Generative Video Propagation

Shaoteng Liu[1,*], Tianyu Wang[2], Jui-Hsien Wang[2], Qing Liu[2], Zhifei Zhang[2],
Joon-Young Lee[2], Yijun Li[2], Bei Yu[1], Zhe Lin[2], Soo Ye Kim[2,†], Jiaya Jia[3,4,†]

[1] The Chinese University of Hong Kong [2] Adobe Research
[3] The Hong Kong University of Science and Technology [4] SmartMore
https://genprop.github.io//

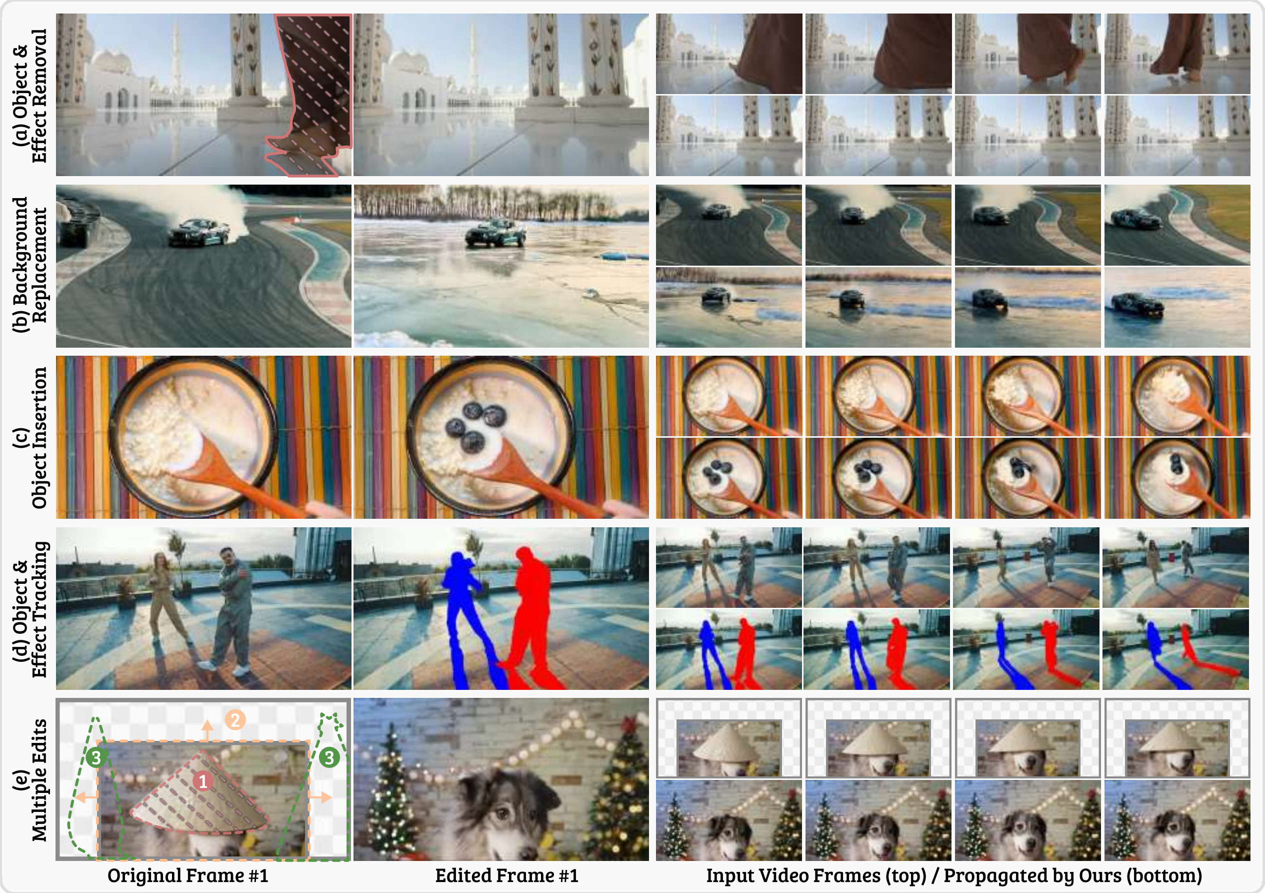

(a) Object & Effect Removal
(b) Background Replacement
(c) Object Insertion
(d) Object & Effect Tracking
(e) Multiple Edits
1
2
3
3
Original Frame #1
Edited Frame #1
Input Video Frames (top) / Propagated by Ours (bottom)

Figure 1. GenProp. We propose a generative video propagation framework (GenProp), which can seamlessly propagate any first frame edit through the video. GenProp supports a wide range of video applications, including (a) complete object removal with effects such as shadows and reflections, (b) background replacement with realistic effects, (c) object insertion where inserted objects have physically plausible motion (i.e., blueberries falling while spoon goes up), (d) tracking of objects and their associated effects, and (e) multiple edits (outpainting, insertion, removal) at a single inference run.

Abstract

Large-scale video generation models have the inherent ability to realistically model natural scenes. In this paper, we

* Work done during an internship at Adobe.
† Co-corresponding authors.

demonstrate that through a careful design of a ***generative video propagation framework****, various video tasks can be addressed in a unified way by leveraging the generative power of such models. Specifically, our framework, GenProp, encodes the original video with a selective content encoder and propagates the changes made to the first frame using an image-to-video generation model. We propose a data generation scheme to cover multiple video tasks based on instance-level video segmentation datasets. Our model is trained by incorporating a mask prediction decoder head and optimizing a region-aware loss to aid the encoder to preserve the original content while the generation model propagates the modified region. This novel design opens up new possibilities: In editing scenarios, GenProp allows substantial changes to an object's shape; for insertion, the inserted objects can exhibit independent motion; for removal, GenProp effectively removes effects like shadows and reflections from the whole video; for tracking, GenProp is capable of tracking objects and their associated effects together. Experiment results demonstrate the leading performance of our model in various video tasks, and we further provide in-depth analyses of the proposed framework.*

1. Introduction

Recently, large-scale video generation models [7, 16, 17, 31, 34, 41, 46, 57] have shown impressive performance, generating highly realistic videos while being able to simulate the complexities of the real world. In this rapidly evolving domain, following works in video generation have extended the text-to-video (T2V) generation to image-to-video (I2V) [2, 5, 31, 53, 60], and are further exploring various video editing tasks such as video inpainting [65], appearance editing [33, 42], object insertion [30], usually focusing on that specific task. In this paper, we bring a different perspective by observing that many of such video applications can be modeled as a more general *video propagation* problem.

Video propagation itself is not a new concept, with traditional methods often relying on optical flow [9, 43], depth [6, 55], radiance fields [32], and atlases [19, 23] to propagate the changes in sparse intermediate frames (typically the first frame) to the rest of the video. However, such approaches can be prone to error accumulation, leading to limited robustness and generalization ability. Furthermore, they often focus on a single task [9, 32] or entail retraining for a specific task for propagation [27, 30, 33, 61]. In contrast, we define a new problem of *generative video propagation* by leveraging the inherent power of video generation models in modeling real-world scenes.

Our model, *GenProp*, is able to propagate the changes in the first frame to the whole video while keeping other parts consistent to the original video, without requiring any additional motion predictions (e.g., optical flow). This general formulation enables many downstream applications such as removal, insertion, replacement (object and/or background), text-based editing, outpainting and even object tracking, some of which are shown in Fig. 1. We further demonstrate that our model is also able to *expand* the scope of what is usually achievable in each task, specifically: (1) substantial shape modifications in object editing tasks, (2) independent motion of inserted objects in insertion tasks, (3) removal of object effects like shadows and reflections in removal tasks, and (4) accurate tracking of objects along with their associated effects. Note that unlike existing video editing models that often require a dense mask labeling for all individual frames (e.g., for object removal), GenProp does not require any mask input, thanks to the propagation-based approach, greatly simplifying the editing process.

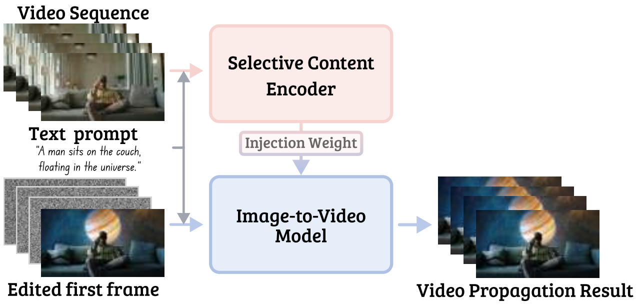


Figure 2. Model Overview. During inference, our framework takes in the original video as input through a selective content encoder (SCE) to retain content in unchanged regions. Changes applied to the first frame are propagated throughout the video using an I2V model while other regions remain intact.

Our model architecture consists of two main components as shown in Fig. 2: the Selective Content Encoder (SCE) that encodes the information of the original video, and the I2V generation model that takes in the edited first frame for propagation. The training objective is to allow SCE to selectively encode the features of the unchanged parts of the video, while preserving the generation capabilities of I2V models to propagate the altered parts. To effectively disentangle these two functions, we introduce a region-aware loss and penalize the gradients within the modified region for SCE, as ideally, SCE should not encode content in the edited area. For training the model, we propose using synthetic data derived from video instance segmentation datasets. As shown in the attention map visualizations in Fig 3, we observe that GenProp indeed attends to the region to be modified and the I2V model is guided to generate (propagate) the new content into those regions. To further aid the model, we incorporate an auxiliary decoder head during training to predict the modified region.

Our contributions are summarized as follows:

- We define a novel problem of *generative video propagation* that aims to propagate various changes in the first frame of the video to the entire video by leveraging the generative power of I2V models.

- We carefully design our model, *GenProp*, with a Selective Content Encoder (SCE), dedicated loss functions and a mask prediction head and propose a synthetic data generation pipeline for training this model.
- Our model supports various downstream applications such as removal, insertion, replacement, editing, and tracking. We observe that it further supports outpainting even without any task-specific data during training.
- Experiment results show that our model outperforms SOTA methods in video editing and object removal while expanding the scope of existing tasks including tracking.

2. Related Work

Video Propagation. Traditional methods are typically designed for a single task and often require retraining for new tasks [9, 20, 32]. Many approaches address propagation by first tracking instance masks, then performing inpainting [24, 25], with segmentation often as the initial step. SAM 2 [38], the current SotA tracking model, can track the masks accurately and efficiently. Some methods rely on optical flow [9, 43] or depth [6, 55] to ensure consistent motion and spatial coherence. CoDef [32] uses deformation fields from the source video to guide edits from the first frame. While these representations aid motion tracking and structural consistency, they add complexity and may limit flexibility, especially with significant shape changes or complex backgrounds. Depth, sketches, and optical flow can also be combined with diffusion models [11, 28, 48, 52, 56].

Diffusion-based Video Editing. Most diffusion-based video editing methods rely on text control, where the primary goal is to make edits that are coherent to text prompts while preserving the unchanged regions of the video. Some methods utilize text-to-image models for zero-shot editing through attention control [6, 13, 22, 36, 40, 47, 49]. Some other works require intermediate variables like optical flows or depth maps to stabilize motion. Others rely on per-case fine-tuning to adapt to specific motion [4, 29, 50, 62], but this approach is typically slow and prone to generate similar results from the original video due to reconstruction tuning. SORA [31] denoises the noised videos under the target description to do editing. These methods are generally limited to altering the appearance rather than making significant changes to object shapes. Additionally, because of unclear attention maps, especially in complex scenes, background changes often lack precision and coherence. InsV2V [8] and EVE [42] edit videos based on text instructions but are also limited to appearance changes. Some recent efforts have attempted to directly edit motion based on text prompts [21, 58, 63], but their resulting video output tends to strike a balance between the text-based guidance and the original video's motion, which can be hard to control.

