ROSE: Remove Objects with Side Effects in Videos

Chenxuan Miao[1,4], **Yutong Feng**[2,*] **Jianshu Zeng**[3], **Zixiang Gao**[3], **Hantang Liu**[2],
Yunfeng Yan[1], **Donglian Qi**[1], **Xi Chen**[4], **Bin Wang**[2], **Hengshuang Zhao**[4†]
[1]Zhejiang University, [2]KunByte AI, [3]Peking University, [4]The University of Hong Kong
{weiyuchoumou526, fengyutong.fyt, zengjianshu.AI, gzx2401210062 }@gmail.com
{liuhantang77, chauncey0620, binwang393}@gmail.com
{yyff, qidl}@zju.edu.cn hszhao@cs.hku.hk

Abstract

Video object removal has achieved advanced performance due to the recent success of video generative models. However, when addressing the side effects of objects, *e.g.,* their shadows and reflections, existing works struggle to eliminate these effects for the scarcity of paired video data as supervision. This paper presents ROSE, termed **R**emove **O**bjects with **S**ide **E**ffects, a framework that systematically studies the object's effects on environment, which can be categorized into five common cases: shadows, reflections, light, translucency and mirror. Given the challenges of curating paired videos exhibiting the aforementioned effects, we leverage a 3D rendering engine for synthetic data generation. We carefully construct a fully-automatic pipeline for data preparation, which simulates a large-scale paired dataset with diverse scenes, objects, shooting angles, and camera trajectories. ROSE is implemented as an video inpainting model built on diffusion transformer. To localize all object-correlated areas, the entire video is fed into the model for reference-based erasing. Moreover, additional supervision is introduced to explicitly predict the areas affected by side effects, which can be revealed through the differential mask between the paired videos. To fully investigate the model performance on various side effect removal, we presents a new benchmark, dubbed ROSE-Bench, incorporating both common scenarios and the five special side effects for comprehensive evaluation. Experimental results demonstrate that ROSE achieves superior performance compared to existing video object erasing models and generalizes well to real-world video scenarios. The project page is `https://rose2025-inpaint.github.io`.

1 Introduction

Removing objects in visual contents represents a valuable technique with widespread applications in both daily and industrial scenarios. This task targets to re-fill the masked region of objects via reasonable and consistent content, regarding the context in surrounding environment. Prior works [15, 38, 14, 3] towards either image or video object removal have explored to leverage flow-based pixel propagation to restore the masked region with neighboring information [44], or adopt the inpainting paradigm to directly generate the masked content [33, 23]. Powered by the significant capability of large-scale models [27, 8, 19, 23] on generalized visual creation, the inpainting-based methods exhibit satisfying erasing performance in diverse scenarios of image and video.

Despite the advanced performance, however, existing works are still restricted due to the lack of paired training samples that follows real-world physical rules. The paired samples represents data

[*] Project Leader
[†] Corresponding Author

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

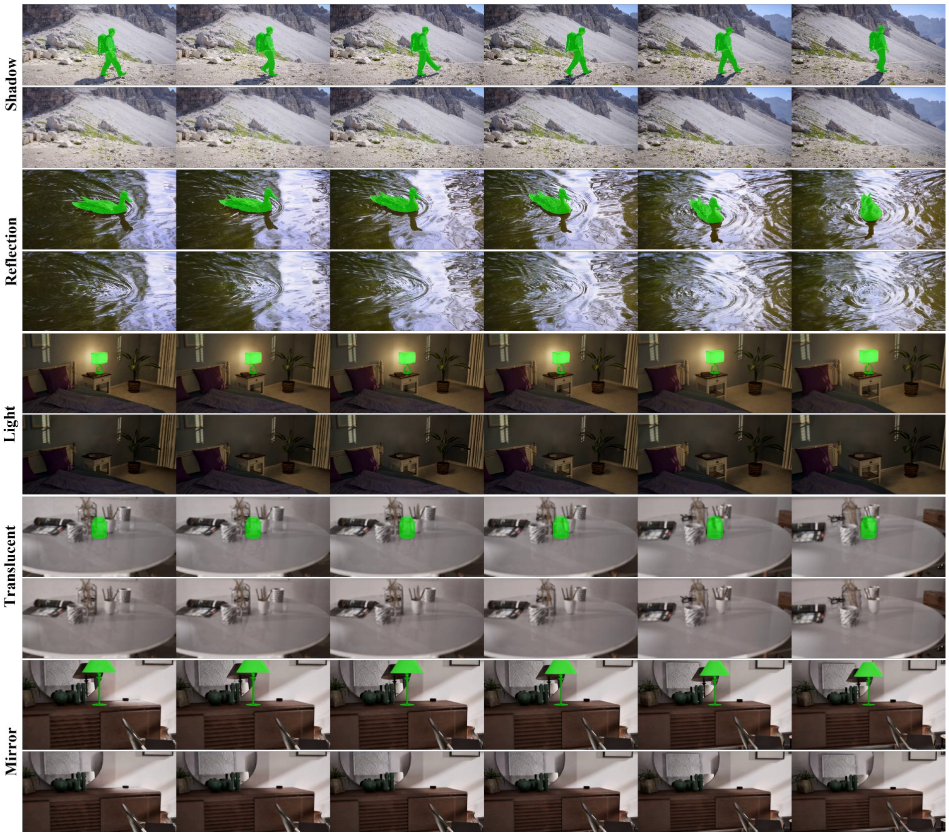

Figure 1: **Video object removal results generated by** ROSE (zoom in for better view). Every two lines are an example where the above is input video with mask and the bottom is inference result. We sequentially show cases of various side effects studied in this paper.

with and without the object, where the object's influence on the environment is correspondingly changed, such as its shadow on the ground. Most works leverage the segmentation dataset, *e.g.,* DAVIS [26] and YouTube-VOS [40], to construct artificial pairs, either directly pasting an object from another sample, or masking the object with zero value. While simple and scalable, these strategies fail to reflect the side effects of object, *e.g.,* shadows, reflections and lighting changes. Therefore, models supervised by the artificial pairs typically generate unnatural outputs with side effects left in environment. To tackle this, OmniEraser [38] manages to filter out such image pairs from the sequential frames in videos with static camera motion. However, when confronting with video object removal, it is impractical to leverage higher dimensional data to construct the paired dataset.

To address these problems, we propose to prepare the paired video samples via 3D rendering. Recent advancements on the rendering engines [7] make it practical to generate high-qualified and strictly-aligned synthetic video pairs. We design a fully-automatic data preparation pipeline to create a large-scale video set for object removal. More concretely, we collect a batch of base environments and split them into multiple scenes containing various objects. Following that, the pipeline automatically generates cameras focusing on the objects to be removed, and apply random camera trajectories. The rendering engine enables us to activate or disable the object, and also precisely render the object masks. Thus, we could obtain a list of triples consisting of the original video, edited video with object removed, and the corresponding mask video, which contain perfectly simultaneous temporal contents. Furthermore, we systematically study the various types of side effects in videos, including light source, mirror, reflection, shadow, and translucency. Equipped with the data preparation pipeline, we efficiently construct a comprehensive dataset including all the above side effects on diverse scenes.

To fully utilize the synthetic data, we present ROSE, an efficient framework based on video inpainting to remove object in videos with their side effects. To help distinguish the object-interacted region in environment, we directly feed the whole video into the model, in contrast to previous works that fills the object area with zero mask. The complete video serves as a powerful reference guidance on model, to localize the side effects concerning the intrinsic attributes of the object. We also apply random augmentation strategies on the mask to cope with various input in inference. Furthermore, we introduce an additional supervision to explicitly predict the difference mask between edited and original videos. We implement this by injecting a mask predictor based on the hidden representations of the inpainting model. The aforementioned architectures of `ROSE` are observed to enhance the model's capability to attend and erase the side effects in videos.

To facilitate a comprehensive evaluation on the object removal results with side effects, we construct a new benchmark, named ROSE-Bench, consisting of both realistic and synthetic video data. Through extensive experiments, we demonstrate that `ROSE` achieves state-of-the-art performance on video object removal, and effectively adapts to real-world scenarios.

