

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. Synthetic data is less likely to cause artifacts as it is only fed into SCE, with all supervisory (ground truth) data still being real videos.

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 $V_1 = (v_{1,1}, \dots, v_{1,n})$ 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:

$$V_{\text{aug}} = (1 - M_2) \odot V_1 + M_2 \odot V_2, \quad (8)$$

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 $V = (v_1, \dots, v_n)$ 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 [45] 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.

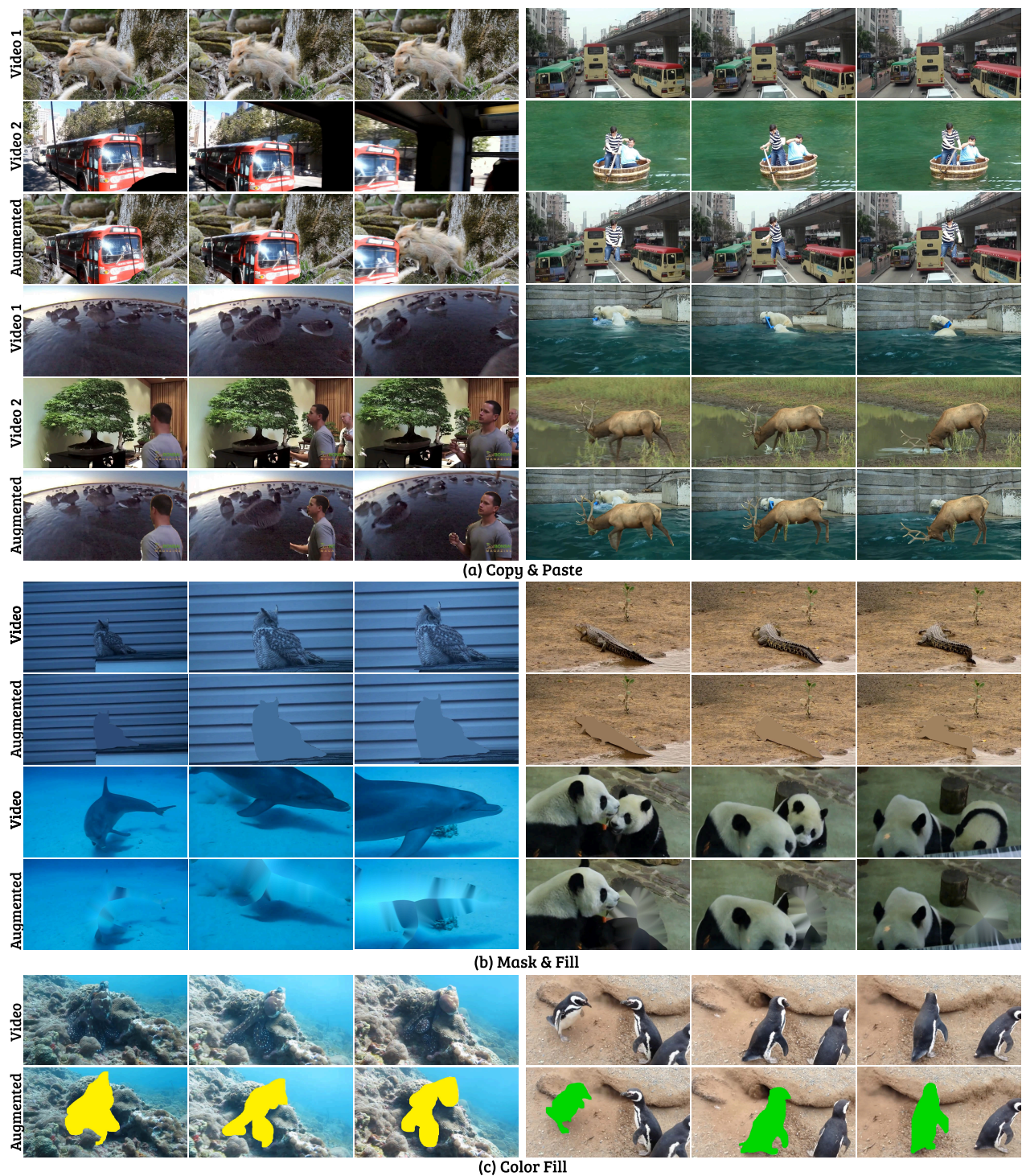


Figure 8. Synthetic Data Generation. We use different ways to generate our training data by simulating a task: (a) Copy-and-Paste for object removal; (b) Mask-and-Fill for editing and insertion; (c) Color Fill for both tracking and editing enhancement.