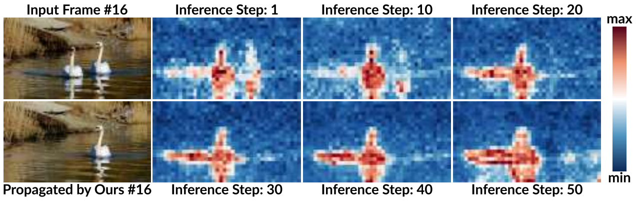


Figure 3. Attention Map Visualization. We observe that the attention maps gradually focus on the regions to be removed and the I2V model is guided to generate new content in those regions.

Image-to-Video Generation and Editing. Image-to-video (I2V) generation models take an input image along with a text prompt to generate a sequence of frames, making them a foundational application in video generation due to its familiarity and versatility. Notable open-source models include Stable Video Diffusion [5], I2VgenXL [60], and SparseCtrl [15], while high-performance commercial models, such as Gen-2, PikaLabs [2], SORA [31], and Movie Gen [34], further push the boundaries in this field.

Several methods propagate edits based on modifications made to the first frame. For example, some works [12, 39, 55, 56] rely on first-frame edits but require auxiliary inputs like optical flow or depth maps for motion continuity. VideoSwap [14] uses sparse key points to control the motion. AnyV2V [26] can also propagate first-frame edits across a video sequence; however, as a training-free framework, its generalization ability is limited. I2VEdit [33], in contrast, necessitates learning motion LoRAs [18] for each video clip, adding computational complexity. Revideo [30], built on Stable Video Diffusion (SVD), enables control over the generation using the edited first frame and a specified motion trajectory. However, its approach involves masking parts of the input video with a black square, which removes significant information and restricts the method in handling complex background edits and large shape alterations.

3. Method

Generative video propagation has the following key challenges: (1) Realism – changes in the first frame should be naturally propagated to the following frames, (2) Consistency – all other regions should remain consistent to the original video, and (3) Generality – the model should be general enough to be applicable to multiple video tasks. In GenProp, we leverage an I2V generation model for (1); we introduce a selective content encoder and a mask prediction decoder and train the model with a region-aware loss to address (2); and we propose a data generation scheme and also benefit from the versatile I2V model for (3).

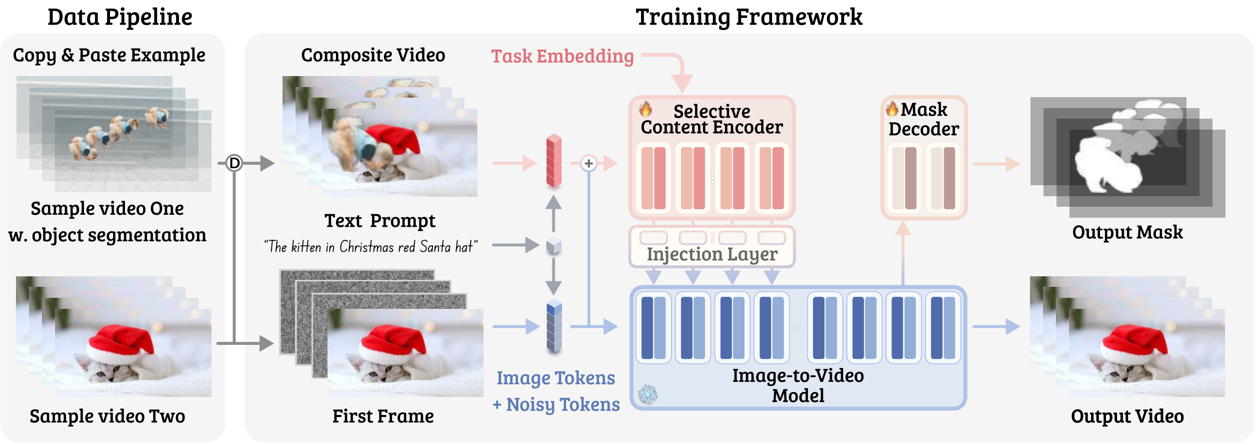


Figure 4. Training Framework of GenProp. Our framework integrates a Selective Content Encoder and a Mask Prediction Decoder on top of the I2V generation model, enforcing the model to propagate the edited region while preserving the content in the original video for all other regions. With synthetic data augmentations and task embeddings, our model is trained to propagate various changes in the first frame.

3.1. Problem Formulation

Given an input video $V = \{v_1, v_2, \ldots, v_T\}$ with T frames, let v'_1 denote the modified first frame. The goal is to propagate this modification, producing a modified video $V' = \{v'_1, v'_2, \ldots, v'_T\}$, where each frame v'_t (for $t = 2, \ldots, T$) retains the modification applied to the key frame v_1 while maintaining consistency in both appearance and motion throughout the sequence. We employ a latent diffusion model that encodes pixel information in the latent space. With a slight abuse of notations, we continue using v_t for this latent representation. In formal terms, during inference, GenProp generates each frame v'_t as:

$$v'_t = \mathcal{G}(\mathcal{E}(V), v'_1, t), \quad \forall t \in \{2, \ldots, T\}, \tag{1}$$

where $\mathcal{G}$ is the I2V generation model guided by the selective content encoder (SCE), $\mathcal{E}(V)$.

For training, we use synthetic data constructed from existing video instance segmentation datasets to create paired samples (details given in Sec. 3.4). We define a data generation operator $\mathcal{D}$ that constructs training data pairs $(v_i, \hat{v}_i)$ from an original video sequence V. Let $\mathcal{D}(V)$ denote the synthetic data generation operator applied to the original video sequence, where:

$$(v_i, \hat{v}_i) \in \mathcal{D}(V), \quad \forall i \in \{1, \ldots, T\}. \tag{2}$$

Then $\hat{V} = \{\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_T\}$ is the synthetic video sequence. GenProp is trained to satisfy the following objective across all frames $i \in \{2, \ldots, T\}$:

$$\min_{\mathcal{E}} \sum_{i=2}^{T} \mathcal{L}(\mathcal{G}(\mathcal{E}(\hat{V}), v_1, i), v_i) \tag{3}$$

where $\mathcal{L}$ is a region-aware loss designed to disentangle the modified and unmodified regions, enforcing stability in the unchanged areas while allowing for accurate propagation in the edited regions (details in Sec. 3.3). To ensure that the final output adheres to real video data distributions, synthetic data is fed exclusively to the content encoder. The I2V generation model, however, uses the original video, preventing the model from inadvertently learning synthetic artifacts.

3.2. Model Design

To preserve the unchanged parts of the original video and only propagate the modified regions, we integrate two additional components to the base I2V model: Selective Content Encoder and Mask Prediction Decoder, as shown in Fig. 4.

Selective Content Encoder. The architecture of our SCE is a replicated version of the initial N blocks of the main generation model, similar to ControlNet [59]. After each encoder block, the extracted features are added to the corresponding features in the I2V model, allowing a smooth and hierarchical flow of content information. The injection layer is one multilayer perceptron with zero initialization which will also be trained. Furthermore, for bidirectional information exchange, the features of the I2V model are fused with the SCE's input before the first block. This lets SCE be aware of the modified regions so that it can selectively encode the information in the unchanged region as intended.

Mask Prediction Decoder. The Mask Prediction Decoder (MPD) is designed to estimate the spatial regions requiring editing, helping the encoder disentangle changes from the unchanged content. While SCE utilizes the initial N blocks of the I2V model, MPD mirrors this by using the final block along with one multilayer perceptron (MLP) as the final layer. It takes the latent representation from the penultimate block, which contains rich spatial and temporal information, and processes it through the MLP layer. This restores the temporal dimension, matching it to the number of video

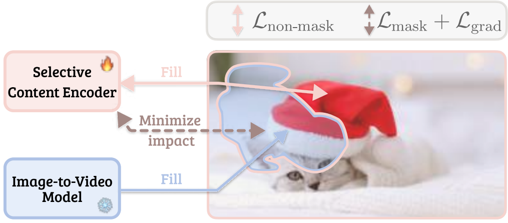

$\mathcal{L}_{\text{non-mask}}$
$\mathcal{L}_{\text{mask}} + \mathcal{L}_{\text{grad}}$
Selective Content Encoder
Fill
Minimize impact
Image-to-Video Model
Fill

Figure 5. Region-Aware Loss. This loss helps the model to disentangle the edited region from the original content.

frames. The final output is trained to match the instance mask of the video via an MSE loss [10] $\mathcal{L}_{\text{MPD}}$. This guides the model to focus on the edited regions and significantly improves the accuracy of the attention maps.

3.3. Region-Aware Loss

In our training process, we use instance segmentation data to ensure that both the edited and unedited regions receive appropriate supervision. We design a Region-Aware Loss (RA Loss), shown in Fig. 5, to balance the loss of both regions, even when the edited areas are proportionally small.

For an input video $\hat{V} = \{\hat{v}_1, \hat{v}_2, \dots, \hat{v}_T\}$ and instance-level masks $M = \{m_1, m_2, \dots, m_T\}$, where $m_t \in \{0,1\}^{H \times W}$ indicates edited regions in frame $\hat{v}_t$, we apply Gaussian downsampling over the spatial dimensions and repeat over the temporal dimension to obtain a mask $\tilde{m}_t$ that is aligned to the shape of the latent representation of the video. The loss is separately computed for the mask and non-mask region, giving:

$$\mathcal{L}_{\text{mask}} = \mathbb{E}_{t \sim \mathcal{U}(1,T)} \left[\mathcal{L}_{\text{d}}(\tilde{m}_t \cdot v_t^{\text{out}}, \tilde{m}_t \cdot v_t) \right] \text{ and } \quad (4)$$

$$\mathcal{L}_{\text{non-mask}} = \mathbb{E}_{t \sim \mathcal{U}(1,T)} \left[\mathcal{L}_{\text{d}}((1 - \tilde{m}_t) \cdot v_t^{\text{out}}, (1 - \tilde{m}_t) \cdot v_t) \right],$$

where $\mathcal{L}_{\text{d}}$ denotes the diffusion MSE loss that measures the pixel-wise error between the generated frame v_t^{out} and ground truth v_t.

To further reduce the SCE's influence on the masked regions, we add a gradient loss $\mathcal{L}_{\text{grad}}$ that minimizes the effect of the masked area in the encoder's input. Instead of computing second-order gradients, we approximate using a finite difference:

$$\Delta f = \frac{f(\mathcal{E}(\hat{V} + \delta)) - f(\mathcal{E}(\hat{V}))}{\delta} \quad (5)$$

where $f(\mathcal{E}(\hat{V}))$ represents the encoder's feature, and δ is a small perturbation. The gradient loss is defined as:

$$\mathcal{L}_{\text{grad}} = \mathbb{E}_{t \sim \mathcal{U}(1,T)} \left[\tilde{m}_t \cdot \|\Delta f\|_2 \right]. \quad (6)$$

The RA Loss $\mathcal{L}$ is a weighted sum of all three terms to ensure sufficient supervision on both masked and unmasked areas:

$$\mathcal{L} = \mathcal{L}_{\text{non-mask}} + \lambda \cdot \mathcal{L}_{\text{mask}} + \beta \cdot \mathcal{L}_{\text{grad}} + \gamma \cdot \mathcal{L}_{\text{MPD}} \quad (7)$$

3.4. Synthetic Data Generation

Creating a large-scale paired video dataset can be costly and challenging especially for video propagation, as it is difficult to encompass all video tasks. To address this, we propose to use synthetic data derived from video instance segmentation datasets. In our training, we use Youtube-VOS [54], SAM-V2 [38], and an internal dataset. However, this data generation pipeline can be applied to any available video instance segmentation dataset. Specifically, we adopt a mix of augmentation techniques to the segmentation data, tailored to various propagation sub-tasks: (1) *Copy-and-Paste*: Objects from one video are randomly segmented and pasted into another, simulating object insertion; (2) *Mask-and-Fill*: The masked region undergoes inpainting, creating realistic edits within selected regions; (3) *Color Fill*: The masked area is filled with specific colors, representing basic object tracking scenarios. For (3), V will be sent to $\mathcal{E}$ and $\hat{v}_1$ will be sent to $\mathcal{G}$ in Eq. 3. Each synthetic data type aligns with a distinct task, enabling our model to generalize across diverse applications. Task embeddings corresponding to these augmentation methods are injected into the model, guiding the model to adapt based on the augmentation type. Note that despite the variety of data creation methods and tasks, the core function of SCE remains consistent: encode the unedited information while the I2V model maintains the generative capabilities to propagate the edited regions. More details about each augmentation technique are provided in the Supplementary Material.