2 Related Work

Diffusion Transformers for Video Generation. Recent diffusion models [10, 28, 31, 32] have shown strong performance in text-to-video generation. By integrating transformers [34], diffusion transformers (DiTs)[25] improve video quality and temporal consistency. State-of-the-art methods leverage large-scale video-text datasets[2, 41] and hybrid architectures for efficiency and fidelity. Recent DiT-based latent diffusion models, such as Wan2.1 [35] and MAGI-1 [1], excel in long video generation: Wan2.1 uses causal 3D VAEs with 1:256 compression and flow matching for real-time synthesis, while MAGI-1 employs an autoregressive DiT for chunk-wise generation with strict causality. These advances underscore DiTs' strength in balancing quality, efficiency, and control.

Video Inpainting. Early video inpainting methods primarily used 3D CNNs [5, 36, 12] to model spatial-temporal features, but their limited receptive fields hindered long-range propagation. Subsequently, optical flow [16, 22, 46] and homography [21, 4] were introduced to guide pixel propagation. To improve efficiency and accuracy, Zhou et al.[44] proposed ProPainter, combining optical flow and attention mechanisms. Recently, with the rise of diffusion models[10, 28], diffusion-based video inpainting has emerged [30, 20, 39, 45, 29]. Li et al.[23] proposed DiffuEraser, extending the image inpainting model BrushNet[15] to videos via a two-stage training scheme.

3 Dataset Construction

3.1 Paired Erasing Videos Preparation using 3D data

Acquiring paired data samples that depict scenes with and without objects and their side effects represents a significant challenge in object removal task. Though recent work explores generating such image pairs from videos with static camera motion [38], it is impossible to obtain video pairs in a higher dimension using this technique. To tackle this problem, we propose to utilize the adequate 3D data together with advanced game engine, *i.e.,* the Unreal Engine [7], to synthesize the paired video data. As illustrated in Fig. 2, we present an automatic data preparation pipeline as follows:

Scene and Object Sampling. We begin by collecting large-scale virtual environments from public 3D asset platforms such as Fab [6]. Each environment is sufficiently complex and diverse, covering a wide range of indoor and outdoor scenes, including urban settings, natural landscapes, and artificial constructions. We manually subdivide these base environments into smaller scenes, each containing one or more candidate objects for removal. In total, we collect 28 high-quality environments and split them into 450 unique scenes. The selected scenes include a wide variety of object types—both static and dynamic—including vehicles, animals, plants, and more. This ensures a diverse training corpus that enhances the generalization ability of the inpainting model.

Multi-view Generation with Object Masks. Given a sampled object in a scene, we randomly assign multiple camera views with varying angles and distances within predefined ranges. A key advantage of using a 3D engine is the ability to generate accurate object masks via programmable post-process shaders, avoiding reliance on segmentation models [17]. For each object, we apply a custom shader that renders the object in white and masks the rest in black, producing precise binary masks. Per-frame mask videos are automatically generated through scripting.

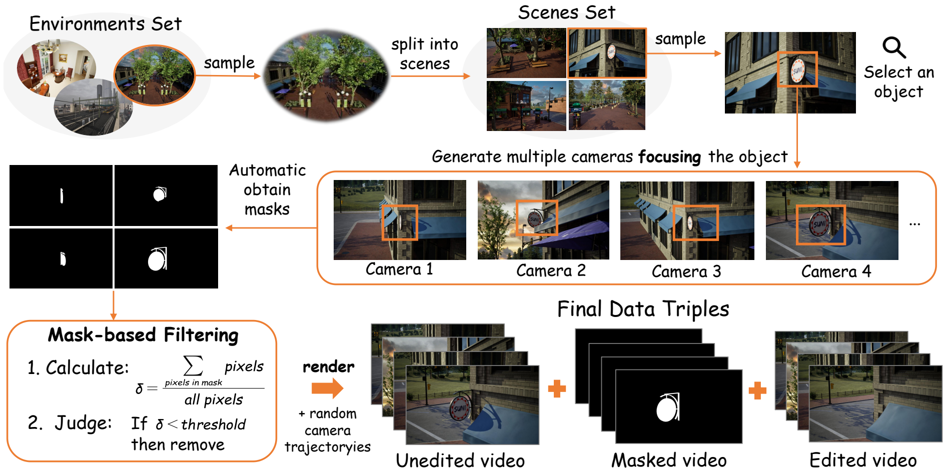
Environments Set
sample
split into scenes
Scenes Set
sample
Select an object
Generate multiple cameras **focusing** the object
Automatic obtain masks
Camera 1
Camera 2
Camera 3
Camera 4
...
Mask-based Filtering
1. Calculate: $\delta = \frac{\sum_{\text{pixels in mask}} \text{pixels}}{\text{all pixels}}$
2. Judge: If $\delta <$ threshold then remove
render
+ random camera trajectoryies
Final Data Triples
Unedited video
Masked video
Edited video

Figure 2: **Paired video preparation pipeline using 3D data**, which can be divided into: scene and object sampling, multi-view generation with masks, valid view filtering and video data rendering.

Valid View Filtering. To ensure the quality of videos and avoid object-occlusion cases, we further filter out views by calculating the ratio of foreground pixels in the mask. Ratios lower than a threshold suggest videos with insufficient mask coverage, *e.g.,* due to occlusion or mislabeling. Such videos are discarded to avoid introducing noisy supervision into the training set.

Video Pair Rendering. After filtering, we render both the unedited (original) and edited (object-removed) video sequences by toggling the visibility of the selected objects in the engine. The camera moving is sampled from a pre-defined set with random disturbing, *e.g.,* zooming in and out. All video pairs are rendered at a resolution of 1920×1080 and a frame length of 90 frames (6 seconds). Since the camera trajectories and object placement are determined via scripted generation, the original video, the corresponding mask video, and the edited video remain spatially and temporally aligned on a per-frame basis. Such an alignment is critical for enabling pixel-wise supervised learning.

3.2 Categorize Side Effect in Videos

To improve the generalization ability of the model and its robustness under various complex real-world conditions, we deliberately construct the dataset composed of six distinct categories. These categories are carefully designed to simulate typical yet challenging side effects that commonly occur in practical scenarios, such as object-light interactions, mirror reflections, and translucent materials. By explicitly injecting such variations into the training process, we aim to equip the model with the capacity to understand and handle diverse object-environment relationships beyond trivial inpainting cases. We summarize the definition of side effects on the environment as follows:

Common: Objects with minimal interaction with surrounding context, representing typical inpainting cases. Their removal causes little disruption to spatial layout or visual semantics.

Light Source: This category includes objects that function as light emitters. Their removal changes the global illumination, affecting shadows, reflections, and overall scene appearance.

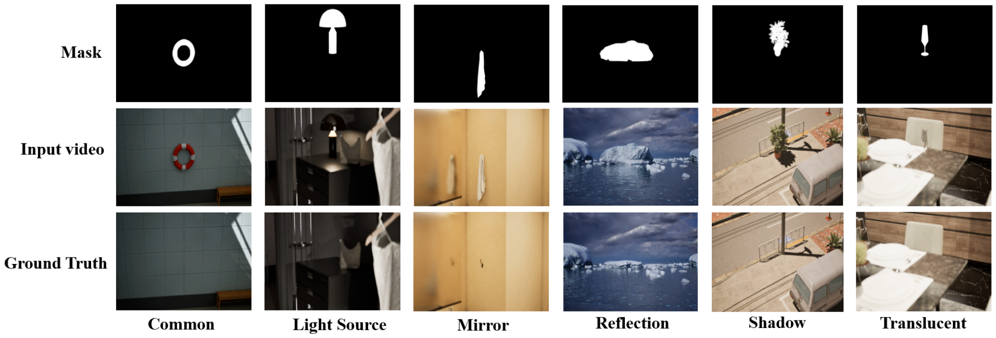

Figure 3: Illustration of the various **side-effect categories** studied in the dataset of ROSE.

