Video Inpainting with Region-Aware Propagation

Akshata Kumble Amit Gurpur Raghav Jadia Rituraj Navindgikar
Northeastern University, Boston, MA
Email: {kumble.a, gurpur.a, jadia.r, navindgikar.r }@northeastern.edu

Abstract

Video object removal is a fundamental task in video editing, requiring the seamless elimination of unwanted objects while preserving temporal consistency and visual fidelity. While recent approaches like Generative Video Propagation (GenProp) have demonstrated the efficacy of rectifying single-frame edits across a sequence, the lack of public implementation details presents a barrier to adoption and further research. This report presents a reproducibility study and architectural analysis of the GenProp architecture, implemented using a frozen Stable Video Diffusion (SVD) backbone. We introduce a framework that utilizes a Selective Content Encoder (SCE) to extract edit-aware features and a Mask Prediction Decoder (MPD) for explicit temporal supervision. Beyond standard implementation, we investigate methods to reduce training costs by experimenting with lightweight Ctrl-Adapter modules as an alternative to full encoder training. Our experiments show that while the lightweight Ctrl-Adapter approach led to latent variance explosion and visual artifacts due to feature density incompatibilities, the ControlNet-style SCE ultimately provided temporally consistent removal, proving to be the more reliable solution. We analyze these trade-offs, providing a baseline implementation for generative propagation and experimenting with fine-tuning the model for sparse inpainting tasks.

1. Introduction

Video object removal is a critical task in computational video editing, aimed at eliminating unwanted objects from a scene while preserving temporal coherence and perceptual quality. Achieving seamless edits across frames remains challenging due to issues such as temporal drift, flickering artifacts, and the complexity of handling fast motion, occlusions, and dynamic backgrounds. Traditional methods like frame-by-frame inpainting and iterative diffusion often break temporal consistency, incur high computational costs, and struggle with structural alignment across long video sequences.

Recent advancements in image-to-video (I2V) generative models, particularly Stable Video Diffusion (SVD), have demonstrated impressive capabilities in generating temporally coherent motion from static images. Leveraging these generative priors for object removal has become a promising direction. One notable framework is Generative Video Propagation (GenProp)[10], which propagates edits from a single keyframe to an entire video sequence. However, the lack of an open-source implementation and detailed architectural insights into GenProp necessitates a rigorous reproduction to validate its efficacy and better understand its underlying mechanics.
In this work, we present a system based on the GenProp methodology that addresses these challenges by utilizing a frozen SVD backbone and training only lightweight conditioning modules. Like GenProp, we employ a Selective Content Encoder (SCE) to process both original and edited latents, injecting features into the UNet via adapter layers, and a Mask Prediction Decoder (MPD) to ensure proper propagation of edits. Given our limited computational resources, designing lightweight conditioning modules was crucial, leading to experiments with parameter-efficient solutions. Additionally in our final implementation of GenProp we applied techniques to improve model stability.

Our key contributions include:

- **GenProp Reproduction & Adaptation**: We provide a verified implementation of GenProp using SVD, detailing the conditioning, feature injection, and region-aware loss designs necessary for stable propagation.
- **Analysis of Failure Modes**: We document the limitations of dense feature adapters in inpainting tasks and analyze why Ctrl-Adapter approaches led to issues such as VAE collapse and spatial misalignment, in contrast to our robust final implementation.
- **Architectural Efficiency Analysis**: We compare different architectural choices, including ControlNet-based SCE and Ctrl-Adapter variants, offering insights into the trade-offs between efficiency and performance.

Source code is available on our GitHub Repository.

2. Related Work

2.1. Video Inpainting and Object Removal

Early video inpainting methods [3] relied on patch-based techniques combined with optical flow to ensure temporal consistency. Flow-guided methods propagate information from known pixels into missing regions, but accumulate errors due to misalignments and struggle with occlusions [14]. ProPainter [19] improved efficiency by using dual-domain propagation with a mask-guided Transformer, but still requires per-frame object masks and struggles with complex, non-rigid deformations. FlowEdit [6] also utilizes flow-based techniques, preserving structural integrity during edits. However, its focus on maintaining the original structure limits its application for object removal, where altering background content and removing objects is the goal.