4. Experiments

4.1. Implementation Details

As GenProp is a general framework, we experiment with both a DiT architecture similar to Sora [31] and a U-Net architecture based on Stable Video Diffusion (SVD) [5] as the base video generation model. For the former, it is trained for I2V generation on 32, 64, and 128 frames at 12 and 24 FPS, with a base resolution of 360p. SCE (24 blocks) and MPD are trained while the I2V model is frozen. The results can be upscaled to 720p using a super-resolution model. The learning rate is set to 5e-5 with a cosine-decay scheduler and a linear warmup. An exponential moving average is applied for training stability. A gradient norm threshold of 0.001 prevents training instability. Classifier-free guidance (CFG) value is set to 20, and the data augmentation ratio is set to 0.5/0.375/0.125 for copy-and-paste/mask-and-fill/color fill. In the RA loss, λ is 2.0, β is 1.0, and γ is 1.0. All experiments were conducted on 32/64 NVIDIA A100 GPUs for different architectures. We find that the DiT backbone has a better video generation quality. Our main results are from this DiT variant while the ablation studies are conducted with the SVD-based architecture. Please refer to the Supplementary Material for the results based on SVD.

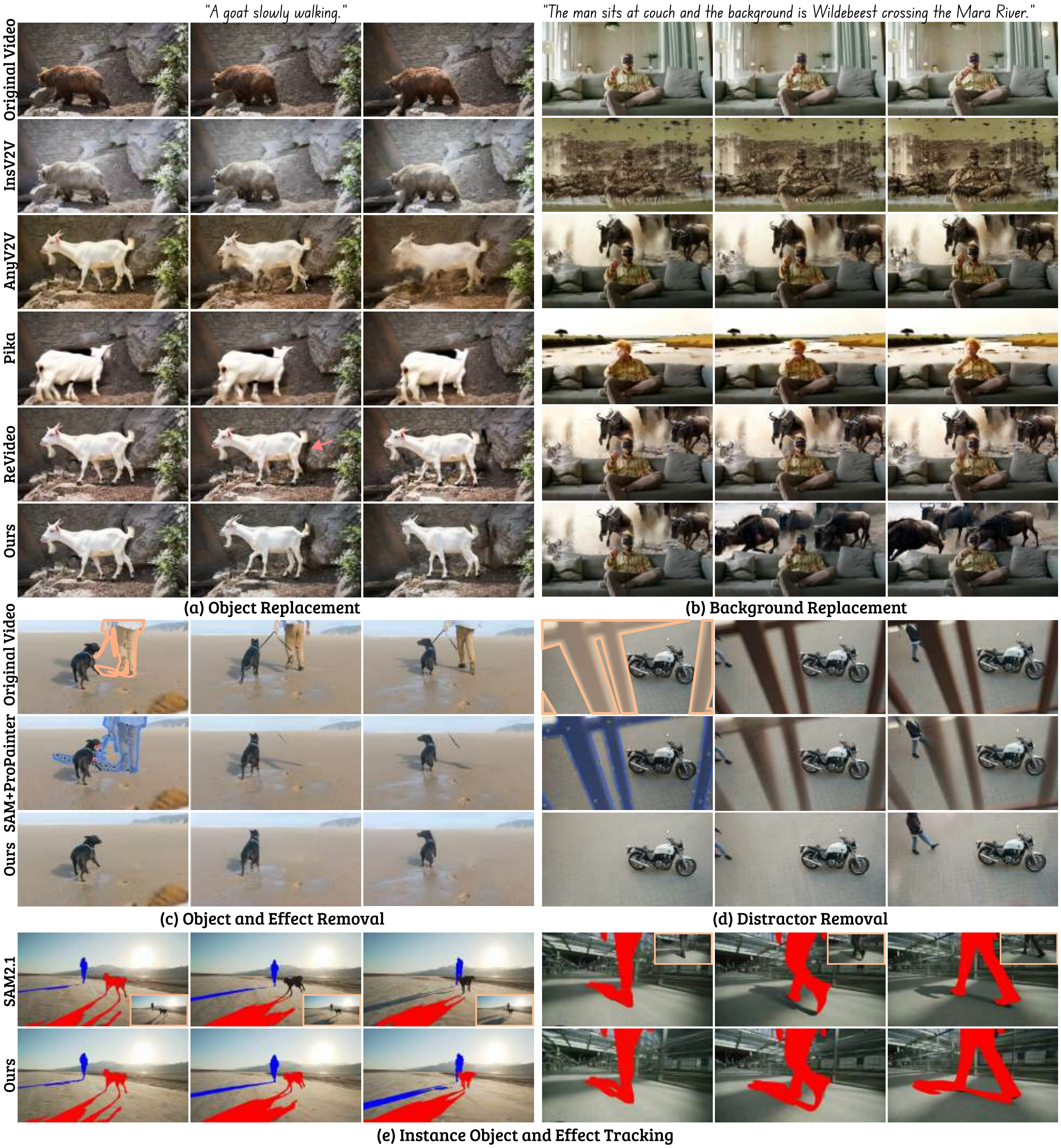

"A goat slowly walking."
"The man sits at couch and the background is Wildebeest crossing the Mara River."
Original Video
InsV2V
AnyV2V
Pika
ReVideo
Ours
(a) Object Replacement
(b) Background Replacement
Original Video
SAM+ProPainter
Ours
(c) Object and Effect Removal
(d) Distractor Removal
SAM2.1
Ours
(e) Instance Object and Effect Tracking

Figure 6. Visual Comparison in Multiple Video Tasks. GenProp demonstrates versatile editing capabilities, (a) allowing seamless modification of objects into those with vastly different shapes with independent motion and (b) enabling background edits. For object removal, GenProp excels at (c) effectively removing object effects together with the object and (d) realistically reconstructing large occluded areas. It is further able to perform instance tracking of objects and their effects when solid color fills are given as the first frame (see (e)).

4.2. Comparisons

As generative video propagation is a new problem, we compare the SotA methods in each of the three sub-tasks of GenProp. Note that our model is able to handle these tasks within the same model and further cover additional tasks such as outpainting as well as combinations of these subtasks as shown in the bottom row of Fig. 1. We provide extensive results in the Supplementary Material.

Method	Classic Test Set					Challenging Test Set				
	PSNR$_m$ ↑	CLIP-T ↑	CLIP-I ↑	GenProp preference %		PSNR$_m$ ↑	CLIP-T ↑	CLIP-I ↑	GenProp Preference %	
				Alignment	Quality				Alignment	Quality
InsV2V [8]	28.999	0.3049	0.9737	60.00	60.00	28.842	0.2906	0.9718	81.82	75.00
AnyV2V [26]	32.090	0.3050	0.9676	95.56	86.67	28.338	0.3302	0.9576	97.78	95.56
Pika [2]	32.568	0.3226	**0.9923**	62.22	55.56	31.329	0.3023	0.9886	88.89	86.67
ReVideo [30]	31.765	0.3196	0.9777	75.56	71.11	29.920	0.3226	0.9798	84.44	82.22
GenProp (Ours)	**33.837**	**0.3229**	0.9825	-	-	**32.163**	**0.3336**	**0.9904**	-	-

Table 1. Video editing benchmark compared to existing models. PSNR$_m$ measures the consistency outside the edited region and Text Alignment and Consistency metrics measure the edit quality. User study shows the percentage of users who preferred Ours over the compared method on alignment (left) and quality (right). GenProp significantly outperforms the other methods on the Challenging Set.

Method	CLIP-I ↑	GenProp Preference %	
		Alignment	Quality
SAM + Propainter	0.9809	82.22	75.56
ReVideo [30]	0.9728	86.36	77.27
GenProp (Ours)	**0.9879**	-	-

Table 2. Object removal comparison to other methods. GenProp outperforms baselines on consistency, alignment, and quality.

Method	CLIP-T ↑	CLIP-I ↑
w/o MPD	0.3252	0.9834
w/o RA Loss	0.3261	0.9825
GenProp (Ours)	**0.3316**	**0.9872**

Table 3. Ablation study. Both MPD and RA loss can improve the success rate of editing and the quality of the output video.

Diffusion-based Video Editing In Fig. 6 (a) and (b), we compare GenProp with other diffusion-based video editing methods, including text-guided and image-guided approaches. InsV2V [8] relies on instruction text for controlling generation. However, due to its limited training data, it struggles with significant shape changes and does not support object insertion. Pika [2] also uses text prompts to edit within a box region, but it performs poorly when the object's shape changes substantially and cannot handle background edits or object insertion. AnyV2V [26] is a training-free method that uses the first frame to guide editing. While it handles appearance changes, it fails when there are large shape or background modifications, often resulting in degradation or ghosting effects. Like InsV2V and Pika, it also cannot insert objects. We use ReVideo [30] to manage large shape changes by first removing an object and then re-inserting it, but this two-stage process has drawbacks. The box-based region can cause blurry boundaries, and object motion is affected by the original point tracking, leading to accumulated errors. Additionally, the box region limits its ability to edit complex backgrounds effectively.

Video Object Removal For object removal, we compare GenProp with a traditional inpainting pipeline, where we cascade two SotA models to achieve a propagation-like inpainting, since traditional methods require a dense mask annotation for all frames: SAM-V2 [38] for mask tracking, then Propainter [64] for inpainting the regions in the estimated masks. As shown in Fig. 6 (c) and (d), GenProp has several advantages: (1) no need for a dense mask annotation as input; (2) removal of object effects like reflections and shadows; (3) removal of large objects and natural filling within large areas.

Video Object Tracking We compare GenProp with SAM-V2 [38] on tracking performance in Fig. 6 (e). Since SAM-V2 is trained on the large-scale SA-V dataset, it is expected that SAM-V2 often produces more precise tracking masks than GenProp. Additionally, GenProp is slower than real-time tracking methods like SAM-V2. However, it has notable advantages. Due to its video generation pretraining, GenProp has a strong understanding of physical rules. As shown in Fig. 6, unlike SAM-V2, which struggles with object effects like reflections and shadows due to limited and biased training data, GenProp can consistently track these effects. This highlights the potential of approaching classic vision tasks with generation-based models.

Quantitative Results We conduct a quantitative evaluation on several test sets. For video editing (reported in Tab. 1), we evaluate on two types of test sets: (1) Classic Test Set, which is TGVE [51]'s DAVIS [35] part and its "Object Change Caption" as the text prompt, focusing on object replacement and appearance editing; (2) Challenging Test Set, which is 30 manually collected videos from Pexels [1] and Adobe Stock [3] including large object replacement, object insertion and background replacement. For (2), the first frame is edited using a commercial photo editing tool. For Pika [2], we use the online boxing tool, running it three times for each result. For ReVideo [30], we select a box region, then to track appearance changes, we use its code to extract the original object's motion points. For edits with significant shape changes, we first remove the original object and then insert the new object, assigning a future trajectory. For assessing the consistency in the unchanged regions, we measure the PSNR outside the edit mask, denoted as PSNR$_m$. For cases with large shape changes, we apply a rough mask over the original and edited regions, only calculating the PSNR on areas outside these masks. For text alignment, we compute the cosine similarity between the CLIP [37] embeddings of the edited frame and the text prompt (CLIP-T) [30, 33, 51]. For result quality, we calculate the distance between CLIP [37] features across frames (CLIP-I) [30, 33, 51]. As shown in Tab. 1, GenProp outperforms the other methods on most metrics, especially on the Challenging Test Set. Pika exhibits better consistency on the Classic Test Set, as its bounding box performs reasonably well when object shapes remain relatively unchanged. ReVideo degrades on multiple objects.