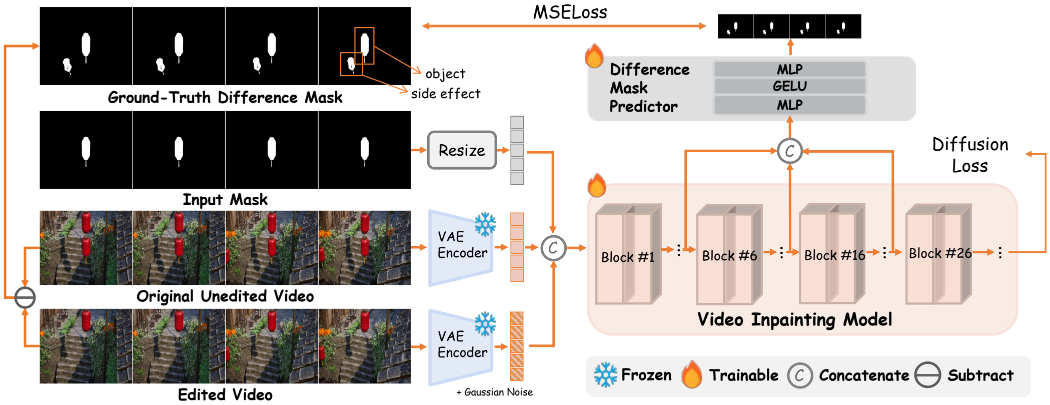

Figure 4: **The framework of** ROSE. We concatenate the noisy latents with the original input video and masks, consumed by a video inpainting model. An additional difference mask predictor is introduced to predict the correlated area in video, automatically computed from the input video pairs.

Mirror: Objects reflected in mirrors require spatial reasoning and semantic understanding to inpaint both the object and its mirrored counterpart, ensuring visual consistency.

Reflection: Compared to the Mirror category, it emphasizes reflective surfaces like water, requiring the model to infer and complete indirect visual cues from reflections.

Shadow: Shadows linked to objects require joint removal, making inpainting sensitive to lighting and spatial structure to ensure coherence across both object and shadow regions.

Translucent: Semi-transparent objects expose the background with blending or refraction. Inpainting must recover both visible cues and hidden structures for realistic restoration.

4 Method

4.1 Overview

In this section, we elaborate the model architecture of ROSE for conducting video object removal, as illustrated in Fig. 4. In brief, ROSE is implemented as an inpainting model continued from the foundation video generative models [35, 18] (the Wan2.1 model [35] in this paper). Following the general architecture in diffusion-based inpainting models [19, 3], we extend the model input with the original input video together with object masks. Distinguished from the typical setting that multiply the mask onto input video, we directly feed the whole video to assist the understanding on environment. The input masks, with precise boundary generated by 3D engine, are further augmented to enhance model robustness. To better supervise the model to localize the subtle object-environment interactions, we introduce an additional difference mask predictor to explicitly predict the side effect areas. We present the detail of ROSE in the following sub-sections.

4.2 Reference-based Object Erasing

We start by formulate the video inpainting task in ROSE. Given an input video $\mathcal{V}$ and a binary mask sequence $\mathcal{M}$, where the area of object to be removed is filled with value 1, the target is to generate an object-erased video $\hat{\mathcal{V}}$. For the video condition consumed by the model, most prior methods [23, 5, 44, 36] follow a "mask-and-inpaint" paradigm, feeding the network with only the non-object area $\mathcal{V} \odot (1 - \mathcal{M})$, where $\odot$ indicates point-wise product. Suppose the noisy latents of diffusion model as X, then the model input can be regarded as $[X; \mathcal{V} \odot (1 - \mathcal{M}); \mathcal{M}]$. Such a manner explicitly eliminate the object from input, and is friendly for model convergence. However, when confronting the side effect removal, isolating the object from the model makes it challenging to localize the object-related region. In contrast, recent work on image modality has explored to guide the model with the masked region for reference [14]. In this paper, we adapt such reference-based erasing, modifying the model input as $[X; \mathcal{V}; \mathcal{M}]$. Experimental results suggest introducing the whole video as guidance significantly increase the performance. We attribute the advancement that the inner

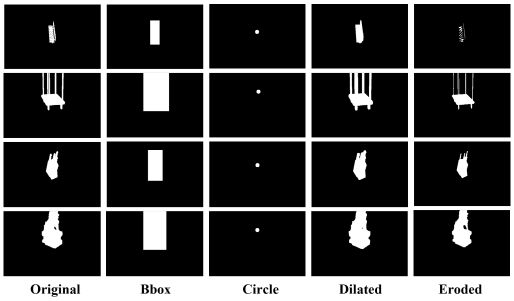

Figure 5: Visualization of various mask augmentation strategies adopted in training.

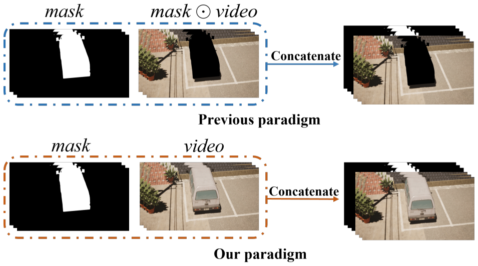

Figure 6: Comparison between the previous paradigm and our reference-based paradigm.

attention mechanism is effective for seeking the inter-region correlations in videos. Given the object region as input, the model thereby leverage it prior knowledge to localize the side effect regions, thus outperforms the model with masked video input. Furthermore, the complete video as input serves to enhance the temporal consistency of output video, for introducing the original object-environment interactions. The visualization comparisons between the two paradigms are shown in Fig. 6.

4.3 Mask Augmentation

In real-world applications, user-provided masks often vary in precision, size, and shape—ranging from accurate segmentation maps to coarse bounding boxes or sparse point annotations. Since the masks generated by 3D engine is perfectly accurate, training solely on such ideal masks can lead to a performance gap at deployment. To mitigate this, we introduce a set of mask augmentation strategies that simulate diverse mask types likely to appear in practice. As shown in Fig. 5, we adopt five variants: (i) *Original mask*, a precise binary map from ground-truth annotations; (ii) *Point-wise mask*, an extremely sparse point simulating minimal user input; (iii) *Bounding box mask*, a coarse rectangular region enclosing the target; (iv) *Dilated mask*, obtained via morphological dilation to simulate loose annotations; and (v) *Eroded mask*, generated by erosion to mimic under-segmentation. These variants are randomly sampled during training, which exposes the model to diverse, imperfect masks and improves its generalization to real-world inputs.

4.4 Explicit Supervision via Difference Mask Prediction

Beyond the diffusion loss targeting reconstruct the regions of object and its side effect, we introduce an additional supervision into ROSE. Specifically, we inject a *difference mask predictor* into the framework, predicting binary masks indicating all the areas to be modified in video.

The core idea is to leverage the complementary information in the video pairs for training. When an object is removed from a scene, it often leaves behind subtle but semantically significant side effects, such as shadows, reflections, and occlusions. To explicitly guide the model in attending to these regions, we compute a binary difference mask by comparing the original video $\mathbf{x}_0 \in \mathbb{R}^{c\times f\times h\times w}$ and its edited counterpart $\tilde{\mathbf{x}}_0$. The difference mask $\mathbf{d}_0 \in \{0,1\}^{f\times h\times w}$ is defined as:

$$\mathbf{d}_0^{(t,h,w)} = \begin{cases} 1, & \text{if } \left\| \mathbf{x}_0^{(t,h,w)} - \tilde{\mathbf{x}}_0^{(t,h,w)} \right\|_2 > \delta \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where $\delta > 0$ is a fixed threshold ($\delta = 0.09$ in this paper). The resulting binary mask highlights pixel-level differences induced by object removal and is downsampled to match the latent resolution, yielding the ground-truth difference mask $\mathbf{d}_t \in \{0,1\}^{f\times h/s\times w/s}$.

