content drift; Unicorn preserves the layout and cat identity under a coherent style. For object replacement, Lance produces a handled mug and VACE a malformed object, whereas Unicorn renders the same insulated cup across different shots. For background replacement, several baselines retain parts of the original environment or alter the vehicle, while Unicorn transforms all three shots into desert landscapes without changing the SUV. The latter two cases particularly demonstrate consistent edited appearance across disparate shots.

Conclusion

In this work, we presented Unicorn, the first controllable framework for multi-shot video editing. Unicorn leverages edited shot-wise keyframes as anchors, addresses inaccurate edit localization via keyframe planner module, and improves cross-shot consistency via in-context keyframe editing module. We also introduce MultiShotBench, a benchmark designed to evaluate multiple dimensions in this field. Experiments and analyses show that Unicorn achieves state-of-the-art performance across all methods on multiple benchmarks. Given the substantial challenges in multi-shot video editing, we adopt this three-stage framework to address them. In future work, we will explore the feasibility of developing an end-to-end model for multi-shot video editing.

References

Atzmon, Y.; Gal, R.; Tewel, Y.; Kasten, Y.; and Chechik, G. 2024. Multi-Shot Character Consistency for Text-to-Video Generation. *IEEE Transactions on Image Processing*, 35: 4802–4815.

Bai, Q.; Wang, Q.; Hao, O.; Yu, Y.; Wang, H.; Wang, W.; Cheng, K.; Ma, S.; Zeng, Y.; Liu, Z.; et al. 2025a. Scaling Instruction-Based Video Editing with a High-Quality Synthetic Dataset. *arXiv.org*.

Bai, S.; Cai, Y.; Chen, R.; Chen, K.-q.; Chen, X.-H.; Cheng, Z.; Deng, L.; Ding, W.; Fang, R.; Gao, C.; et al. 2025b. Qwen3-VL Technical Report. *arXiv.org*.

Black Forest Labs. 2025. FLUX.2: Frontier Visual Intelligence. https://bfl.ai/blog/flux-2.

Bradley, R. A.; and Terry, M. E. 1952. Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons. *Biometrika*, 39(3/4): 324.

Chen, L.; Mao, Q.; Gu, Y.; and Shou, M. Z. 2025a. Edit Transfer: Learning Image Editing via Vision In-Context Relations. *arXiv.org*.

Chen, Y.; Zhang, J.; Hu, T.; Zeng, Y.; Xue, Z.; He, Q.; Wang, C.; Liu, Y.; Hu, X.; and Yan, S. 2025b. IVEBench: Modern Benchmark Suite for Instruction-Guided Video Editing Assessment. In *arXiv.org*.

Decart AI Team. 2025. Lucy Edit: Open-Weight Text-Guided Video Editing. Technical report, Decart AI.

Elmoghany, M.; Zhao, L.; Shen, X.; Mukherjee, S.; Zhou, Y.; Wu, G.; Lai, V. D.; Yoon, S.; Rossi, R.; Rashwan, A.; et al. 2026. InfinityStory: Unlimited Video Generation with World Consistency and Character-Aware Shot Transitions. *arXiv.org*.

Fang, X.; Mao, K.; Duan, H.; Zhao, X.; Li, Y.; Lin, D.; and Chen, K. 2024. MMBench-Video: A Long-Form Multi-Shot Benchmark for Holistic Video Understanding. *Neural Information Processing Systems*, 89098–89124.

Fu, F.; Huang, M.; Wu, S.; Jiang, Y.; Huo, Y.; Li, H.; Song, Y.; Ding, F.; Guo, J.; He, Q.; et al. 2026. Lance: Unified Multimodal Modeling by Multi-Task Synergy. *arXiv.org*.

Geyer, M.; Bar-Tal, O.; Bagon, S.; and Dekel, T. 2023. TokenFlow: Consistent Diffusion Features for Consistent Video Editing. *International Conference on Learning Representations*.

Han, M.; Yang, L.; Chang, X.; and Wang, H. 2023. Shot2Story: A New Benchmark for Comprehensive Understanding of Multi-shot Videos. *arXiv:2312.10300*.

He, H.; Wang, J.; Zhang, J.; Xue, Z.; Bu, X.; Yang, Q.; Wen, S.; and Xie, L. 2025. OpenVE-3M: A Large-Scale High-Quality Dataset for Instruction-Guided Video Editing. *arXiv.org*. Code/project: https://github.com/lewandofskee/OpenVE.

Jiang, Z.; Han, Z.; Mao, C.; Zhang, J.; Pan, Y.; and Liu, Y. 2025. VACE: All-in-One Video Creation and Editing. *IEEE International Conference on Computer Vision*, 17191–17202.

Ku, M.; Wei, C.; Ren, W.; Yang, H.; and Chen, W. 2024. AnyV2V: A Tuning-Free Framework For Any Video-to-Video Editing Tasks. *Trans. Mach. Learn. Res.*

Lin, Y.; Liang, G.; Zeng, Z.; Bai, Z.; Chen, Y.; and Shou, M. Z. 2026. Kiwi-Edit: Versatile Video Editing via Instruction and Reference Guidance. *arXiv.org*.