For object removal, we collect 15 videos with complex scenes, including object effects and occlusions, as existing test sets lack coverage of these cases. For SAM, we click on the object and side effects to ensure complete coverage. As shown in Tab. 2, GenProp achieves the highest consistency, while ReVideo may produce bounding box artifacts, and ProPainter struggles with object effects.

As quality metrics often do not correctly capture the realism of the generated results, we use Amazon MTurk [45] to conduct a user study with a total of 121 participants. Each participant views several videos generated by GenProp and a random baseline, along with the original video and the text prompt. They are asked two questions: 1) Which video aligns better with the instructions? 2) Which video is visually better? Participants then select one video for each question. In Tables 1 and 2, we show the percentage of time users prefer Ours over the competing baselines (alignment/quality). GenProp outperforms all baselines by a large margin, especially on the Challenging Test Set.

4.3. Ablation Study

Mask Prediction Decoder In Tab. 3, we evaluate the effect of MPD on the Challenging Test Set, showing that it can improve both Text Alignment and Consistency. As shown in Fig. 7 rows 1 and 2, without MPD, the output mask is often highly degraded, leading to worse removal quality. Without explicit supervision with MPD, the model may be confused which part to propagate and which part to preserve in the original video, causing partially removed objects to reappear in the following frames. MPD helps the disentanglement and both the removal results and predicted masks become more accurate with MPD, allowing for full object removal even with heavy occlusion.

Region-Aware Loss In Tab. 3, we further test the effectiveness of the proposed RA Loss on the Challenging Test Set. A core challenge in GenProp is that SCE can mistakenly select all regions from the original video including the edited areas, weakening the I2V generation ability due to the reconstruction loss. As shown in Fig. 7 rows 3-5, without RA Loss, the original object tends to gradually reappear, hindering the propagation of the first-frame edit (the green motor). With RA Loss, the edited areas are able to be propagated in a stable and consistent way.

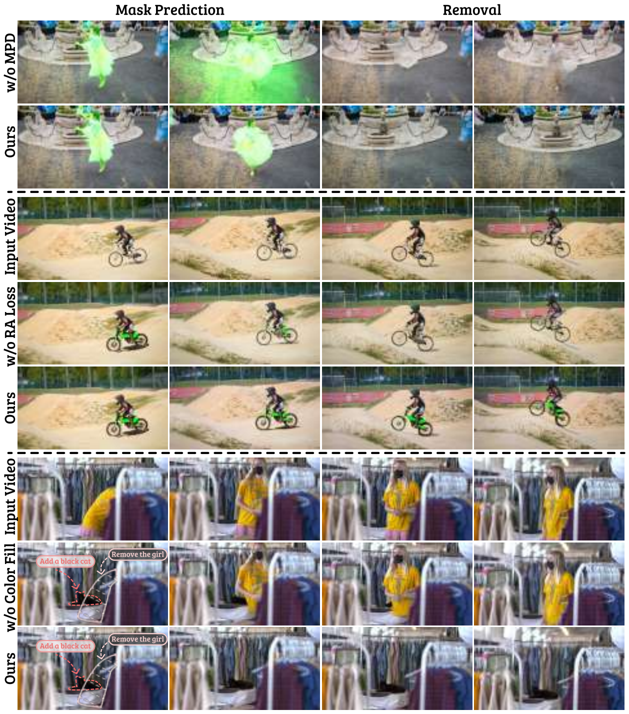


Figure 7. Visual comparison of model variants, showing the effect of MPD (top), RA loss (middle) and Color Fill (bottom).

Color Fill Augmentation Color Fill augmentation is another crucial factor for addressing the propagation failure. While copy-and-paste and mask-and-fill augmentations allow the model to implicitly learn object modifications, replacements, and deletions, color filling explicitly trains it for tracking, guiding the model to maintain modifications made in the first frame throughout the sequence, with the prompt "track colored regions". As shown in Fig. 7 rows 6-8, changing the girl into a small cat is challenging due to the significant shape difference. However, with color fill augmentation, GenProp successfully propagates this large modification throughout the sequence.

5. Conclusion

In this paper, we design a novel *generative video propagation* framework, GenProp, that harnesses the inherent video generation power of I2V models to achieve various downstream applications including removal, insertion and tracking. We demonstrate its potential by showing that it is able to expand the range of achievable edits (e.g., remove or track objects together with their associated effects) and generate highly realistic videos, without relying on traditional intermediate representations like optical flow or depth maps. By integrating a selective content encoder and leveraging an

I2V generation model, GenProp consistently preserves unchanged content while dynamically propagating the changes. Synthetic data and the region-aware loss further enhance its ability to disentangle and refine edits across frames. Experimental results demonstrate its effectiveness, establishing it as a robust, flexible solution that surpasses prior methods in scope and precision. In the future, we plan to extend the model to take in more than one key frame edits and uncover additional video tasks that can be supported.

References

[1] https://www.pexels.com/. 7

[2] https://www.pika.art/. 2, 3, 7

[3] https://stock.adobe.com/. 7

[4] Omer Bar-Tal, Dolev Ofri-Amar, Rafail Fridman, Yoni Kasten, and Tali Dekel. Text2live: Text-driven layered image and video editing. In *European conference on computer vision*, 2022. 3

[5] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 2, 3, 5

[6] Duygu Ceylan, Chun-Hao P Huang, and Niloy J Mitra. Pix2video: Video editing using image diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. 2, 3

[7] Haoxin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter2: Overcoming data limitations for high-quality video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 2

[8] Jiaxin Cheng, Tianjun Xiao, and Tong He. Consistent video-to-video transfer using synthetic dataset. *arXiv preprint arXiv:2311.00213*, 2023. 3, 7

[9] Yuren Cong, Mengmeng Xu, Christian Simon, Shoufa Chen, Jiawei Ren, Yanping Xie, Juan-Manuel Perez-Rua, Bodo Rosenhahn, Tao Xiang, and Sen He. Flatten: optical flow-guided attention for consistent text-to-video editing. *arXiv preprint arXiv:2310.05922*, 2023. 2, 3

[10] Mean Squared Error. Mean squared error. *MA: Springer US*, 2010. 5

[11] Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. 3

[12] Ruoyu Feng, Wenming Weng, Yanhui Wang, Yuhui Yuan, Jianmin Bao, Chong Luo, Zhibo Chen, and Baining Guo. Ccedit: Creative and controllable video editing via diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 3

[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. 3

[14] Yuchao Gu, Yipin Zhou, Bichen Wu, Licheng Yu, Jia-Wei Liu, Rui Zhao, Jay Zhangjie Wu, David Junhao Zhang, Mike Zheng Shou, and Kevin Tang. Videoswap: Customized video subject swapping with interactive semantic point correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 3

[15] Yuwei Guo, Ceyuan Yang, Anyi Rao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Sparsectrl: Adding sparse controls to text-to-video diffusion models. In *European Conference on Computer Vision*, 2025. 3

[16] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022. 2

[17] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv preprint arXiv:2205.15868*, 2022. 2

[18] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 3

[19] Jiahui Huang, Leonid Sigal, Kwang Moo Yi, Oliver Wang, and Joon-Young Lee. Inve: Interactive neural video editing. *arXiv preprint arXiv:2307.07663*, 2023. 2

[20] Varun Jampani, Raghudeep Gadde, and Peter V Gehler. Video propagation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017. 3

[21] Hyeonho Jeong, Geon Yeong Park, and Jong Chul Ye. Vmc: Video motion customization using temporal attention adaption for text-to-video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 3

[22] Ozgur Kara, Bariscan Kurtkaya, Hidir Yesiltepe, James M Rehg, and Pinar Yanardag. Rave: Randomized noise shuffling for fast and consistent video editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 3

[23] Yoni Kasten, Dolev Ofri, Oliver Wang, and Tali Dekel. Layered neural atlases for consistent video editing. *ACM Transactions on Graphics (TOG)*, 2021. 2

[24] Lei Ke, Yu-Wing Tai, and Chi-Keung Tang. Occlusion-aware video object inpainting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 3

[25] Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Deep video inpainting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019. 3

[26] Max Ku, Cong Wei, Weiming Ren, Huan Yang, and Wenhu Chen. Anyv2v: A plug-and-play framework for any video-to-video editing tasks. *arXiv preprint arXiv:2403.14468*, 2024. 3, 7

[27] Xirui Li, Chao Ma, Xiaokang Yang, and Ming-Hsuan Yang. Vidtome: Video token merging for zero-shot video editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 2

[28] Feng Liang, Bichen Wu, Jialiang Wang, Licheng Yu, Kunpeng Li, Yinan Zhao, Ishan Misra, Jia-Bin Huang, Peizhao Zhang, Peter Vajda, et al. Flowvid: Taming imperfect optical flows for consistent video-to-video synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 3

[29] Shaoteng Liu, Yuechen Zhang, Wenbo Li, Zhe Lin, and Jiaya Jia. Video-p2p: Video editing with cross-attention control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 3

[30] Chong Mou, Mingdeng Cao, Xintao Wang, Zhaoyang Zhang, Ying Shan, and Jian Zhang. ReVideo: Remake a Video with Motion and Content Control. *arXiv preprint arXiv:2405.13865*, 2024. 2, 3, 7, 8

[31] OpenAI. Sora: Creating video from text. https://openai.com/index/sora/, 2024. 2, 3, 5

[32] Hao Ouyang, Qiuyu Wang, Yuxi Xiao, Qingyan Bai, Juntao Zhang, Kecheng Zheng, Xiaowei Zhou, Qifeng Chen, and Yujun Shen. Codef: Content deformation fields for temporally consistent video processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 2, 3

[33] Wenqi Ouyang, Yi Dong, Lei Yang, Jianlou Si, and Xingang Pan. I2VEdit: First-Frame-Guided Video Editing via Image-to-Video Diffusion Models. *arXiv preprint arXiv:2405.16537*, 2024. 2, 3, 8

[34] Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv:2410.13720*, 2024. 2, 3

[35] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 7

[36] Chenyang Qi, Xiaodong Cun, Yong Zhang, Chenyang Lei, Xintao Wang, Ying Shan, and Qifeng Chen. Fatezero: Fusing attentions for zero-shot text-based video editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. 3

[37] 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*, 2021. 8

[38] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment Anything in Images and Videos. *arXiv preprint arXiv:2408.00714*, 2024. 3, 5, 7

[39] Xiaoyu Shi, Zhaoyang Huang, Fu-Yun Wang, Weikang Bian, Dasong Li, Yi Zhang, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, et al. Motion-i2v: Consistent and controllable image-to-video generation with explicit motion modeling. In *ACM SIGGRAPH 2024 Conference Papers*, 2024. 3

[40] Chaehun Shin, Heeseung Kim, Che Hyun Lee, Sang-gil Lee, and Sungroh Yoon. Edit-a-video: Single video editing with object-aware consistency. In *Asian Conference on Machine Learning*, 2024. 3

[41] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022. 2

[42] Uriel Singer, Amit Zohar, Yuval Kirstain, Shelly Sheynin, Adam Polyak, Devi Parikh, and Yaniv Taigman. Video editing via factorized diffusion distillation. In *European Conference on Computer Vision*, 2025. 2, 3

[43] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, 2020. 2, 3

[44] Alexandru Telea. An image inpainting technique based on the fast marching method. *Journal of graphics tools*, 9(1): 23–34, 2004. 1