2.2. Diffusion-Based Video Editing

The success of diffusion models in image generation has led to their extension to video editing. Video diffusion models [4] and Stable Video Diffusion (SVD) [1] use temporal attention layers and large-scale video datasets to generate realistic motion. Methods like Tune-A-Video [16] and Video-P2P [11] fine-tune image diffusion models for video, but they often suffer from collateral changes in the background when applied to tasks like object removal. Additionally, their high computational cost makes frame-by-frame editing slow.

FlowEdit [6] introduced a flow-matching approach using direct ODE paths to minimize transport costs, preserving structure during edits. However, its strong structural bias makes it less suited for object removal, where background hallucination is necessary.

2.3. Generative Video Propagation

Generative propagation methods train models to apply single-frame edits to entire videos in one pass, combining segmentation propagation with deep generative modeling. GenProp [10] exemplifies this approach with a one-shot propagation framework handling object removal, insertion, replacement, and tracking using a unified model. It leverages a pre-trained image-to-video diffusion model augmented with a Selective Content Encoder (SCE) and Mask Prediction Decoder (MPD) to ensure targeted modifications. By training on synthetic data with known edited regions and employing a region-aware loss, GenProp learns to preserve unedited areas while propagating changes effectively.

2.4. Segmentation and User Guidance

Accurate segmentation is essential for video object removal. Traditional methods rely on manual rotoscoping or mask tracking for per-frame object masks. SAM (Segment Anything Model) [5] simplifies this by providing zero-shot segmentation across various objects.

In modern workflows, users apply SAM to the first frame to select the object, and generative models track and remove it in subsequent frames. Unlike traditional methods, generative propagation only requires an initial mask, reducing user effort and improving efficiency.

2.5. Parameter-Efficient Control Mechanisms

Recent work has focused on improving control over diffusion models, balancing precision with computational efficiency. ControlNet [17] introduced a framework for spatial control by freezing the base model and training a copy of the encoding layers, enabling conditioning on inputs like edge maps and segmentation while maintaining the generative power of the pre-trained model.

Building on this, Ctrl-Adapter [8] introduced lightweight adapter modules, allowing control over diffusion models without the need for full retraining of the ControlNet. This reduces computational cost and mitigates feature mismatches, making it efficient for adapting pre-trained models to new tasks. However, our experiments show that while Ctrl-Adapter works well for image tasks, its dense features are incompatible with sparse edit-propagation tasks, such as object removal in video latents, causing training instability and visual artifacts.

2.6. Image-to-Video Models for Guided Video Editing

Large-scale image-to-video models have advanced video generation by learning spatio-temporal representations from vast datasets. Models like VideoCrafter [2] and SVD [1] achieve strong temporal coherence using spatio-temporal UNet architectures, enabling realistic motion from single images. While excelling at unconditional or text-conditional video generation, these models require specialized conditioning for more complex tasks like object removal.

Our work integrates insights from these prior approaches with an emphasis on computational efficiency. We adopt the generative propagation paradigm introduced in GenProp, leverage the temporal modeling capabilities of the SVD backbone, and incorporate ControlNet-inspired conditioning via parameter-efficient adapter modules. Our objective is to implement GenProp and experiment with lightweight variant that mitigates the computational bottlenecks hindering practical deployment, while remaining fully compatible with pretrained video diffusion models through careful architectural design.

3. Main Idea

Our work builds upon the central principles introduced in the Generative Video Propagation (GenProp) framework.

While GenProp provides a high-level description of its methodology, many architectural and training details remain unspecified. To enhance computational efficiency during training and training stability, we explored various architectural and conditioning variants, as well as different training configurations.

3.1. Problem Formulation

Consider an original video sequence, $V_{orig} = \{v_1, v_2, \ldots, v_T\}$. The sequence contains an object that we attempt to remove. Let v_1^{edited} be the edited first frame, where the object is removed.

Given the original video and the edited first frame, the model should be able to propagate the edit to the whole video. While editing the video, our aim is to ensure:

1. The object is removed in all frames of the video.
2. The temporal consistency of the video is maintained.
3. The unedited potions of the video remain similar to those in the original video

3.2. Architecture Overview

In our implementation, we make use of the state-of-the-art features that the GenProp[10] paper has to offer:

1. Selective Content Encoder (SCE): The SCE encodes structural information from the original video, allowing the model to retain unchanged content while providing a stable reference for propagating edits across time.
2. Mask Prediction Decoder (MPD): The MPD predicts which spatiotemporal regions are expected to change, offering an auxiliary supervisory signal while training, that helps the model distinguish edited areas from preserved content.
3. Injection layers: Lightweight $1 \times 1 \times 1$ convolutional adapters inject SCE features into the frozen diffusion backbone, enabling controlled modulation of the generative process.