Difference Mask Predictor. To guide the model in identifying regions influenced by object removal, we design a difference mask predictor $\mathcal{D}_\theta$, which takes as concatenated token features as input, extracted from multiple transformer blocks. Let $\mathbf{x} \in \mathbb{R}^{B\times L\times D_{\text{total}}}$ denote the fused feature sequence, where $L = F_p \times H_p \times W_p$ represents the total number of tokens and D_{total} is the aggregated channel dimension after selecting and concatenating multiple transformer layers. The difference mask predictor consists of a two-layer MLP that reduces D_{total} to a scalar prediction per token. Its

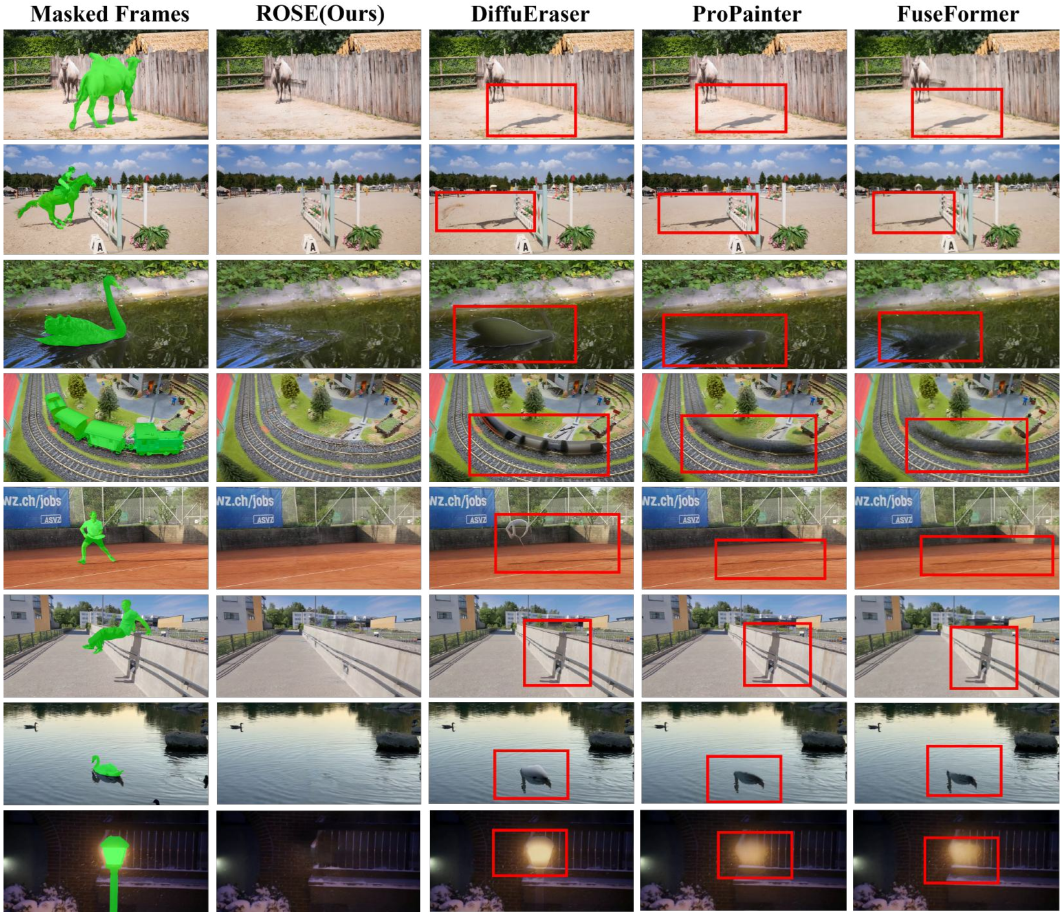

Figure 7: Qualitative comparison between our method and existing approaches on real-world samples. Our model demonstrates superior ability and effectively handles complex object-environment interactions, including shadows, reflections, and illumination changes.

output is then reshaped into a 3D spatio-temporal grid with the same shape of video latents:

$$\hat{\mathbf{d}}_t = \text{Interpolate}\left(\text{Reshape}(\mathcal{D}_\theta(\mathbf{x})), \text{size} = (F, H, W)\right), \tag{2}$$

where the predicted mask $\hat{\mathbf{d}}_t \in [0,1]^{B\times 1\times F\times H\times W}$ is upsampled via trilinear interpolation from a coarse patch-level grid (F_p, H_p, W_p) to the full resolution (F, H, W). The module is trained under MSE loss supervision against the ground-truth difference mask $\mathbf{d}_t$ described in Eq. (3). It functions as an auxiliary self-localization signal to encourage the model to be sensitive to subtle visual effects introduced by object edits. Then the training objective of ROSE consists of two terms: the standard diffusion denoising loss and the auxiliary mask prediction loss:

$$\mathcal{L} = \mathbb{E}_{t,\mathbf{z}_0,\boldsymbol{\epsilon}}\left[\|\boldsymbol{\epsilon} - \hat{\boldsymbol{\epsilon}}\|_2^2 + \lambda\|\hat{\mathbf{d}}_t - \mathbf{d}_t\|_2^2\right], \tag{3}$$

where λ balances the two objectives. This formulation enables the difference mask predictor to guide the model in localizing and identifying regions where object-environment interactions occur.

5 Experiments

5.1 Experiment Settings

Training Data. Our dataset contains 16,678 synthetic video pairs rendered in Unreal Engine, each 6 seconds (90 frames) at 1920×1080 resolution. It features diverse urban, rural, and natural scenes with dynamic weather, lighting, and interactive objects.

Table 1: Quantitative comparison on the **synthetic paired** benchmark (PSNR↑ / SSIM↑ / LPIPS↓)

Category	Metric	ROSE(Ours)	DiffuEraser [23]	ProPainter [44]	FuseFormer [24]	FloED [9]	FGT [42]
Common	PSNR	**36.5998**	30.9326	31.9972	31.2325	29.8932	28.4331
	SSIM	**0.9517**	0.9204	0.9466	0.9154	0.9066	0.8819
	LPIPS	**0.0413**	0.0825	0.0515	0.0658	0.0738	0.0832
Shadow	PSNR	**33.7876**	28.9976	30.2427	28.5520	27.8932	27.5809
	SSIM	0.9225	0.9220	**0.9353**	0.8972	0.8834	0.8547
	LPIPS	0.0626	0.1119	**0.0619**	0.1035	0.1208	0.1172
Light Source	PSNR	**30.0739**	22.6541	23.4291	22.8571	22.3125	21.4579
	SSIM	**0.9209**	0.8832	0.8924	0.8630	0.8596	0.8433
	LPIPS	**0.0862**	0.1403	0.1174	0.1410	0.1589	0.1347
Reflection	PSNR	**27.7344**	26.2914	26.9373	25.7707	25.1018	24.3986
	SSIM	0.8715	0.8619	**0.8763**	0.8345	0.8413	0.8421
	LPIPS	0.1129	0.1405	**0.1072**	0.1437	0.1651	0.1520
Mirror	PSNR	**28.3498**	22.1228	22.1206	22.3175	21.3789	22.6013
	SSIM	**0.9381**	0.8855	0.8994	0.8671	0.8678	0.8594
	LPIPS	**0.0878**	0.1751	0.1447	0.1596	0.1845	0.1653
Translucent	PSNR	**31.4264**	28.4520	29.8910	28.1712	27.3924	27.4802
	SSIM	**0.9470**	0.9259	0.9397	0.9168	0.8956	0.9034
	LPIPS	**0.0598**	0.1036	0.0722	0.0914	0.1134	0.1210
Mean	PSNR	**31.1221**	26.5024	27.1991	26.2566	25.4847	25.2353
	SSIM	**0.9170**	0.8981	0.9148	0.8795	0.8697	0.8641
	LPIPS	**0.0772**	0.1284	0.0946	0.1208	0.1324	0.1289

Evaluation Benchmark and Metrics. Existing benchmarks in the video inpainting domain mainly suffer two limitations. First, most of them lack access to paired edited videos following real-world physical rules, which restricts quantitative evaluation due to the absence of ground-truth. Second, they overlook the *side effects* induced by object-environment interactions hat are critical for assessing the semantic correctness and realism of inpainting. Consequently, these benchmarks fail to capture fine-grained challenges that frequently arise in real-world applications.

To address these gaps, we construct *ROSE-Bench*, a comprehensive evaluation benchmark on video object removal, consisting of following subsets:

(i) *Synthetic paired* benchmark tailored for evaluation under diverse physical interaction effects. Using the same simulation approach described in Sec. 3, the benchmark consists of 6 representative categories: common, light source, mirror, reflection, shadow, and translucent, each modeling a specific class of object-environment interaction. Every category contains 10 high-quality triplets of video sequences, *i.e.,* original, edited, and mask videos, offering precise and controllable evaluation of model behavior under different side-effect conditions.