[45] Amazon Mechanical Turk. Amazon mechanical turk. *Retrieved August*, 2012. 8

[46] Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable length video generation from open domain textual descriptions. In *International Conference on Learning Representations*, 2022. 2

[47] Wen Wang, Yan Jiang, Kangyang Xie, Zide Liu, Hao Chen, Yue Cao, Xinlong Wang, and Chunhua Shen. Zero-shot video editing using off-the-shelf image diffusion models. *arXiv preprint arXiv:2303.17599*, 2023. 3

[48] Xiang Wang, Hangjie Yuan, Shiwei Zhang, Dayou Chen, Jiuniu Wang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingren Zhou. Videocomposer: Compositional video synthesis with motion controllability. *Advances in Neural Information Processing Systems*, 2024. 3

[49] Bichen Wu, Ching-Yao Chuang, Xiaoyan Wang, Yichen Jia, Kapil Krishnakumar, Tong Xiao, Feng Liang, Licheng Yu, and Peter Vajda. Fairy: Fast parallelized instruction-guided video-to-video synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 3

[50] 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*, 2023. 3

[51] Jay Zhangjie Wu, Xiuyu Li, Difei Gao, Zhen Dong, Jinbin Bai, Aishani Singh, Xiaoyu Xiang, Youzeng Li, Zuwei Huang, Yuanxi Sun, Rui He, Feng Hu, Junhua Hu, Hai Huang, Hanyu Zhu, Xu Cheng, Jie Tang, Mike Zheng Shou, Kurt Keutzer, and Forrest Iandola. CVPR 2023 Text Guided Video Editing Competition, 2023. 7, 8, 9

[52] Jinbo Xing, Menghan Xia, Yuxin Liu, Yuechen Zhang, Yong Zhang, Yingqing He, Hanyuan Liu, Haoxin Chen, Xiaodong Cun, Xintao Wang, et al. Make-your-video: Customized video generation using textual and structural guidance. *IEEE Transactions on Visualization and Computer Graphics*, 2024. 3

[53] Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Wangbo Yu, Hanyuan Liu, Gongye Liu, Xintao Wang, Ying Shan, and Tien-Tsin Wong. Dynamicrafter: Animating open-domain images with video diffusion priors. In *European Conference on Computer Vision*, 2025. 2

[54] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. Youtube-vos:

A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*, 2018. 5

[55] Hanshu Yan, Jun Hao Liew, Long Mai, Shanchuan Lin, and Jiashi Feng. Magicprop: Diffusion-based video editing via motion-aware appearance propagation. *arXiv preprint arXiv:2309.00908*, 2023. 2, 3

[56] Wilson Yan, Andrew Brown, Pieter Abbeel, Rohit Girdhar, and Samaneh Azadi. Motion-conditioned image animation for video editing. *arXiv preprint arXiv:2311.18827*, 2023. 3

[57] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. CogVideoX: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024. 2

[58] Danah Yatim, Rafail Fridman, Omer Bar-Tal, Yoni Kasten, and Tali Dekel. Space-time diffusion features for zero-shot text-driven motion transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 3

[59] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 4

[60] Shiwei Zhang, Jiayu Wang, Yingya Zhang, Kang Zhao, Hangjie Yuan, Zhiwu Qin, Xiang Wang, Deli Zhao, and Jingren Zhou. I2vgen-xl: High-quality image-to-video synthesis via cascaded diffusion models. *arXiv preprint arXiv:2311.04145*, 2023. 2, 3

[61] Zicheng Zhang, Bonan Li, Xuecheng Nie, Congying Han, Tiande Guo, and Luoqi Liu. Towards consistent video editing with text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 2024. 2

[62] Min Zhao, Rongzhen Wang, Fan Bao, Chongxuan Li, and Jun Zhu. Controlvideo: Adding conditional control for one shot text-to-video editing. *arXiv preprint arXiv:2305.17098*, 2023. 3

[63] Rui Zhao, Yuchao Gu, Jay Zhangjie Wu, David Junhao Zhang, Jia-Wei Liu, Weijia Wu, Jussi Keppo, and Mike Zheng Shou. Motiondirector: Motion customization of text-to-video diffusion models. In *European Conference on Computer Vision*, 2025. 3

[64] Shangchen Zhou, Chongyi Li, Kelvin CK Chan, and Chen Change Loy. Propainter: Improving propagation and transformer for video inpainting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. 7

[65] Bojia Zi, Shihao Zhao, Xianbiao Qi, Jianan Wang, Yukai Shi, Qianyu Chen, Bin Liang, Kam-Fai Wong, and Lei Zhang. CoCoCo: Improving Text-Guided Video Inpainting for Better Consistency, Controllability and Compatibility. *arXiv preprint arXiv:2403.12035*, 2024. 2

Supplementary Material

S1. Synthetic Data Generation

Our model (GenProp) is trained on synthetic data derived from video instance segmentation datasets. The synthetic data pairs are generated using a combination of methods: (1) Copy-and-Paste for object removal, (2) Mask-and-Fill for editing, and (3) Color-Fill techniques for tracking. These methods ensure diverse training scenarios while maintaining control over the generated content.

S1.1. Copy-and-Paste

To generate synthetic training data, we employ a copy-and-paste strategy in the dataloader. For each iteration, two videos $V_1 = (v_{1,1}, \ldots, v_{1,n})$ and V_2 are sampled. We check whether V_2 contains an instance mask in the first frame, as our model modifies the video based on the first frame. If neither video has an instance mask in the first frame, the sample is skipped.

Otherwise, the augmented video V_{aug} is created as:

$$V_{\text{aug}} = (1 - \mathrm{M}_2) \odot V_1 + \mathrm{M}_2 \odot V_2, \tag{8}$$

where M_2 represents the instance mask of V_2, and $\odot$ denotes element-wise multiplication. This operation pastes the object from V_2 onto V_1.

As illustrated in Fig. 8 (a), rows 1–6, this approach is simple and efficient, enabling rapid generation of large-scale synthetic data. However, it does not explicitly address harmonization between the pasted object and the target video. The size, position, and motion trajectory of the pasted object vary.

S1.2. Mask-and-Fill

For the Mask-and-Fill strategy, a single video $V = (v_1, \ldots, v_n)$ is sampled at each iteration. Similar to the copy-and-paste strategy, we ensure that the first frame contains an instance mask. If no mask is present in the first frame, the sample is skipped. To fill the instance mask, we employ two approaches:

Surrounding Background Mean Fill This method fills masked regions using the mean pixel value of a rectangular area surrounding the mask, as shown in Fig. 8 (b), rows 1–2. For each frame, the bounding box of the mask is identified and expanded by a margin of 5 pixels. The mean pixel value of the unmasked region within this area is then computed and used to replace the masked region. This approach is simple and efficient, providing a quick solution for local content replacement or insertion.

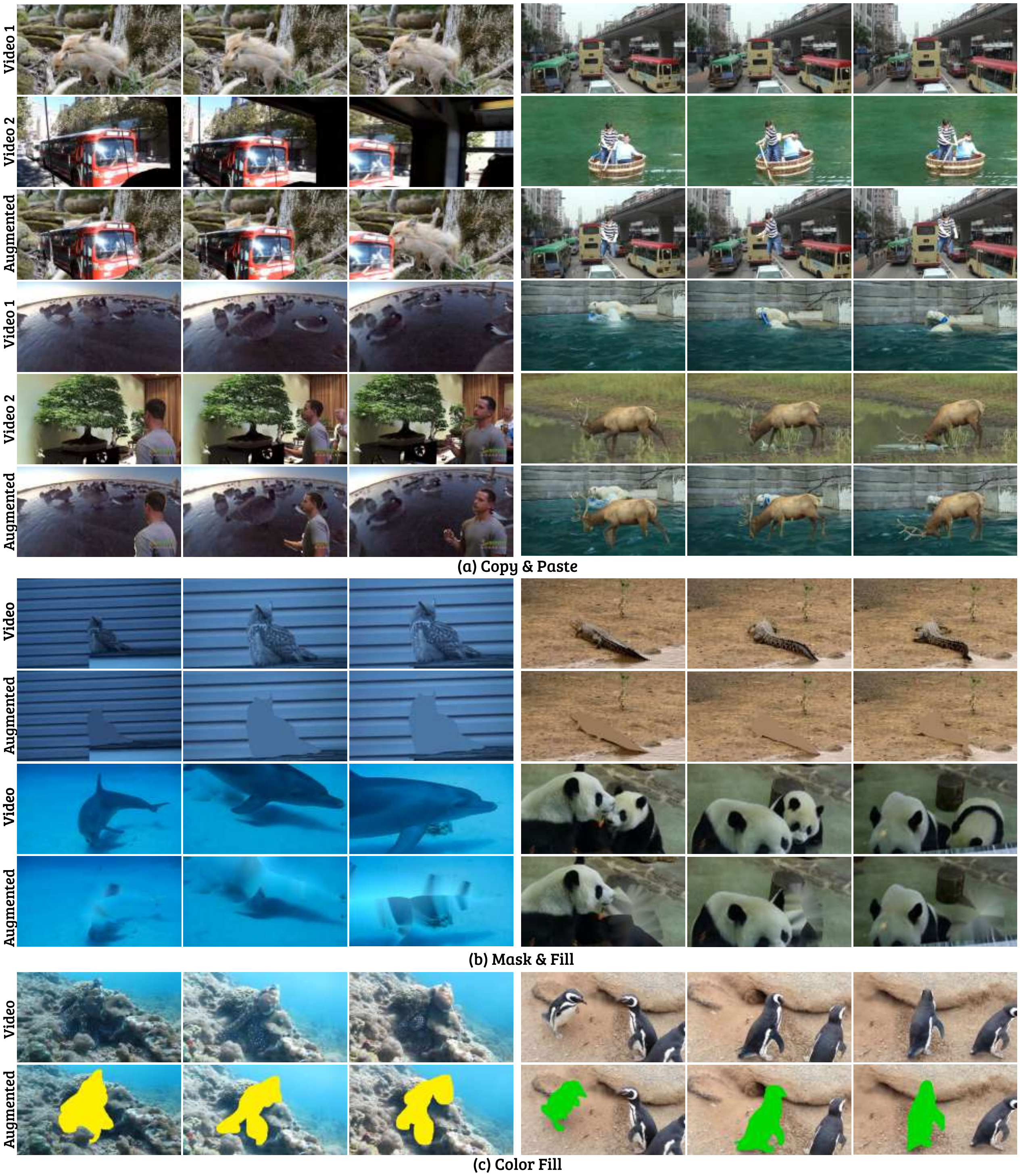
OpenCV-Based Inpainting As shown in Fig. 8 (b), rows 3–4, this method utilizes OpenCV's `cv2.inpaint()` function with the `INPAINT_TELEA` algorithm. The algorithm [44] reconstructs the masked regions by interpolating from the surrounding pixels.

Both methods are lightweight and designed for real-time data generation, allowing synthetic data to be processed online during training. Surrounding Background Mean Fill prioritizes simplicity and speed, while OpenCV-Based Inpainting offers more sophisticated results at a slightly higher computational cost. The ratio between the two methods is approximately 2:1.

S1.3. Color-Fill

In this method, the segmentation masks are used to directly fill occluded regions with a predefined color. The default color is red (R=1.0, G=0.0, B=0.0), but a random color is sampled from a predefined palette, including green, blue, yellow, purple, and cyan. Specifically, given a binary segmentation mask, regions marked with "1" are replaced with the randomly selected color, while regions marked with "0" are preserved from the original frame. In 30% of the cases, a second color is randomly sampled for another instance, promoting the model's ability to track multiple instances.

The procedure is straightforward yet effective, as it introduces strong visual cues that highlight the areas where propagation tasks occur. As illustrated in Fig. 8 (c), this method is particularly useful for training tasks that require tracking or editing specific regions, as the distinct colors ensure clear differentiation of object instances across frames.