These components are integrated with an I2V model (Image-to-Video Model) for edit propagation. The auxiliary components of the architecture assist the backbone in editing the object in the video.

A detailed description of the all the architecture implementations are given in Section 4.

3.3. Training Strategy

Our training strategy mirrors the approach implemented in GenProp, where the model is encouraged to retain unchanged content while propagating edits from the keyframe. While we are not given the exact implementation details for training the model, based on the information provided, we can make a rough estimate of the training process. We employ the same supervision structure, utilizing mask-aware guidance to differentiate between edited and unedited regions, but we implement it in a more lightweight form that is suitable for limited computational resources. The rest of the training pipeline follows the structure of the original framework. We keep the same approach for conditioning (using the SCE, MPD and Injection Layers), preparing latents, and applying supervision (using the RA loss), while making the adjustments needed for our SVD-based backbone.

3.4. Datasets

We evaluate our method on two standard video editing datasets:

DAVIS (Densely Annotated Video Segmentation) [13] provides high-quality segmentation masks across diverse scenarios including fast motion, occlusions, and complex backgrounds. We use DAVIS to evaluate temporal consistency and visual quality.

ROVI (Real-world Object Video Inpainting) [15] is designed specifically for video inpainting, providing paired original and edited videos and their masks that enable supervised training and quantitative evaluation of inpainting quality.

For training on DAVIS, we simulate edits by applying Gaussian blur to masked regions in original videos. This approach generates large-scale supervision while maintaining ground-truth correspondences between original and edited sequences.

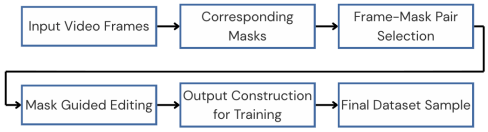

Figure 1. DAVIS Frame Editing Flow

The data construction process begins by sampling a short sequence of frames and their corresponding binary masks from the input video. Each frame–mask pair is processed by a mask-guided editing module that suppresses masked content by replacing it with a blurred approximation while preserving unmasked regions. The edited frame is computed as

$$I_{\text{edit}} = M \odot I_{\text{blur}} + (1 - M) \odot I,$$

where I denotes the original frame, I_{blur} is its blurred counterpart, and M is the binary mask.

After all frames are edited, the pipeline assembles the original sequence, the edited sequence, the edited first frame, and the mask sequence. This structured output provides the supervision necessary for training the propagation model.

Figure 2. Example: Original frame (left) being edited using the custom editing flow and the edited frame (right)

4. Overall System Architecture

4.1. Backbone I2V Model: SVD-XT

Our implementation builds upon Stable Video Diffusion-XT (SVD-XT), a latent diffusion model designed for high-fidelity image-to-video synthesis. SVD-XT is a pretrained generative prior capable of producing temporally coherent sequences from a single input image, making it an effective foundation for our edit-propagation pipeline. In our system, the diffusion backbone remains frozen, and only the auxiliary modules introduced in subsequent subsections are trained.

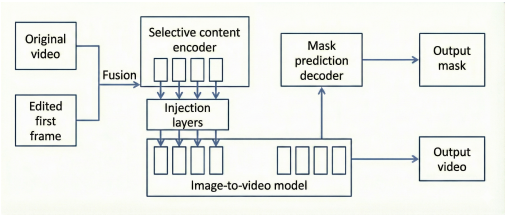

Figure 3. Architecture Flow

SVD-XT compresses each video frame into a compact 4-channel representation in the latent space of a variational autoencoder (VAE). This reduces computational cost while retaining sufficient spatial detail for high-quality video generation. In our implementation, all frames are mapped into this latent space using the pretrained VAE from the SVD-XT pipeline [1]. Given an edited key frame, the backbone constructs an initial latent sequence, perturbs it with Gaussian noise, and iteratively denoises it using a UNet with text, time, and motion conditioning. The resulting latent trajectory is decoded into a coherent video sequence.