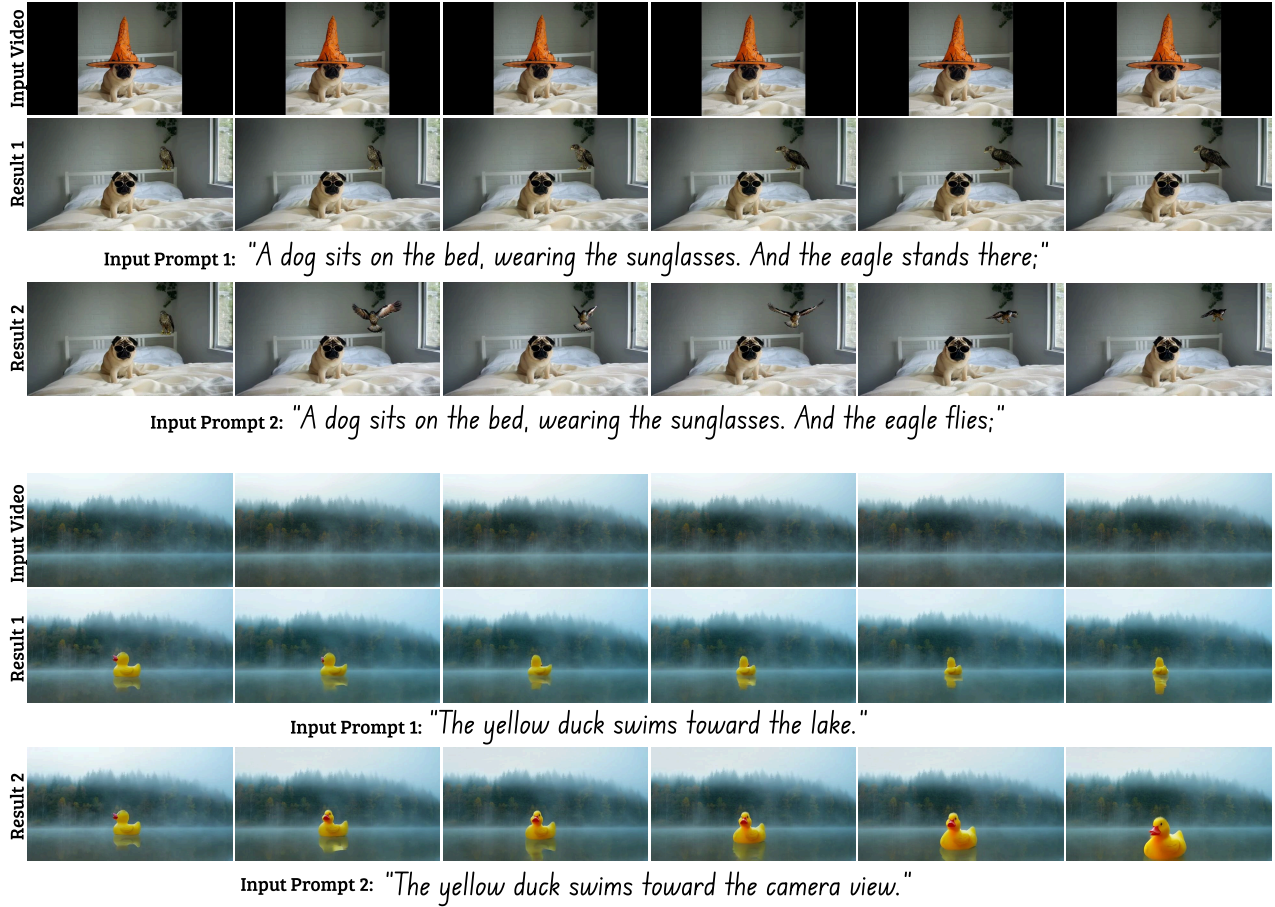


Figure 9. Text Control Analysis. Text prompts can be used to control the result in a desired way.