(a) Copy & Paste

(b) Mask & Fill

(c) Color Fill

Figure 8. Synthetic Data Generation. We use different ways to generate our training data by simulating a task: (a) Copy-and-Paste for object removal; (b) Mask-and-Fill for editing and insertion; (c) Color Fill for both tracking and editing enhancement.

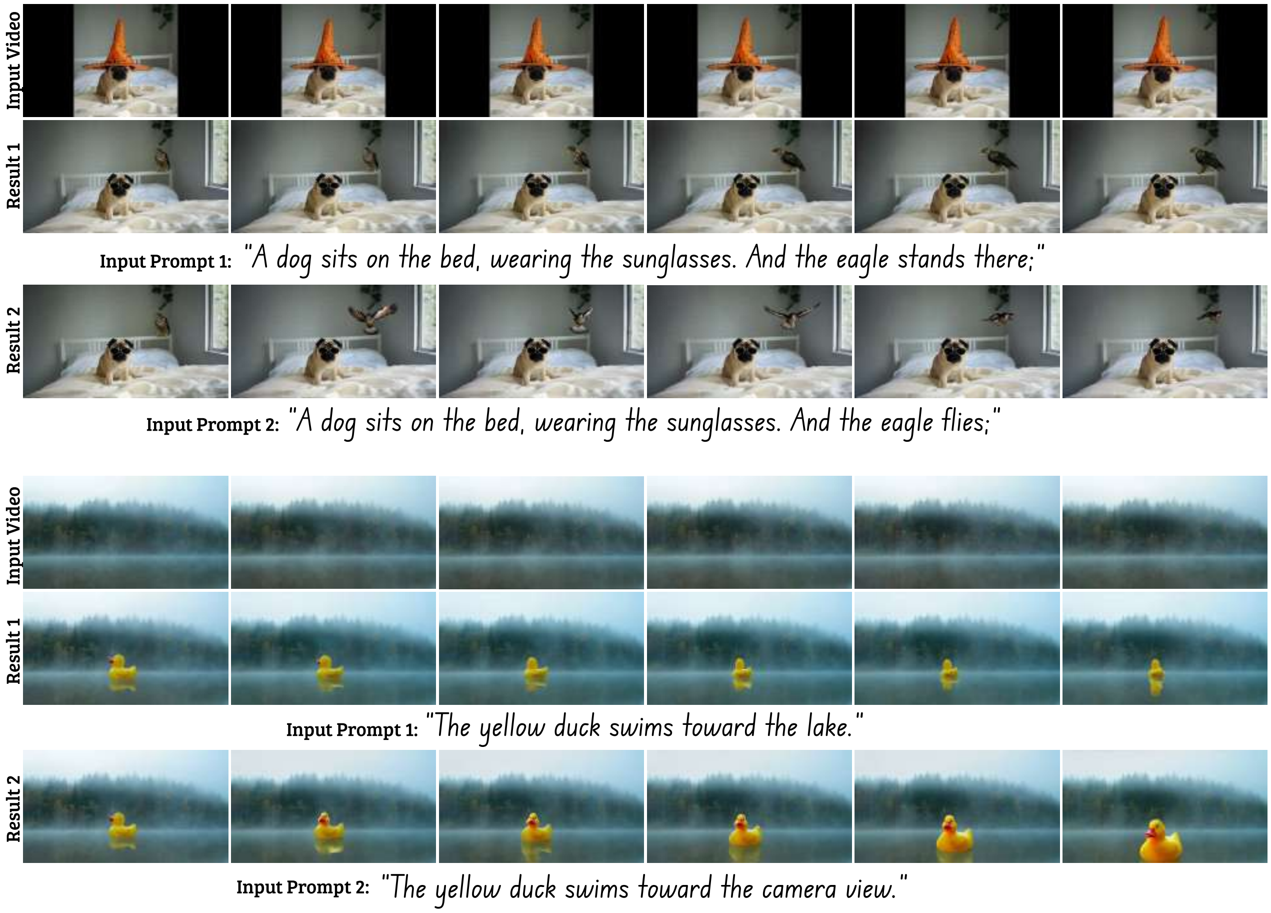

Input Video
Result 1
Input Prompt 1: "A dog sits on the bed, wearing the sunglasses. And the eagle stands there;"
Result 2
Input Prompt 2: "A dog sits on the bed, wearing the sunglasses. And the eagle flies;"
Input Video
Result 1
Input Prompt 1: "The yellow duck swims toward the lake."
Result 2
Input Prompt 2: "The yellow duck swims toward the camera view."

Figure 9. Text Control Analysis. Text prompts can be used to control the result in a desired way.

S2. Controls for Generation

S2.1. Text Control Analysis

In GenProp, the text prompt also plays a role in guiding the model to generate content that aligns with the desired outcome. The interaction between the edited first frame and the input video, combined with the provided text prompt, results in different outputs, demonstrating the potential influence of text control on video propagation.

In Fig. 9 rows 1-3, we illustrate a scenario involving multiple edits, including object removal and editing. In this example, an eagle is inserted into the video, and the text prompt is used to control the eagle's behavior—whether it "stands" or "flies". The text prompt directs how the eagle is depicted and how it moves within the video.

In Fig. 9 rows 4-6, we show a video of a lake surface with mist, where a small yellow duck is inserted in the first frame. By varying the text prompt, the direction in which the duck swims can be controlled. Different text prompts guide the duck's movement, demonstrating the model's ability to follow text cues for spatial and motion control, adding an extra layer of flexibility for dynamic video editing tasks.

These examples underscore the capacity of GenProp to integrate textual instructions effectively, allowing for nuanced and adaptable control over the generated video content, making it a powerful tool for both creative video editing and dynamic scene manipulation.

S2.2. Injection Weight Analysis

As shown in Fig. 10, the injection layer connects the output of the Select Content Encoder (SCE) to the Image-to-Video Model, enabling the selective propagation of content between the original video and the generated edits. To control the balance between preserving the original video and generating the edited content, we introduce an injection weight parameter, ranging

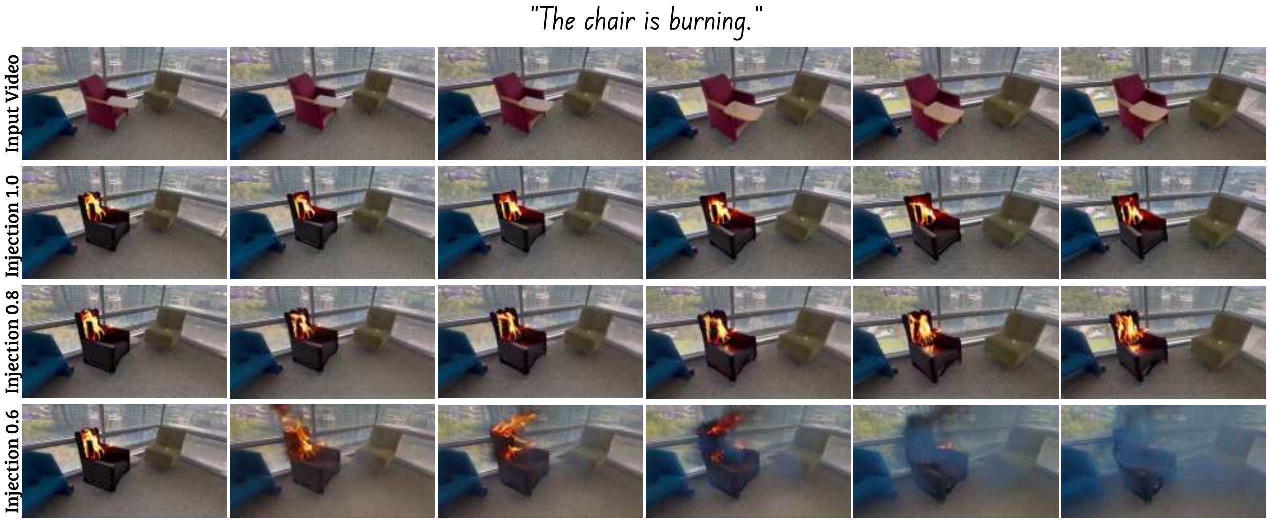


Figure 10. Injection Weight Analysis. The injection weight serves as a way to control the trade-off between reconstruction ability and generation ability. With a lower injection weight, edits with significant changes can appear in the original video as shown in the last row.

from [0.0, 1.0], multiplied by the injection layer, which can be adjusted during the inference phase. This injection weight serves as a trade-off, allowing for more control over how much of the original video is reconstructed versus how much of the newly generated content is introduced.

For instance, as shown in Fig. 10, we use a video of a sofa and edit the first frame to make it appear as if it is burning. When the injection weight is set to 1.0, the reconstruction of the original video is highly accurate, but the flame effects are relatively small. As the injection weight is decreased to 0.8, the flames become more pronounced while still maintaining a strong reconstruction of the original content. At an injection weight of 0.6, the reconstruction of the ground and windows is somewhat weakened, but the generated smoke from the flames can spread over a much larger area, showcasing how the injection weight directly influences the extent to which the model prioritizes either reconstruction or generation of new content.

S2.3. Black Region in Input Video

In the standard GenProp setting, the Selective Content Encoder (SCE) takes the original video as input. The SCE's task is to distinguish between modified and unmodified content. Adding appropriate masks to the input video can help the SCE focus on this task and improve the model's overall performance. We also found that using moving masks in the input video can guide the motion of the modified content. This provides a certain level of control over the motion of the edited regions.

Fig. 11 demonstrates that adding a black region to the input video can help control the motion of the element we want to edit. Specifically, in the first case, we can use the moving black blocks in the input video to simulate the effect of a car being overtaken. In the second case, the black region helps the model to use text to control the motion of the lemon.