The denoising dynamics follow the velocity-prediction parameterization, where the model predicts a velocity vector $v_\theta(z_t, t)$ as a linear combination of the latent sample and injected noise under the diffusion process. This formulation stabilizes optimization and empirically yields smoother temporal evolution in generated videos. In practice, the velocity target is computed from the scheduler coefficients associated with each timestep, and the model adopts these targets directly from the SVD-XT scheduler implementation

SVD-XT aligns well with our propagation-based formulation for the following reasons:

1. It generates plausible motion from a single static input and naturally propagates visual structure across future frames, thus influencing the entire sequence.
2. The UNet architecture's multi-scale hierarchy of spatiotemporal features is conditioned and enabled by the SCE and injection layers, allowing the model to maintain unedited structure while propagating the intended edit across all frames.

Freezing the backbone ensures that the generative prior remains stable while significantly reducing training cost, allowing the learnable components to focus on selective modification rather than relearning video generation.

Comparison to GenProp. The GenProp paper uses two I2V alternatives- a DiT-based I2V backbone and an SVD-style UNet variant, while our implementation uses the SVD-XT backbone exclusively and freezes the VAE, UNet, and image encoder to preserve the generative prior. Although GenProp emphasizes first-frame conditioning, it doesn't specify the velocity-prediction formulation used during training. In contrast, our system computes velocity targets directly from the SVD-XT scheduler coefficients. GenProp also refers to latent interactions between the original video and the edited frame, but the mechanism is not detailed. Our implementation formalizes this by explicitly concatenating and tiling first-frame latents within the conditioning pathway.

4.2. Selective Content Encoder (SCE)

The SCE injects structural information into the generative backbone by encoding the original video and the edited first frame into latent representations Z_{orig} and Z_{cond}. These representations are concatenated and processed through a shallow encoder reusing pretrained UNet down blocks, maintaining spatial–temporal inductive biases while minimizing computational overhead.

Each block processes concatenated latents, diffusion timestep embedding, added-time conditioning, and CLIP-derived image embeddings (from the SVD's image encoder). This enables the SCE to infer spatial regions aligning with the original content and those corresponding to the edited key frame. The encoder generates a multi-scale feature hierarchy $F_{\text{sce}} = \{f_i\}$ capturing global video structure and localized modifications.

These features are injected into corresponding layers of the frozen UNet through lightweight injection layers. The backbone receives guidance to preserve unedited regions while propagating the intended edit across frames. This design ensures the generative model maintains source video fidelity while coherently extending the key-frame edit.

Comparison to GenProp. The GenProp paper states that the Selective Content Encoder reuses the first set of down blocks from the backbone I2V model, but it doesn't specify the number of blocks or how latent concatenation is treated. Our implementation uses two replicated down blocks from the SVD-XT UNet and operates on concatenated original-video and tiled edited-frame latents. While the paper mentions bidirectional fusion between SCE and I2V features, our implementation has a strictly unidirectional flow, with SCE features guiding the backbone through multi-scale injection rather than performing any fusion within the encoder.

4.3. Control Injection Mechanism

To modulate the frozen UNet using the features extracted by the Selective Content Encoder (SCE), we introduce lightweight $1 \times 1 \times 1$ convolutional adapters paired with group normalization. For each scale i, the UNet hidden state h_i is updated using a residual formulation,

$$h_i' = h_i + \alpha \, \mathrm{Reshape}(\mathrm{GN}(\mathrm{Conv3D}(f_i))) \, ,$$

where f_i denotes the SCE feature at scale i, and α is a small scalar controlling the strength of the injected signal. All adapter parameters are initialized to zero, ensuring that the model initially reproduces the behavior of the pretrained SVD-XT backbone without any injected influence. As training proceeds, the adapters gradually learn how to incorporate SCE features in a controlled manner, enabling the model to preserve unchanged regions while coherently propagating edits derived from the first frame.

To interface these features with the UNet, control signals are injected through the following steps:

1. **Spatial Upsampling:** SCE features may exist at a lower spatial resolution, so bilinear interpolation is applied to align them with the resolution of the corresponding UNet feature maps.
2. **Channel Projection:** The SCE feature tensors typically have high channel dimensionality (e.g., 320 channels), whereas the UNet expects a smaller number of channels (e.g., 8 for the latent representation). A 1×1 convolution projects the SCE feature channels into the appropriate dimensionality for injection.
3. **Residual Scaling and Addition:** The projected features are scaled (e.g., by 0.1) and added to the UNet activations as a residual signal, enabling edit-aware conditioning without altering the UNet weights.