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,

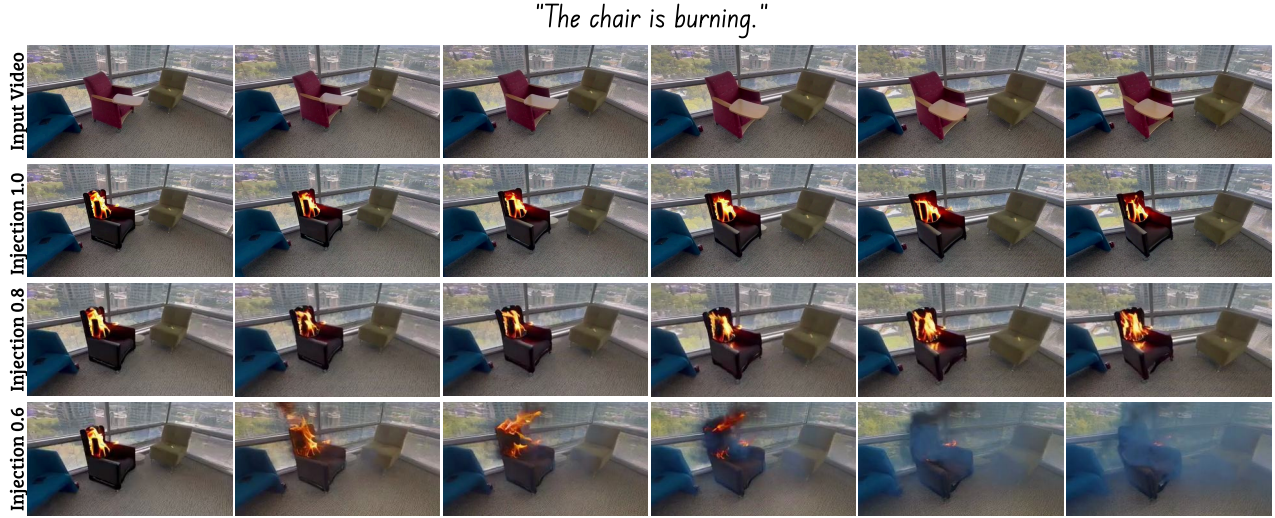


Figure 10. Injection Weight Analysis. The injection weight serves as a way to control the trade-off between reconstruction ability and generation ability. With a lower injection weight, edits with significant changes can appear in the original video as shown in the last row.

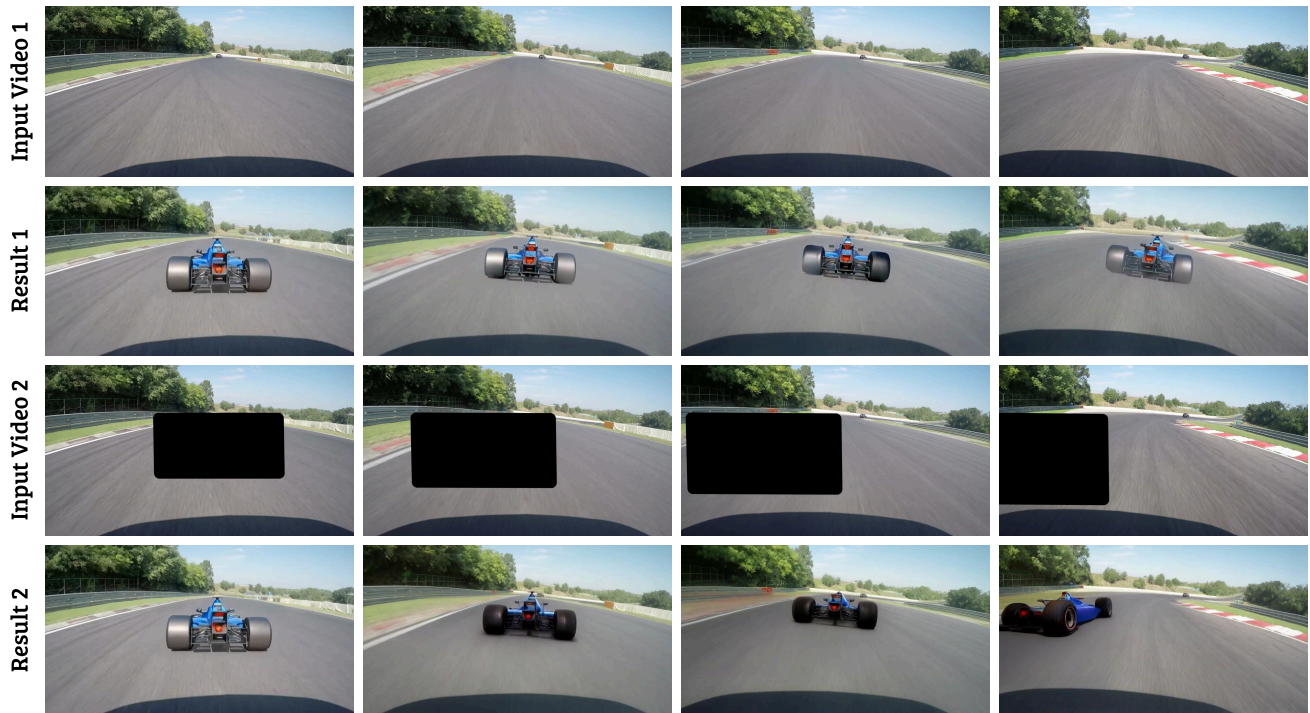
ranging from $[0.0, 1.0]$, multiplied by the injection layer, which can be adjusted during the inference phase. This injection weight serves as a trade-off, allowing for more control over how much of the original video is reconstructed versus how much of the newly generated content is introduced.