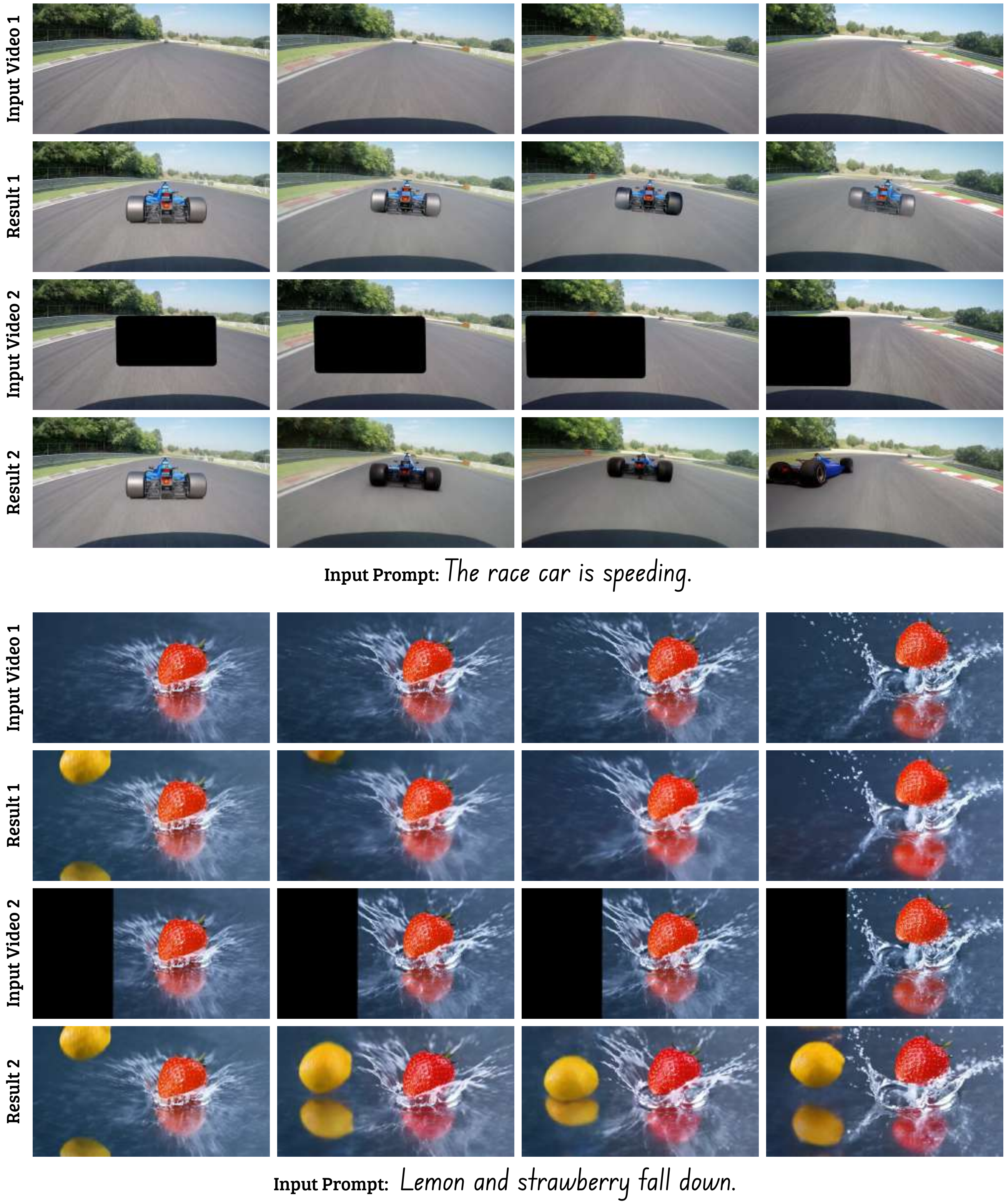
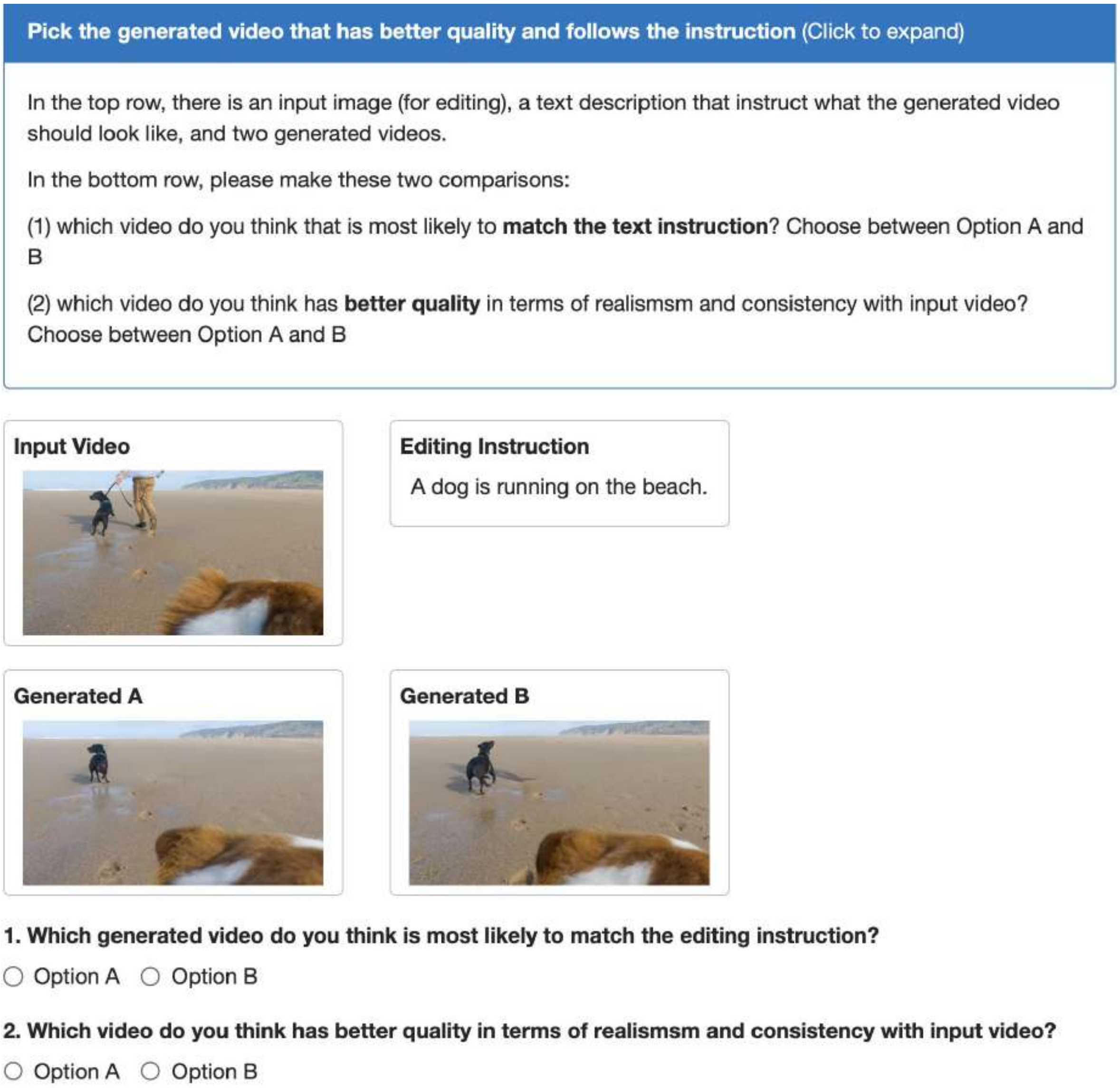

Input Video 1
Result 1
Input Video 2
Result 2
Input Prompt: The race car is speeding.
Input Video 1
Result 1
Input Video 2
Result 2
Input Prompt: Lemon and strawberry fall down.

Figure 11. Motion Control with Black Regions. Adding back regions to the input video can help to control the motion of the element we want to edit in the video. For example, we can simulate overtaking of a racecar (top) or make the lemon fall to the left of the strawberry (bottom).



Pick the generated video that has better quality and follows the instruction (Click to expand)

In the top row, there is an input image (for editing), a text description that instruct what the generated video should look like, and two generated videos.

In the bottom row, please make these two comparisons:

(1) which video do you think that is most likely to **match the text instruction**? Choose between Option A and B

(2) which video do you think has **better quality** in terms of realismsm and consistency with input video? Choose between Option A and B

Input Video

Editing Instruction

A dog is running on the beach.

Generated A

Generated B

1. Which generated video do you think is most likely to match the editing instruction?

○ Option A ○ Option B

2. Which video do you think has better quality in terms of realismsm and consistency with input video?

○ Option A ○ Option B

Figure 12. User Study Interface. Screenshot of a user study screen where two questions are asked to the user for assessing (1) alignment to the text and (2) overall video quality.

S3. User Study Details

Fig. 12 shows the interface used in our user study. In this study, users are presented with an input video, a corresponding text prompt, and the results generated by both our GenProp model and a random baseline (with users unaware of which result corresponds to which model). The users are asked to evaluate the outputs based on two criteria: "alignment to the editing goal" and "output video quality". Specific questions related to these criteria are detailed in the figure. At the end of the study, participants' responses are collected in a CSV format. To ensure the reliability of the results, we perform a systematic filtering of user responses, excluding those from participants who exhibited unreasonable response times (less than 1 second), ensuring that the data reflects thoughtful and accurate assessments. This user study setup allows us to compare the performance of GenProp against a baseline and gain insights into the effectiveness of our model in real-world editing tasks.

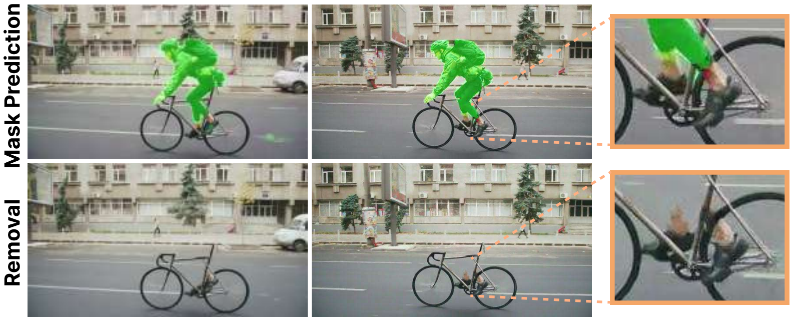


Figure 13. Observation. When the mask prediction fails, the editing may fail in a similar manner.

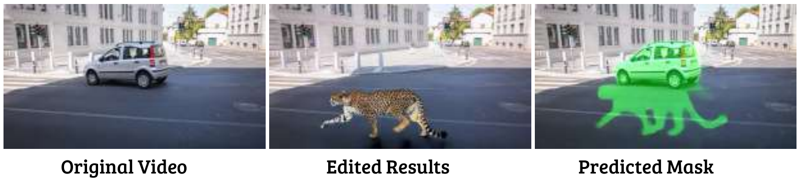


Figure 14. Mask Visualization. Mask prediction decoder can estimate the edited region, even when its shape extends beyond the original object.

S4. Mask Prediction Analysis

For the Mask Prediction Decoder (MPD), we make additional observations. As shown in Fig. 13, the editing outcomes and the mask prediction results often succeed or fail in the same way. This correlation highlights the importance of accurate mask predictions for generating high-quality edits. As further illustrated in Fig. 14, MPD is not only capable of predicting the object that is removed from the original video (which it is trained to) but can also estimate its effect (shadow) and the future appearance areas of inserted objects. This ability to anticipate the placement of new elements ensures that edits are seamlessly integrated with the existing video content, leading to more natural and consistent results.

S5. More Results

More comparison results are shown in Fig. 15 (removal), Fig. 16 (TGVE [51]), and Fig. 17 (Challenging Test Set). We further provide video results as part of the Supplementary Material. Please refer to the folders `1-Showcase` for various video results of our model and `2-Comparison` for video comparisons to existing work. HTML file provided inside each folder will visualize an HTML gallery with all video clips. Additionally, a demo video `demo.mp4` is provided for reference.