This mechanism provides a stable and efficient way to inject structural cues from the SCE into a fully frozen UNet, ensuring that the pretrained generative prior is preserved while still allowing effective propagation of edits throughout the video.

Comparison to GenProp. GenProp describes a zero-initialized MLP that injects SCE features into a generative model but doesn't specify how these signals are aligned in spatial resolution or projected into appropriate channel dimensions. Our implementation provides an explicit formulation: each SCE feature map is projected with a $1 \times 1 \times 1$ convolution, normalized, and added as a residual through registered forward hooks on designated UNet blocks. This operationalizes the paper's description and specifies the resolution-matching and channel-projection steps, producing a stable and transparent injection pathway. To ensure the injected signals do not overwhelm the UNet, we add a GroupNorm layer to the injection mechanism.

Table 1. Analysis of feature magnitude imbalance between Injection Layers and the backbone UNet after training without GroupNorm. We compare the standard deviation of the injected features (σ_{inj}) versus the UNet hidden states (σ_{unet}). The extremely high ratios indicate that the injection layers dominate and corrupt the signal unless regularized.

Layer Block	**Injection Std** (σ_{inj})	**UNet Std** (σ_{unet})	**Ratio** ($\sigma_{inj}/\sigma_{unet}$)
Block 0	17.2731	1.1724	14.73×
Block 1	812.7489	3.1155	260.88×

4.4. Mask Prediction Decoder (MPD)

The Mask Prediction Decoder (MPD) identifies regions to modify in the video sequence. It uses spatiotemporal features from the penultimate up block of the UNet, which represent spatial structure and temporal evolution. These features are processed by a compact 3D convolutional module with group normalization to capture localized editing cues.

The resulting tensor is upsampled to the latent spatial and temporal resolutions, yielding a predicted mask M_{pred} that assigns an edit likelihood to each latent pixel. During training, this mask is supervised with ground-truth synthetic masks, encouraging the model to accurately localize the regions influenced by the first-frame edit. By providing an explicit estimation of the edited area, the MPD enhances the model's ability to disentangle modified and preserved content, leading to more reliable and consistent propagation.

Comparison to GenProp. The GenProp paper introduces the Mask Prediction Decoder (MPD) as an auxiliary module that predicts edited regions while training. Its architectural details are abstract, referring only to an MLP applied to high-level features. Our implementation incorporates this design by extracting features from the penultimate up block of the UNet and determining the MPD input dimensionality at runtime. A structured sequence of 3D convolutions followed by trilinear upsampling produces a spatiotemporal mask, forming the paper's conceptual description and

resolving unspecified choices regarding feature resolution and decoder depth.

4.5. Region-Aware Loss Function (RA Loss)

We adopt the region-aware loss from GenProp, which separates supervision in edited and preserved regions. The total objective is

$$\mathcal{L} = \mathcal{L}_{\text{non-mask}} + \lambda_m \mathcal{L}_{\text{mask}} + \lambda_g \mathcal{L}_{\text{grad}} + \lambda_p \mathcal{L}_{\text{MPD}}. \quad (1)$$

4.5.1. Masked and Non-Masked Reconstruction Losses

Following GenProp, we apply separate MSE losses to edited and preserved regions:

$$L_{\text{mask}} = \|M \odot (\hat{v} - v_t)\|_2^2$$

$$L_{\text{non-mask}} = \|(1 - M) \odot (\hat{v} - v_t)\|_2^2$$

where M denotes the edited region and v_t is the v-prediction target produced under the diffusion schedule.

4.5.2. Gradient Suppression Loss

To discourage the Selective Content Encoder from encoding information inside the edited region, we penalize finite-difference gradients of its features:

$$L_{\text{grad}} = \mathbb{E}[\, M \odot \|\Delta f_{\text{SCE}}\|\,],$$

ensuring that reconstruction of missing content is handled by the generative model rather than the encoder pathway.