Lipman, Y.; Chen, R. T. Q.; Ben-Hamu, H.; Nickel, M.; and Le, M. 2022. Flow Matching for Generative Modeling. In *International Conference on Learning Representations*.

Liu, C.; Chen, J.; Li, L.; Chi, L.; Sun, M.; Li, Z.; Fu, Y.; Guo, R.; Wu, Y.; Bai, G.; et al. 2026. Bernini: Latent Semantic Planning for Video Diffusion. *arXiv.org*.

Liu, J.; Liu, G.; Liang, J.; Yuan, Z.; Liu, X.; Zheng, M.; Wu, X.; Wang, Q.; Qin, W.; Xia, M.; et al. 2025. Improving Video Generation with Human Feedback. *Neural Information Processing Systems*.

Pan, K.; Tian, Q.; Zhang, J.; Kong, W.; Xiong, J.; Long, Y.; Zhang, S.; Qiu, H.; Wang, T.; Lv, Z.; et al. 2026a. OmniWeaving: Towards Unified Video Generation with Free-form Composition and Reasoning. *arXiv.org*.

Pan, T.; Dai, J.; Yuan, C.; Lv, Z.; Yang, B.; Yin, H.; Li, C.; Lyu, J.; Shan, C.; and Si, C. 2026b. NOVA: Sparse Control, Dense Synthesis for Pair-Free Video Editing. *arXiv.org*.

Qi, C.; Cun, X.; Zhang, Y.; Lei, C.; Wang, X.; Shan, Y.; and Chen, Q. 2023. FateZero: Fusing Attentions for Zero-shot Text-based Video Editing. *IEEE International Conference on Computer Vision*, 15886–15896.

Seo, W.; Moon, J.; Lee, J.; Kim, S. Y.; and Kim, M. 2026. PropFly: Learning to Propagate via On-the-Fly Supervision from Pre-trained Video Diffusion Models. In *arXiv.org*.

Soucek, T.; and Lokoc, J. 2020. TransNet V2: An Effective Deep Network Architecture for Fast Shot Transition Detection. *ACM Multimedia*.

Wang, A.; Ai, B.; Wen, B.; Mao, C.; Xie, C.-W.; Chen, D.; Yu, F.; Zhao, H.; Yang, J.; Zeng, J.; et al. 2025a. Wan: Open and Advanced Large-Scale Video Generative Models. *arXiv.org*.

Wang, Q.; Shi, X.; Li, B.; Bian, W.; Liu, Q.; Lu, H.; Wang, X.; Wan, P.; Gai, K.; and Jia, X. 2025b. MultiShotMaster: A Controllable Multi-Shot Video Generation Framework. *arXiv.org*.

Wang, T.; Zou, M.; Wu, C.; Qu, X.; Liu, L.; Hu, X.; and Liu, T. 2026. MiVE: Multiscale Vision-Language Features for Reference-Guided Video Editing. In *arXiv.org*.

Wei, C.; Liu, Q.; Ye, Z.; Wang, Q.; Wang, X.; Wan, P.; Gai, K.; and Chen, W. 2025. UniVideo: Unified Understanding, Generation, and Editing for Videos. *arXiv.org*.

Wei, J.; Zhang, X.; Li, Y.; Wang, Y.; Zhang, Y.; Chen, Z.; Tang, Z.; Xu, W.; and Liu, Z. 2026. UniVBench: Towards Unified Evaluation for Video Foundation Models. *arXiv.org*.

Wu, C.; Li, J.; Zhou, J.; Lin, J.; Gao, K.; Yan, K.; Yin, S.; Bai, S.; Xu, X.; Chen, Y.; et al. 2025a. Qwen-Image Technical Report. *arXiv.org*.

Wu, Y.; Chen, L.; Li, R.; Wang, S.; Xie, C.; and Zhang, L. 2025b. InsViE-1M: Effective Instruction-based Video Editing with Elaborate Dataset Construction. *IEEE International Conference on Computer Vision*, 16692–16701.

Yang, S.; Zhou, Y.; Liu, Z.; and Loy, C. C. 2023. Rerender A Video: Zero-Shot Text-Guided Video-to-Video Translation. *ACM SIGGRAPH Conference and Exhibition on Computer Graphics and Interactive Techniques in Asia*, 1–11.

Zhang, Z.; Long, F.; Li, W.; Qiu, Z.; Liu, W.; Yao, T.; and Mei, T. 2025a. Region-Constraint In-Context Generation for Instructional Video Editing. *arXiv.org*.

Zhang, Z.; Xie, J.; Lu, Y.; Yang, Z.; and Yang, Y. 2025b. In-Context Edit: Enabling Instructional Image Editing with In-Context Generation in Large-Scale Diffusion Transformers. In *Neural Information Processing Systems*.