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.

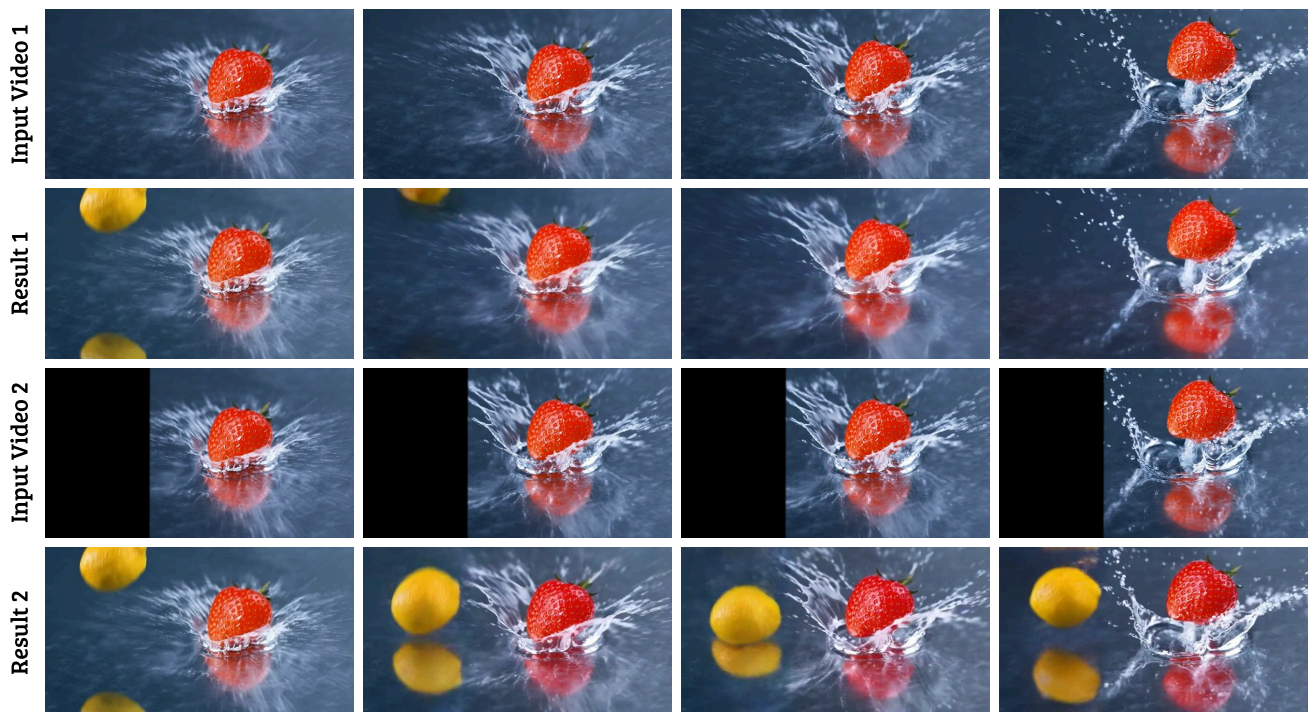
S2.3. Black Region in Input Video

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 Prompt: *The race car is speeding.*



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)

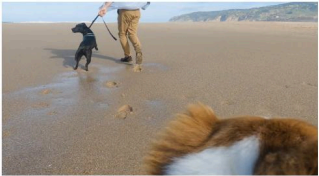
In the top row, there is an input image (for editing), a text description that instruct what the generated video should look like, and two generated videos.

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 realism 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 realism 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.