4.5.3. Mask Prediction Loss

The Mask Prediction Decoder is trained with a binary cross-entropy objective,

$$L_{\text{MPD}} = \text{BCE}(\hat{M}, M),$$

providing explicit supervision for identifying regions expected to change.
Where, $\hat{\text{M}}$ is the mask produced by the MPD and M is the ground truth mask.

The total training loss is a weighted combination of these terms, and only the SCE, injection layers, and MPD parameters are updated. All components of the SVD backbone remain frozen, preserving the pretrained generative prior while substantially reducing training cost. We use $\lambda_m = 2.0$ and $\lambda_g = \lambda_p = 1.0$.

Comparison to GenProp. The region-aware loss in GenProp includes masked and non-masked reconstruction terms, a gradient-suppression term, and an auxiliary mask-prediction loss. Our implementation reproduces this formulation with the same weighting coefficients, but introduces two clarifications not explicitly stated in the paper. First, the gradient penalty is applied only to the final SCE feature block due to computational constraints, whereas the paper does not specify which layers are used. Second, masks are interpolated using nearest-neighbor sampling in latent space rather than Gaussian smoothing. These adaptations preserve the intent of the loss while aligning it with the architecture and training pipeline used in our system.

5. Experiments

We conducted comprehensive experiments to evaluate different architectural choices and explore more computationally efficient methods to implement the GenProp design.

We first describe our experimental setup, then present our experiments comparing architectural alternatives that we attempted to implement. Our experiments compares three main approaches:

1. Initial Custom 3D U-Net with SCE
2. Ctrl-Adapter based experiments
   (a) Ctrl-Adapter Based SCE
   (b) Ctrl-Adapter without Attention Layers
3. ControlNet-based SCE (Our final approach)

5.1. Implementation Details

As mentioned previously, we build on Stable Video Diffusion XT (SVD-XT), keeping the base model frozen throughout training. Input videos are processed at 576×576 resolution with 25 frames per clip during training, yielding latent representations of size 72×72 after $8\times$ spatial downsampling.

The SCE reuses the first two UNet down blocks, and the MPD consists of three convolutional layers with channel dimensions $C \to 256 \to 128 \to 1$.

We train using AdamW with learning rate 5×10^{-5} and batch size 2 per GPU with 2-step gradient accumulation. Training uses BF16 mixed precision with gradient checkpointing for memory efficiency, running for 17 epochs total. Loss weights are set to $\lambda_m = 2.0$ and $\lambda_g = \lambda_p = 1.0$.

At inference, we use the Euler discrete scheduler with 25 sampling steps and guidance scale 3.0, generating 5–14 frames depending on the evaluation setting.

5.2. Exploratory Study

5.2.1. Approach 1: Initial custom 3D U-Net with SCE

As an initial baseline, we developed a custom 3D U-Net architecture conditioned by a Selective Content Encoder (SCE). The model processes original videos as 5D tensors $[B, C, T, H, W]$ through a 3D U-Net generator with three downsampling blocks to build spatio-temporal feature hierarchies. The SCE extracts edit-specific information from the original, edited, and masked first frames using 2D convolutions. These features are injected into the U-Net bottle-

neck and upsampled to match the feature resolution.

Results:

- **Feasibility**: This baseline demonstrated the potential of SCE-based conditioning for video edit propagation.
- **Limitations**:
  - Limited capacity compared to pre-trained models like SVD.
  - Difficulty handling complex scenes and fast motion due to training from scratch on limited computational resources.
  - Blurry results in inpainted regions, which we attribute to insufficient training iterations (only 2 epochs) given the constrained GPU memory and compute budget.

These limitations motivated our shift to leveraging a frozen pre-trained SVD backbone, which provides stronger generative priors and enables training with reduced computational requirements.

5.2.2. Approach 2: Ctrl-Adapter Based SCE

We experimented with using Ctrl-Adapter as a lightweight alternative to the full ControlNet-based SCE. The motivation was to reduce the number of trainable parameters by adapting pre-trained ControlNets rather than training full encoder copies.