(ii) *Realistic paired* benchmark constructed using a *copy-and-paste* strategy based on the video segmentation dataset dataset DAVIS [26]. We copy a masked object from one video into another. The resulting video with inserted object is treated as input, while the original unaltered video serves as the ground-truth. This process allows us to construct realistic and diverse test cases that mirror practical editing scenarios while preserving access to ground-truth supervision. For quantitative evaluation on paired benchmark, we compute PSNR [11], SSIM [37],

Table 2: Ablation study on ROSE-Bench (PSNR↑ / SSIM↑ / LPIPS↓)

Category	Metric	Base	w/ MRG	w/ MA	w/ DMP
Common	PSNR	32.58	35.24	33.54	35.68
	SSIM	0.937	0.950	0.949	0.943
	LPIPS	0.053	0.040	0.046	0.045
Shadow	PSNR	30.65	33.29	31.63	32.85
	SSIM	0.914	0.920	0.922	0.920
	LPIPS	0.081	0.061	0.072	0.064
Light	PSNR	24.99	30.37	28.89	30.13
	SSIM	0.894	0.923	0.911	0.922
	LPIPS	0.112	0.074	0.082	0.077
Reflect.	PSNR	25.39	27.71	26.97	27.41
	SSIM	0.836	0.843	0.845	0.841
	LPIPS	0.131	0.109	0.111	0.110
Mirror	PSNR	22.63	28.45	26.50	27.65
	SSIM	0.905	0.941	0.932	0.932
	LPIPS	0.142	0.076	0.086	0.092
Translucent	PSNR	27.43	30.98	31.14	31.24
	SSIM	0.925	0.949	0.948	0.946
	LPIPS	0.087	0.052	0.056	0.059
Mean	PSNR	27.28	30.84	29.77	30.82
	SSIM	0.902	0.918	0.916	0.917
	LPIPS	0.101	0.071	0.076	0.074

Table 3: Quantitative comparison on **realistic paired** benchmark.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
ROSE(Ours)	31.34	**0.923**	**0.092**
DiffuEraser [23]	29.97	0.901	0.128
ProPainter [44]	**32.81**	0.917	0.122
FuseFormer [24]	26.52	0.885	0.151
FloED [9]	28.48	0.881	0.147
FGT [42]	27.53	0.874	0.135

and LPIPS [43] across both synthetic and real-world test sets. These metrics capture both low-level structural fidelity and perceptual similarity, assessing the model performance under various side-effect challenges.

(iii) *Realistic unpaired* benchmark containing real videos with masks. Different from the second subset, we directly feed real-world videos into model, which are also sampled from DAVIS [26]. To conduct evaluation without ground-truth, we select related metrics from the VBench [13], a widely-adopted benchmark on text-to-video generation, for evaluating the quality of output videos on motion smoothness, background consistency and temporal flickering.

Implementation Details. In the training process, we resize all the video pairs into the resolution of 720×480 and use 81 frames for training. The backbone model is a controllable generation variant of Wan2.1 1.3B version [35]. We fully train the model together with the difference mask predictor in 80000 optimization steps with 0.00002 learning rate on 4 NVIDIA H800 GPUs.

5.2 Comparisons with Previous Methods

Quantitative Evaluation. For quantitative evaluation, we compare our method with flow-based transformers (ProPainter [44], FuseFormer [24], FGT [42]) and diffusion-based methods (DiffuEraser [23], FLoED [9]). We evaluate all methods on the three components of ROSE-Bench: synthetic paired benchmark (Tab. 1), realistic paired benchmark (Tab. 3), and real-world videos (Tab. 4). Our model achieves superior performance in object removal, as measured by PSNR [11], SSIM [37], and LPIPS [43], and excels in maintaining motion smoothness, background consistency, and subject consistency in Tab. 4.

Table 4: VBench-based evaluation on the **realistic unpaired** benchmark. (Best scores are **bolded**).

Method	**Motion Smoothness ↑**	**Background Consistency ↑**	**Temporal Flickering ↓**	**Subject Consistency ↑**	**Imaging Quality ↑**
DiffuEraser	0.972	0.902	**0.931**	0.891	**0.658**
ProPainter	**0.975**	0.917	0.932	0.903	0.626
FuseFormer	0.971	0.905	0.938	0.892	0.625
FloED	0.973	0.904	0.932	0.889	0.618
FGT	0.971	0.897	0.933	0.895	0.614
ROSE(Ours)	**0.975**	**0.923**	0.936	**0.908**	0.630

Table 5: Inference Efficiency.

Method	**All Parameters(M)**	**Average Runtime(s/frame)**	**Average GPU Memory(G)**
ROSE(Ours)	1564.4	1.39	21.05
DiffuEraser [23]	1952.3	1.02	26.12
ProPainter [44]	39.4	0.10	11.75
FuseFormer [24]	64.1	0.13	16.30
FloED [9]	1346.8	1.63	31.41
FGT [42]	42.3	1.43	14.38

Table 6: Ablation studies on the backbone model testing on synthetic paired benchmark.

Metrics	**DiffuEraser [23] before training**	**DiffuEraser after training**	**Wan using baseline settings**	**ROSE(Ours)**
PSNR↑	26.5024	27.0162	27.2863	**31.1221**
SSIM↑	0.8981	0.9010	0.9025	**0.9170**
LPIPS↓	0.1284	0.1106	0.1013	**0.0772**

Qualitative Evaluation. For qualitatitve evaluation, we compare our method with ProPainter [44], FuseFormer [24] and DiffuEraser [23]. Qualitative visualization results can be seen in Fig. 7. In the Fig. 7, we demonstrate cases with various different side effects like shadows, reflection and lumination changes and we can obviously find that our model shows superior performance over other methods. The side effects areas that previous works fail to fill in have been framed in red boxes.

Inference Efficiency. Tab. 5 compares the parameter scale, inference latency, and memory footprint across different methods. All evaluations are conducted on 65-frame input videos with a resolution of 720×480, using float16 precision and NVIDIA H800 GPUs.

5.3 Ablation Study

We perform ablation studies to demonstrate the effectiveness of our designs. We keep training settings same as in Sec. 5.1 to ensure the fairness of comparisons and we evaluate our methods on the synthetic paired benchmark. We set the baseline with the following settings: use the "mask-and-inpaint" paradigm, without mask augmentation and difference mask predictor. And in Tab. 2, MRG stands for mask region guidance, MA stands for mask augmentation and DMP stands for difference mask predictor. In Tab. 2, we have shown that our primary designs are effective and useful.

Also, Tab. 6 reports quantitative results comparing (i) DiffuEraser [23] without training on ROSE-Dataset, (ii) DiffuEraser trained on ROSE-Dataset, (iii) Wan2.1 [35] trained on ROSE-Dataset under baseline settings, and (iv) our proposed method ROSE. The results demonstrate that training on our ROSE-Dataset consistently improves overall performance, while the stronger base model further enhances object removal capability.

6 Discussion

This paper introduces ROSE, a unified framework for video object removal that addresses both target objects and their side effects, such as shadows, reflections, and lighting distortions. By leveraging synthetic data from a 3D rendering pipeline, we alleviate real-world data scarcity while ensuring diverse scenes and camera motions. Our diffusion transformer architecture excels in object localization and side effect removal via differential mask supervision. The proposed ROSE-Bench offers systematic evaluation for object-environment interactions, addressing a key gap in video inpainting. Extensive experiments show that ROSE significantly outperforms prior methods and generalizes well to real-world videos. These contributions advance video editing and set new benchmarks for handling complex visual artifacts. Future work will explore real-time optimization and broader environmental effects to further bridge synthetic and real-world domains. Despite its strengths, ROSE has **limitations**: (1) It may produce flickering artifacts under large motion, as shown in Tab. 4; (2) Inference time grows with video length, reducing efficiency on long sequences.

References

[1] Sand AI. Magi-1: Autoregressive video generation at scale. https://github.com/SandAI-org/MAGI-1, 2025.

[2] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: Learning joint representations for vision and text using cross-modal contrastive learning. In *NeurIPS*, 2021.

[3] Yuxuan Bian, Zhaoyang Zhang, Xuan Ju, Mingdeng Cao, Liangbin Xie, Ying Shan, and Qiang Xu. Videopainter: Any-length video inpainting and editing with plug-and-play context control. *arXiv preprint arXiv:2503.05639*, 2025.

[4] Jiayin Cai, Changlin Li, Xin Tao, Chun Yuan, and Yu-Wing Tai. Devit: Deformed vision transformers in video inpainting. In *ACM MM*, 2022.

[5] Ya-Liang Chang, Zhe Yu Liu, Kuan-Ying Lee, and Winston Hsu. Free-form video inpainting with 3d gated convolution and temporal patchgan. In *ICCV*, 2019.

[6] Epic Games. Fab. https://www.fab.com/, 2024.

[7] Epic Games. Unreal engine 5.3. https://www.unrealengine.com/, 2024.

[8] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024.