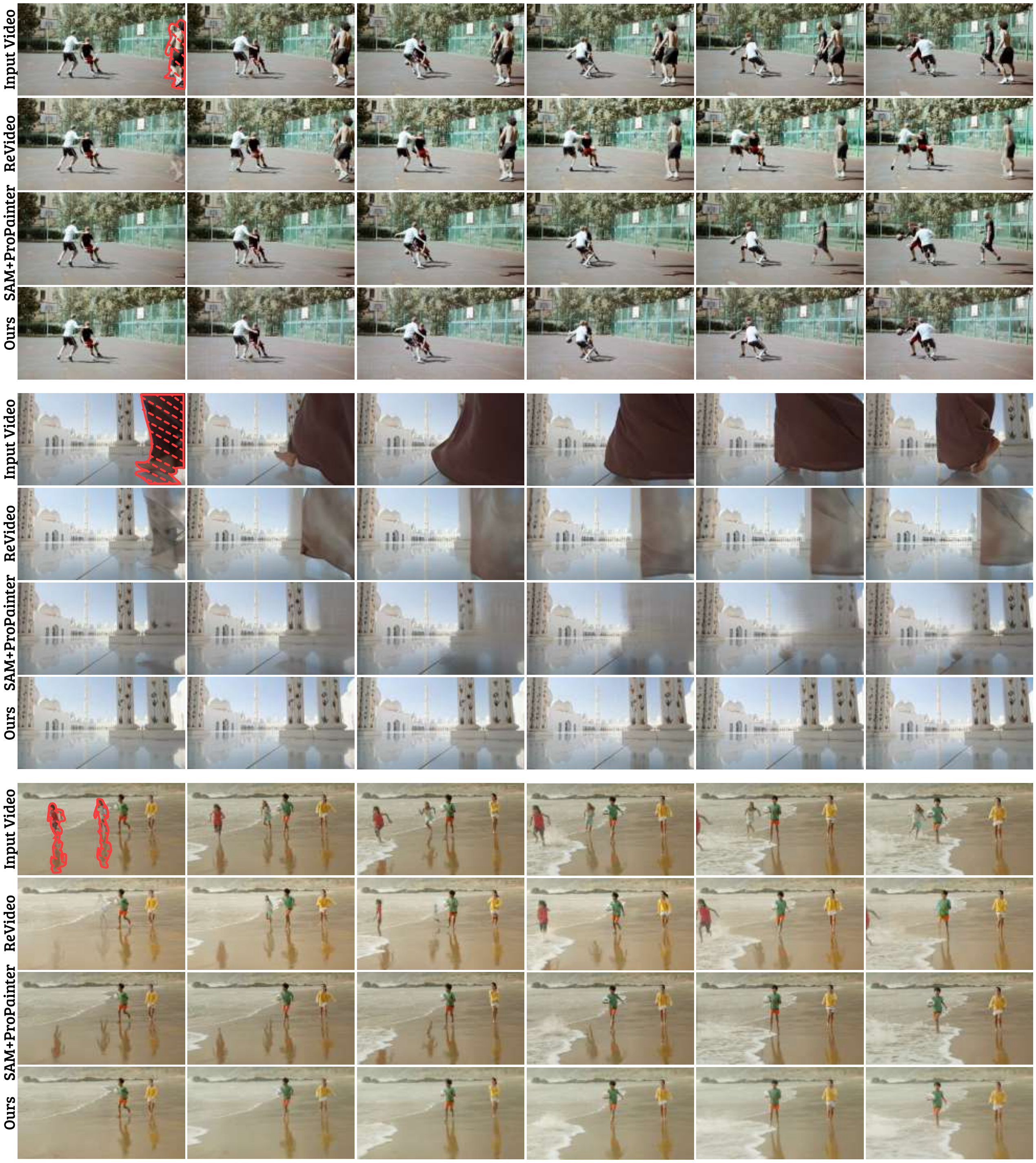


Figure 15. Additional Comparison for Removal. Our model is able to consistently remove the object and its effect (e.g., shadow, reflection) together in the whole video.

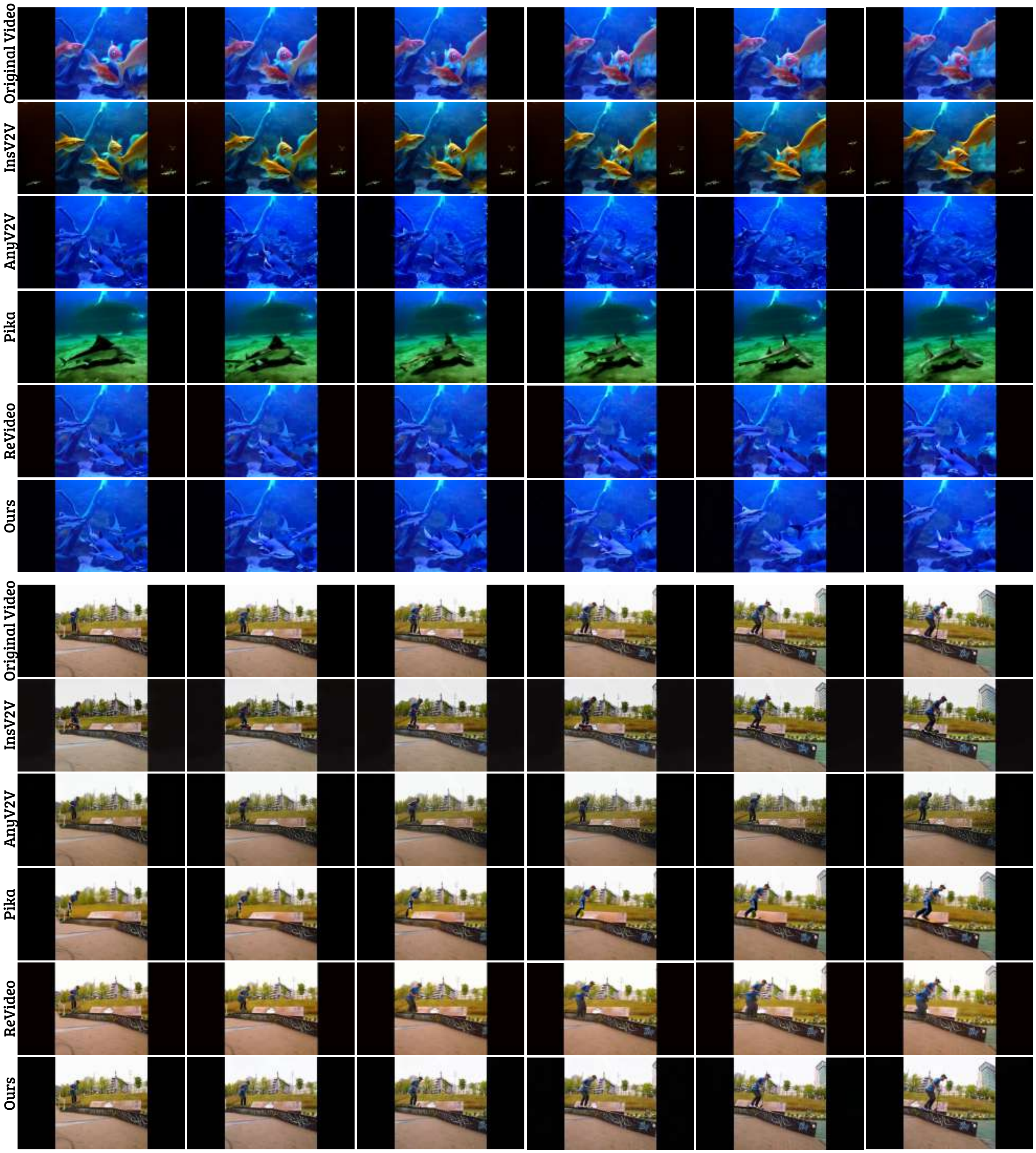


Figure 16. Additional Comparison for Editing on TGVE [51]. We provide additional comparisons on the TGVE dataset [51]. The first frame shown in Ours is the edited frame. As shown, our model is able to propagate the desired edit throughout the video.

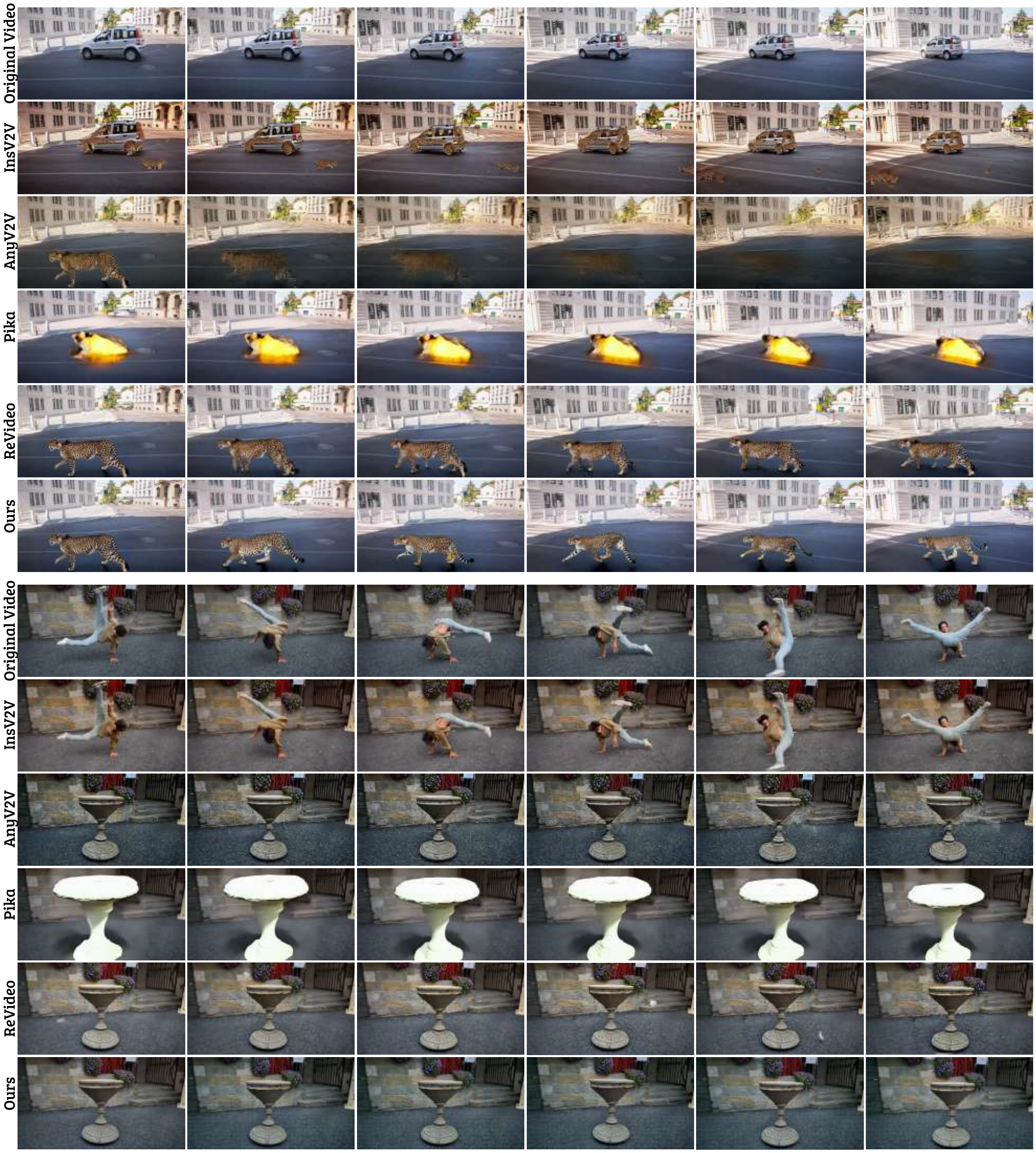


Figure 17. Additional Comparison for Editing on the Challenging Test Set. We provide additional comparisons on the Challenging Test Set. The first frame shown in Ours is the edited frame. Our model is able to replace existing objects and generate independent motion for inserted objects over the video frames.

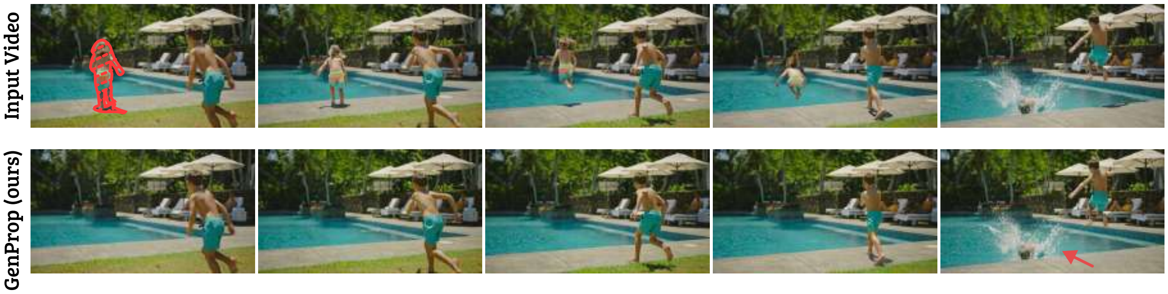


Figure 18. Limitation. It is still challenging to remove the events caused by the object, e.g., the splash of water is not removed when the girl jumping into the pool is removed.

S6. Limitations

As shown in Fig. 18, while GenProp demonstrates the ability to handle side effects such as shadows and reflections during tasks like removal and tracking, higher-level effects caused by objects or events remain challenging to edit. For example, the splash of water generated when the girl jumps into the pool (Fig. 18) cannot be directly modified or controlled within the current framework. This limitation presents an interesting direction for future research.