Steps taken to implement Ctrl-Adapter based SCE:

- Keeping both I2V model and ControlNet backbone frozen
- Training only adapter layers + MPD with RA loss
- Implementing inverse-timestep sampling to map continuous timesteps of the SVD backbone with discrete timesteps expected by ControlNet
- Using ControlNet Tile, which operates on RGB images for resolution enhancement/reconstruction

Issues Encountered:

1. Instability: Multicolor flashes and mosaic patterns appeared in generated videos
2. Variance Explosion: VAE collapse occurred due to extreme latent values
3. Spatial Misalignment: Small padding mismatches caused garbled output
4. Task Incompatibility: ControlNet Tile was trained for reconstruction, not for ignoring objects (as required by SCE)

Attempted Fixes:

- Per-channel and per-pixel clamping of latent values
- Latent-space range normalization
- Lower learning rates (1×10^{-5})

Analysis: The frozen ControlNet produced dense, high-energy features causing latent explosions and mosaic patterns. The Ctrl-Adapter encoder blocks' connection to the SVD latent space was fundamentally incompatible for this task. While Ctrl-Adapter adapts ControlNets to different image models, the dense tile features were incompatible with the sparse edit-propagation task.

Table 2. Analysis of Ctrl-Adapter Features during inference. The observed range consistently exceeds ± 7.7, representing values beyond the expected latent distribution ($\mathcal{N}(0, 1)$). This confirms the Variance Explosion causing VAE collapse.

Step	Min Value	Max Value	Mean
0	-8.1920	7.7523	-0.1299
1	-8.1927	7.7529	-0.1299
2	-8.1926	7.7529	-0.1299

Figure 4. **Visualizing VAE Collapse.** Left: The original input frame. Right: The generated output exhibiting "mosaic" artifacts and color degradation. This visual corruption correlates directly with the latent variance explosion ($> 7\sigma$) shown in Table 2, confirming that the adapter features overwhelmed the VAE decoder.

5.2.3. Control Adapter without Attention Layers

In this experiment, we removed attention layers from the injection mechanism, using pure convolutional layers to try and achieve better computational efficiency:

- Input Hint Block: 4→320 channels, 4× downsampling
- 2× ControlNet Blocks: Spatial Conv2D + Temporal Conv3D
- Zero Conv output layer
- Output shape: $[B * T, 320, H/4, W/4]$

But the results with this approach weren't qualitatively up to the mark and had similar issues as the Ctrl-Adapter with Attention layers.

Figure 5. **Qualitative results of SCE based on Ctrl-Adapter without Attention layers.** Left: The original input frame. Right: The generated garbled output.

5.2.4. Final Approach: ControlNet-Based SCE (Final Approach)

Our final approach uses a ControlNet-inspired SCE that reuses the UNet's down blocks. This provides better com-

patibility with the SVD latent space while maintaining parameter efficiency.

Key Differences when compared to Ctrl-Adapter:

- SCE directly reuses UNet down blocks, ensuring architectural compatibility
- Features are extracted from the same latent space as the UNet operates on
- Injection layers use simple 3D convolutions without attention, reducing complexity
- Training is more stable due to shared architecture between SCE and UNet

6. Results and Discussion

Due to significant constraints on available GPU computational resources, we were unable to train the model to full convergence. Our fine-tuning was limited to a maximum of 17 Epochs. Consequently, our analysis focuses on the training trajectory and the validity of the learning signal rather than achieving state-of-the-art absolute metrics.

6.1. Evaluation Metrics

To quantitatively assess the model's performance, we employed the following metrics:

- **L1 Mask (`l1_mask`):** Measures pixel-wise difference *inside* the edited region.
- **LPIPS Mask (`lpips_mask`):** Measures perceptual similarity. Lower scores indicate more realistic textures.
- **SSIM (`ssim_outside`):** Structural Similarity calculated *outside* the edit (background). Higher scores indicate better background preservation.
- **PSNR (`psnr_outside`):** Peak Signal-to-Noise Ratio calculated *outside* the edit.
- **CLIP (`clip_score`):** Semantic similarity between generated and ground truth frames.

6.2. Quantitative Results

Table 3 summarizes the mean performance metrics across the validation set.

Table 3. Evolution of Metrics over 17 Epochs. Arrows (↓ / ↑) indicate whether lower or higher values are better. Best results are **bold**. We also include the metrics of GenProp from their paper.