[9] Bohai Gu, Hao Luo, Song Guo, and Peiran Dong. Advanced video inpainting using optical flow-guided efficient diffusion. *arXiv preprint arXiv:2412.00857*, 2024.

[10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

[11] Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *ICPR*, 2010.

[12] Yuan-Ting Hu, Heng Wang, Nicolas Ballas, Kristen Grauman, and Alexander G. Schwing. Proposal-based video completion. In *ECCV*, 2020.

[13] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *CVPR*, 2024.

[14] Longtao Jiang, Zhendong Wang, Jianmin Bao, Wengang Zhou, Dongdong Chen, Lei Shi, Dong Chen, and Houqiang Li. Smarteraser: Remove anything from images using masked-region guidance. In *CVPR*, 2025.

[15] Xuan Ju, Xian Liu, Xintao Wang, Yuxuan Bian, Ying Shan, and Qiang Xu. Brushnet: A plug-and-play image inpainting model with decomposed dual-branch diffusion. In *ECCV*, 2024.

[16] Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Deep video inpainting. In *CVPR*, 2019.

[17] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *CVPR*, 2023.

[18] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, Kathrina Wu, Qin Lin, Junkun Yuan, Yanxin Long, Aladdin Wang, Andong Wang, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Hongmei Wang, Jacob Song, Jiawang Bai, Jianbing Wu, Jinbao Xue, Joey Wang, Kai Wang, Mengyang Liu, Pengyu Li, Shuai Li, Weiyan Wang, Wenqing Yu, Xinchi Deng, Yang Li, Yi Chen, Yutao Cui, Yuanbo Peng, Zhentao Yu, Zhiyu He, Zhiyong Xu, Zixiang Zhou, Zunnan Xu, Yangyu Tao, Qinglin Lu, Songtao Liu, Dax Zhou, Hongfa Wang, Yong Yang, Di Wang, Yuhong Liu, Jie Jiang, and Caesar Zhong. Hunyuanvideo: A systematic framework for large video generative models, 2025.

[19] Black Forest Labs. Flux. `https://github.com/black-forest-labs/flux`, 2024.

[20] Minhyeok Lee, Suhwan Cho, Chajin Shin, Jungho Lee, Sunghun Yang, and Sangyoun Lee. Video diffusion models are strong video inpainter. *arXiv preprint arXiv:2408.11402*, 2024.

[21] Sungho Lee, Seoung Wug Oh, DaeYeun Won, and Seon Joo Kim. Copy-and-paste networks for deep video inpainting. In *ICCV*, 2019.

[22] Ang Li, Shanshan Zhao, Xingjun Ma, Mingming Gong, Jianzhong Qi, Rui Zhang, Dacheng Tao, and Ramamohanarao Kotagiri. Short-term and long-term context aggregation network for video inpainting. In *ECCV*, 2020.

[23] Xiaowen Li, Haolan Xue, Peiran Ren, and Liefeng Bo. Diffueraser: A diffusion model for video inpainting. 2025.

[24] Rui Liu, Hanming Deng, Yangyi Huang, Xiaoyu Shi, Lewei Lu, Wenxiu Sun, Xiaogang Wang, Jifeng Dai, and Hongsheng Li. Fuseformer: Fusing fine-grained information in transformers for video inpainting. In *ICCV*, 2021.

[25] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.

[26] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016.

[27] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. 2023.

[28] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

[29] Dvir Samuel, Matan Levy, Nir Darshan, Gal Chechik, and Rami Ben-Ari. Omnimattezero: Fast training-free omnimatte with pre-trained video diffusion models. *arXiv preprint arXiv:2503.18033*, 2025.

[30] Fengyuan Shi, Jiaxi Gu, Hang Xu, Songcen Xu, Wei Zhang, and Limin Wang. Bivdiff: A training-free framework for general-purpose video synthesis via bridging image and video diffusion models. In *CVPR*, 2024.

[31] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. *arXiv preprint arXiv:1503.03585*, 2015.

[32] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021.

[33] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. *arXiv preprint arXiv:2109.07161*, 2021.

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[35] Wan Video. Wan: Open and advanced large-scale video generative models. `https://github.com/Wan-Video/Wan2.1`, 2025.

[36] Chuan Wang, Haibin Huang, Xiaoguang Han, and Jue Wang. Video inpainting by jointly learning temporal structure and spatial details. In *AAAI*, 2019.

[37] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *TIP*, 2004.

[38] Runpu Wei, Zijin Yin, Shuo Zhang, Lanxiang Zhou, Xueyi Wang, Chao Ban, Tianwei Cao, Hao Sun, Zhongjiang He, Kongming Liang, and Zhanyu Ma. Omnieraser: Remove objects and their effects in images with paired video-frame data. *arXiv preprint arXiv:2501.07397*, 2025.

[39] Jianzong Wu, Xiangtai Li, Chenyang Si, Shangchen Zhou, Jingkang Yang, Jiangning Zhang, Yining Li, Kai Chen, Yunhai Tong, Ziwei Liu, et al. Towards language-driven video inpainting via multimodal large language models. *arXiv preprint arXiv:2401.10226*, 2024.

[40] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. Youtube-vos: Sequence-to-sequence video object segmentation. In *ECCV*, 2018.

[41] Yue Xu, Jiabo Ye, Yifan Xu, Hang Zhou, Wayne Wu, and Ziwei Liu. Video-llava: Learning multimodal video instruction-following. In *EMNLP*, 2023.

[42] Kaidong Zhang, Jingjing Fu, and Dong Liu. Flow-guided transformer for video inpainting. In *ECCV*, 2022.

[43] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[44] Shangchen Zhou, Chongyi Li, Kelvin C. K. Chan, and Chen Change Loy. Propainter: Improving propagation and transformer for video inpainting. In *ICCV*, 2023.

[45] 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. In *AAAI*, 2025.

[46] Xueyan Zou, Linjie Yang, Ding Liu, and Yong Jae Lee. Progressive temporal feature alignment network for video inpainting. In *CVPR*, 2021.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims made in the abstract and introduction accurately summarize the contributions of the paper. Specifically, the paper proposes a novel framework that improves video inpainting quality by introducing a Difference Mask Predictor and Mask Region Guidance. These components are explicitly mentioned in the abstract and are substantiated by both qualitative and quantitative experiments. The scope and generalizability are clearly discussed with respect to diverse scenarios in the benchmark.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a dedicated Limitations in Sec. 6. It acknowledges that the current model still has some temporal flickering problems and the inference time increases with the video length.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: The paper includes a theoretical formulation of the loss functions and optimization objectives, all of which are fully stated with the necessary assumptions. While no formal theorems are proved, the mathematical foundations are complete and clearly documented either in the main text or supplemental material.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper fully discloses the architectural details, dataset preprocessing steps, training schedules, loss functions, and evaluation protocols used to produce the experimental results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code and pre-trained models will be released upon acceptance, and the supplemental material contains detailed instructions to run experiments, including dataset preparation, model training, and evaluation. An anonymized GitHub link is included for reviewing purposes.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All relevant details regarding training (e.g., data splits, learning rate, optimizer, batch size, and number of epochs) and evaluation (e.g., metrics, checkpoints) are clearly described in Sec. 5. This allows readers to understand and replicate the experimental protocol.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We follow the convention in prior works and report the performance on the standard settings.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper specifies the compute infrastructure used, the number of training epochs, and the total training time. An estimated compute cost for each model variant is also provided in the supplemental material.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research adheres to the NeurIPS Code of Ethics. All datasets used are publicly available and do not involve personally identifiable information or sensitive content. No human or animal subjects were involved.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss both of them in supplementary materials.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We describe the safeguards in the Appendix.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets and codebases used are properly cited and used under their respective licenses. License information is included in the supplemental material.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The newly introduced benchmark subset and pre-trained models are fully documented and will be made available with metadata, license information, and usage examples. Documentation is provided in the supplemental material.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: This paper does not involve LLMs as any important components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Dataset Details.

In the data preparation process, we synthesize video sequences within the Unreal Engine (UE) environment. The virtual cameras are programmatically controlled to automatically focus on the target object designated for removal. To simulate realistic motion, the cameras move randomly along one of the three principal axes—X, Y, or Z—in the UE world coordinate system. The movement distance along the chosen axis is adaptively adjusted according to the spatial scale and available space within each specific scene, ensuring plausible camera trajectories without clipping or unnatural transitions. By rendering the scene under these controlled camera motions, we obtain synchronized video triplets consisting of the original unedited video, the corresponding masked video with the target object occluded, and the edited video where the object is removed.