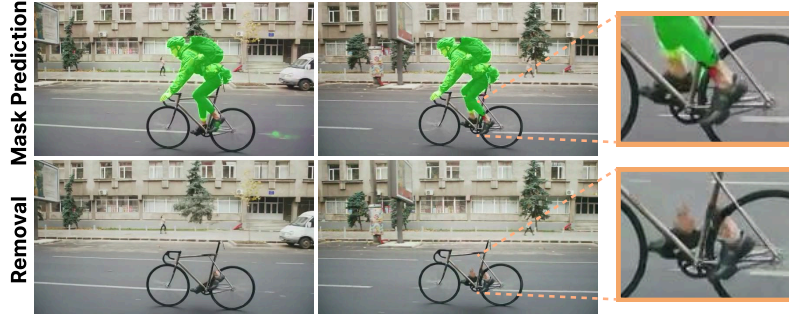


Figure 13. Observation. When the mask prediction fails, the editing may fail in a similar manner.

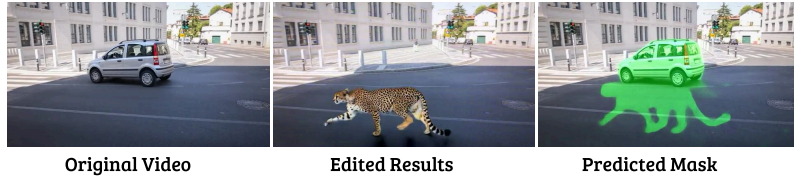


Figure 14. Mask Visualization. Mask prediction decoder can estimate the edited region, even when its shape extends beyond the original object.

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. MPD is part of the diffusion model pipeline, but one denoising step suffices for estimating the mask.

S5. Additional Details

Synthetic data types are encoded as numbers and processed by a 2-layer MLP into task embeddings for removal, insertion, and tracking. Our model is trained for $\sim 10K$ steps.

S6. More Experiment Results

In Tab. 4, we provide an additional quantitative comparison to VideoShop [12] which allows first-frame editing. We observe that VideoShop is fast and effective for object editing/insertion but less successful in object removal and background replacement. In Tab. 5, we provide comparisons on an additional metric, CLIP-I-Fore, which measures foreground consistency. GenProp can effectively model faces and non-rigid motions as shown in Tab. 5 and Fig. 15. An ablation study on the gradient loss $\mathcal{L}_{\text{grad}}$ using the SVD architecture is provided in Tab. 6, showing its benefits.

More comparison results are shown in Fig. 16 (removal), Fig. 17 (TGVE [52]), and Fig. 18 (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.

VideoShop	PSNR $_m$ \uparrow , CLIP-T \uparrow , CLIP-I \uparrow	28.191, 0.3044, 0.9662
ReVideo	PSNR $_m$ \uparrow , CLIP-T \uparrow , CLIP-I \uparrow	31.589, 0.3134, 0.9757
GenProp (Ours)	PSNR $_m$ \uparrow , CLIP-T \uparrow , CLIP-I \uparrow	33.513, 0.3217, 0.9814

Table 4. Quantitative comparison to VideoShop.

Metric	InsV2V	AnyV2V	Pika	ReVideo	GenProp
CLIP-I-Fore \uparrow	0.9693	0.9617	0.9629	0.9765	0.9818

Table 5. Experiment results on CLIP-I-Fore, a foreground consistency metric. GenProp outperforms other models on CLIP-I-Fore, showing good foreground alignment.

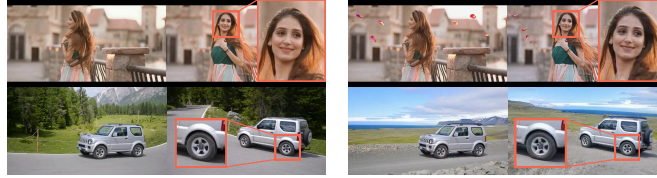


Figure 15. Results on faces and non-rigid motions.

CLIP-T \uparrow , CLIP-I \uparrow	$\mathcal{L}_{\text{mask}}$ (0.3301, 0.9834)	$\mathcal{L}_{\text{mask}} + \mathcal{L}_{\text{grad}}$ (0.3313, 0.9866)
---------------------------------------	--	---

Table 6. Ablation study on the gradient loss.



Figure 16. 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.