B Training Details.

During training, we use a batch size of 1 and randomly select a continuous sequence of 81 frames from triplets of the original, masked, and edited videos as input. We extract and concatenate features from Transformer layers block.5, block.15, and block.25(30 blocks in total) , resulting in a feature tensor of shape $[1, 28350, 4608]$, where 28350 corresponds to the flattened patch grid (e.g., $10 \times 15 \times 189$) and 4608 is the combined feature dimension from the three layers. The Difference Mask Predictor consists of a linear layer projecting the 4608-dimensional features to a 256-dimensional hidden space, followed by a GELU activation and a final linear layer mapping to a single output channel. The output, initially $[1, 28350, 1]$, is reshaped to $[1, 1, 10, 15, 189]$ and then upsampled via trilinear interpolation to the full resolution $[1, 1, 81, 480, 720]$, matching the ground-truth difference mask size. This design enables efficient patch-wise prediction of spatiotemporal difference masks, providing fine-grained supervision for the video editing task.

C Potential impacts.

Positive Impact. This work is expected to significantly contribute to the field of video object removal by providing a more robust and effective framework for erasing undesired regions in video sequences. By greatly enhancing the quality, temporal consistency, and semantic coherence of the inpainting results, it will not only push the boundaries of current video inpainting techniques but also offer a solid foundation for future research and practical applications in video editing, surveillance anonymization, and content restoration.

Negative Impact. Video inpainting technology, while powerful for restoring or editing visual content, can also bring about negative impacts such as facilitating misinformation through realistic content manipulation, undermining the credibility of video evidence in legal and forensic contexts, and raising ethical concerns regarding privacy and consent. If misused, it may distort historical records, violate intellectual property rights, or propagate biased or misleading visual narratives, posing serious risks to information integrity and social trust.

D More Visualization Results.

In Figs. 8 to 10, we provide additional qualitative results to further demonstrate the generalization ability of our model. Specifically, Fig. 8 presents more representative examples under common scenarios. Fig. 9 shows cases where the input masks are imperfect, illustrating that our model remains robust and can still generate plausible results. Furthermore, Fig. 10 includes results on unseen domains such as underwater, mobile, and drone footage, revealing the model's strong adaptability to diverse real-world conditions.

E More Comparison Visualization Results.

In Fig. 11, we present more comparison results of our model with DiffuEraser [23], ProPainter [44], and FuseFormer [24].

F Some Design Thinkings.

Applicability of Mask Region Guidance (MRG) to Other Models. The motivation of MRG is well justified by its significant improvements as shown in Tab. 2. To further examine its generality, we initially considered applying MRG to other representative video inpainting models, such as ProPainter [44] and DiffuEraser [23]. However, we found that both models are structurally incompatible with our MRG paradigm, which makes a fair comparison infeasible. *ProPainter* relies on an image propagation mechanism that requires a masked video as input. This pipeline design fundamentally conflicts with the pixel-level regional supervision of MRG, which operates on explicit mask-aware features. Consequently, adapting ProPainter to MRG would require

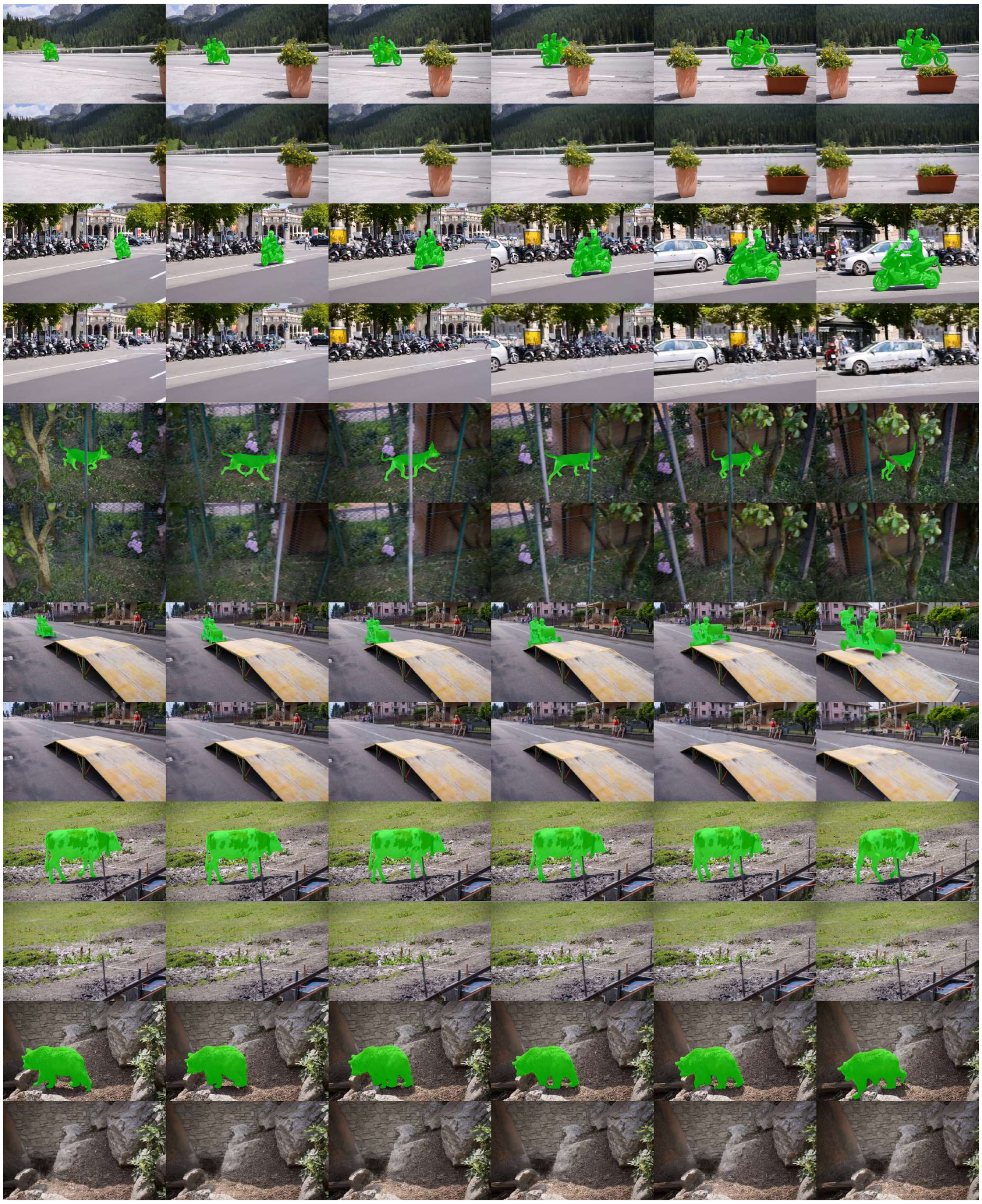

Figure 8: More visualizations.

re-engineering its propagation module, rendering it impractical for our framework. *DiffuEraser*, on the other hand, employs a dual-branch architecture built upon a pretrained image inpainting network (BrushNet). To integrate MRG into DiffuEraser, one must redesign BrushNet's architecture and introduce additional training objectives for mask-region alignment. Such modifications would alter the model's intrinsic inductive biases and deviate from its original optimization setup, preventing a meaningful comparison.

Given these architectural and methodological constraints, we decided not to pursue further experiments with these two models. Nonetheless, our investigation highlights that MRG is conceptually orthogonal and could, in principle, be generalized to future architectures designed with explicit mask-aware conditioning.

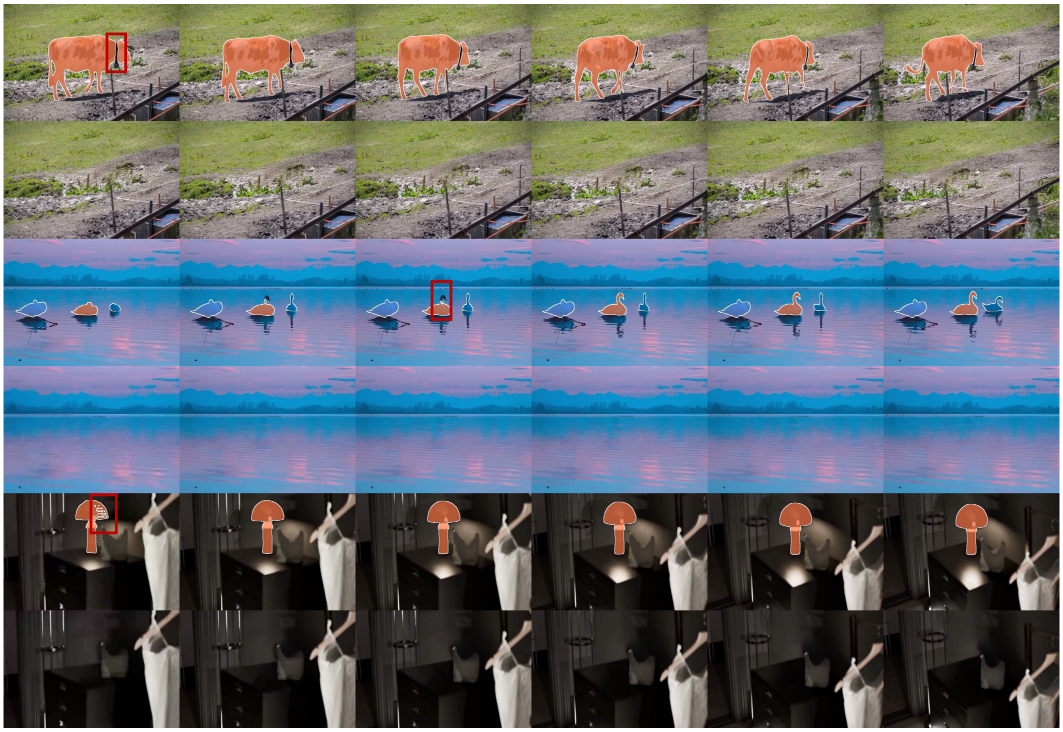

Figure 9: Cases about dealing with imperfect mask input.(The imperfect parts have been marked by red box. Zoom in for better view.)

G Safeguards

All data used in this work are synthetically generated within Unreal Engine, containing no real human, biometric, or copyrighted content. To prevent potential misuse, we plan to release the ROSE model and dataset under a research-only license. Model checkpoints will be accessible only to verified academic users who agree to responsible-use terms. We explicitly prohibit applications related to deepfake creation, disinformation, or any malicious video manipulation.

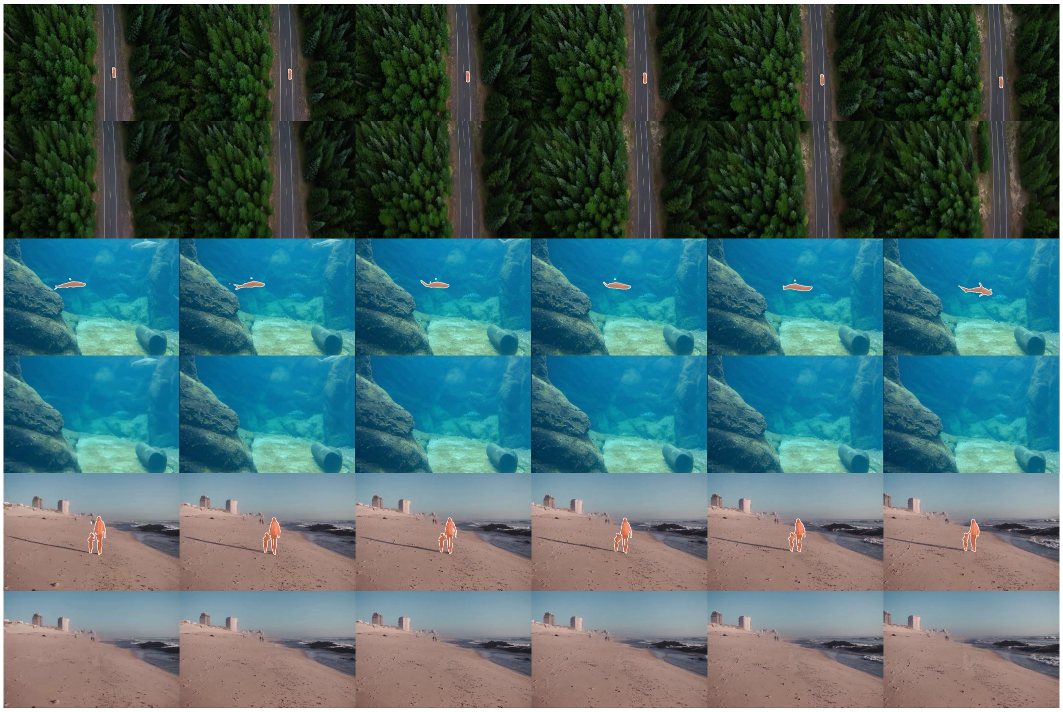

Figure 10: More unseen domain cases like underwater, mobile and drone footage.

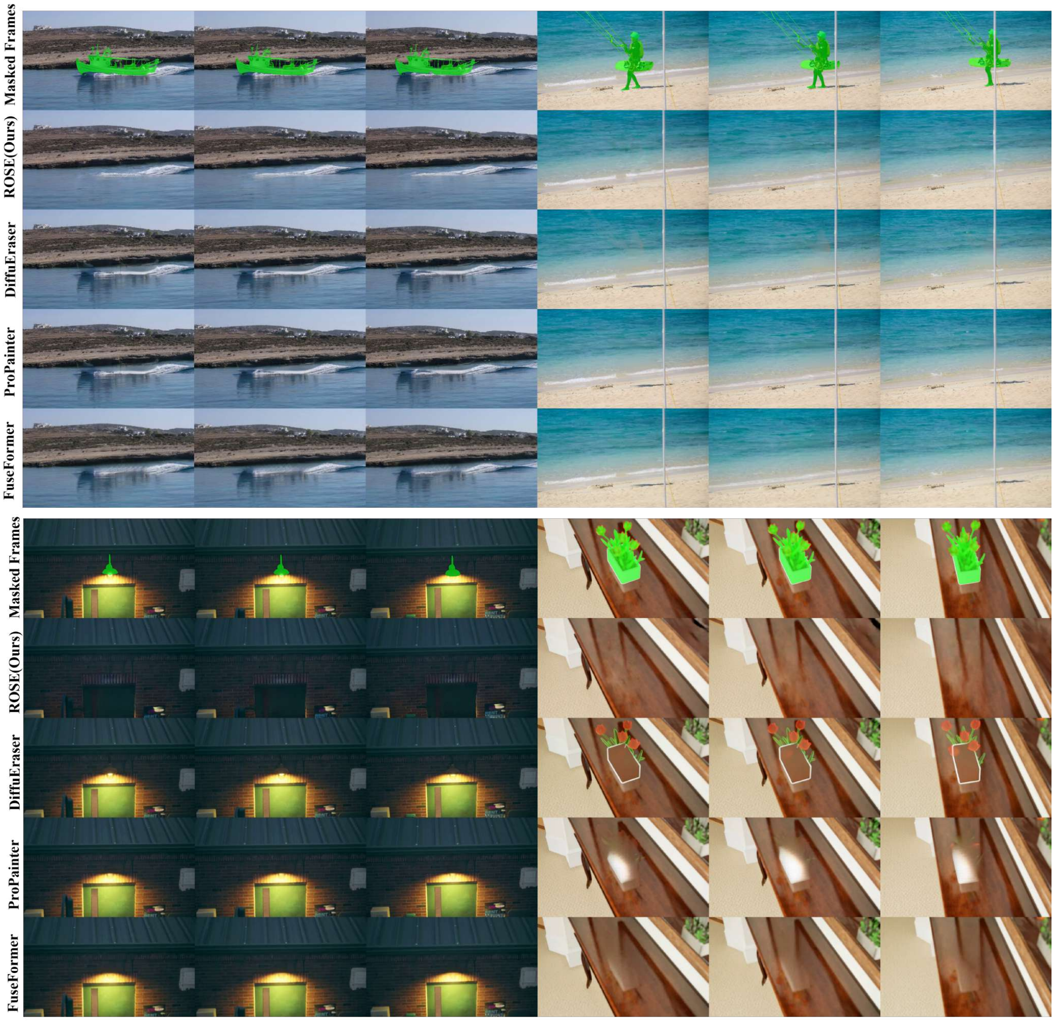

Figure 11: More comparisons.