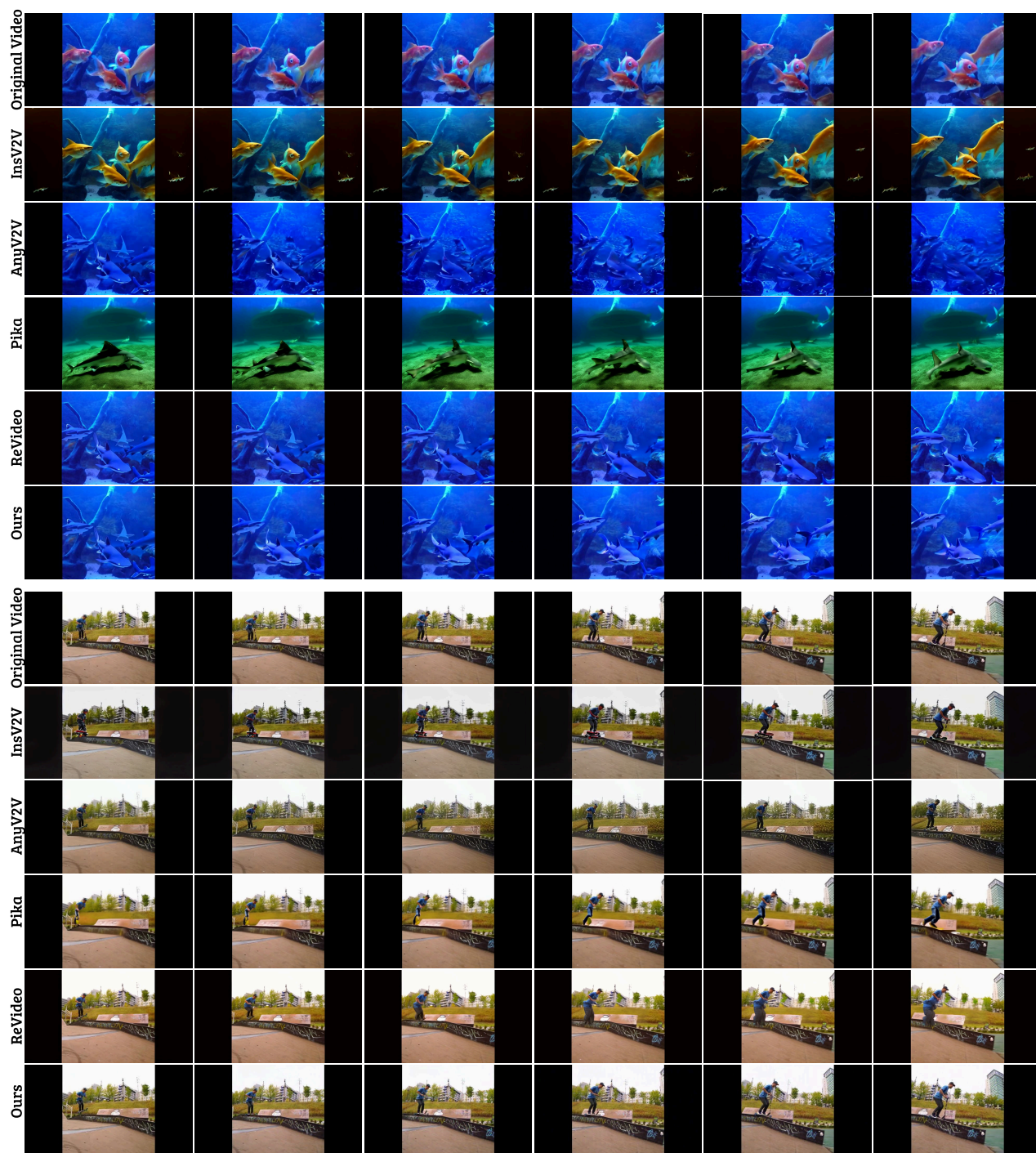


Figure 17. Additional Comparison for Editing on TGVE [52]. We provide additional comparisons on the TGVE dataset [52]. The first frame shown in Ours is the edited frame. As shown, our model is able to propagate the desired edit throughout the video.



Figure 18. 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 19. 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.

S7. Limitations

As shown in Fig. 19, 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. 19) cannot be directly modified or controlled within the current framework. This limitation presents an interesting direction for future research.

GenProp is not designed for global edits but shows potential for tasks like motion blur, as shown in Fig. 20. Further global editing applications (e.g., focus, FOV) are open for exploration.



Figure 20. An example of global edit propagation. GenProp is able to propagate motion blur across the following frames.