Ep.	L1 ↓	LPIPS ↓	SSIM ↑	PSNR ↑	CLIP ↑
1	1.06	0.51	0.08	5.12	0.78
6	1.11	0.52	0.03	4.80	0.79
8	1.07	0.50	0.08	5.30	0.80
10	1.02	0.48	0.11	5.90	0.81
15	1.02	0.47	0.10	5.90	0.81
17	**0.98**	**0.46**	**0.14**	**6.44**	**0.82**
GenProp(Paper)	-	-	-	33.837	0.9825

6.3. Qualitative Results

Figure 6 visualizes the qualitative performance of our model at Epoch 17. The figure displays the **Edited First Frame** (Left), which serves as the condition for the GenProp model. The **Original Video** frames (Top Right) show the ground truth sequence, while the **Propagated Output** (Bottom Right) shows our model's generation.

As observed, the model successfully propagates the semantic edit from the first frame to the subsequent frames, maintaining temporal consistency for the edited object. However, consistent with our quantitative findings (specifically the low PSNR/SSIM scores), the background details in the propagated frames show slight variations compared to the original video, reflecting the limited training time of 17 epochs.

6.4. Comparison with State-of-the-Art

We compare our model (Epoch 17) against baselines reported in the GenProp paper, including SAM + ProPainter and ReVideo. We utilize the CLIP-I score as the primary metric.

Table 4. Comparison of Object Removal (CLIP-I Score). Baselines mentioned are from GenProp paper.

Method	CLIP-I Score (↑)
ReVideo	0.9728
SAM + ProPainter	0.9809
GenProp (Paper)	**0.9879**
GenProp (Ours - Epoch 17)	0.8170

6.5. Analysis and Conclusion

Despite limited training, the results demonstrate a clear learning signal.

First, we observe consistent improvement in semantic metrics (Table 3). The CLIP Score improved steadily from 0.78 to 0.82, suggesting the model is learning to align generated content with the semantic context. Similarly, LPIPS dropped from 0.51 to 0.46, indicating edits are becoming more realistic.

Second, the comparison in Table 4 highlights the impact of training time. While there is a gap between our model (0.82) and the fully converged GenProp model (0.99), our score demonstrates that the model has begun to capture the semantic requirements of the task.

Finally, regarding background preservation, the PSNR (6.44 dB) and SSIM (0.14) remain low. This indicates that at Epoch 17, the Selective Content Encoder (SCE) has not yet converged to perfectly "copy" the background. However, the jump in SSIM at Epoch 17 ($\sim$30% improvement)

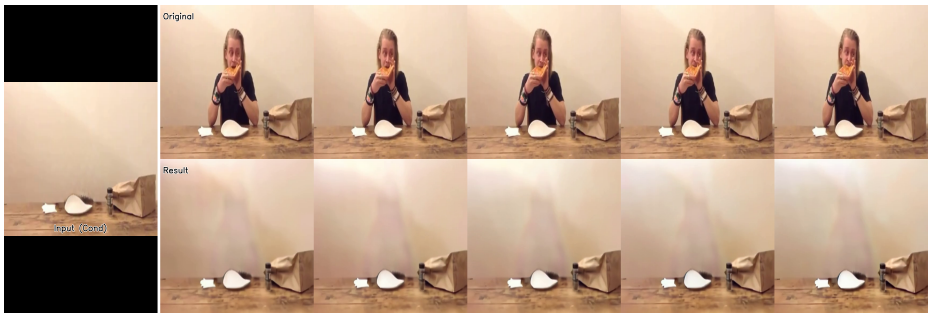

Figure 6. **Qualitative Results at Epoch 17.** The *Edited First Frame* (Left) is propagated through the video sequence. The Top Right row shows the *Original Video* frames, and the Bottom Right row shows *Our Result*. While the edit is propagated semantically, the background exhibits artifacts due to limited fine-tuning.

suggests the background preservation mechanism was beginning to take effect.

Observations: The metrics confirm the GenProp architecture is functional. While 17 epochs were insufficient for high-fidelity background preservation, the strong improvements in LPIPS and CLIP scores demonstrate the model successfully learned semantic editing even within a constrained compute budget.

7. Limitations and Future Work

Our method has several limitations that suggest directions for future research. First, inference requires 25 diffusion steps, precluding real-time applications. Distillation could reduce this to 1–4 steps while preserving quality, using our frozen backbone as the teacher model in a hybrid objective that combines consistency loss with quality rewards from T2V-Turbo metrics for few-step inference. Second, computational constraints prevented full implementation of GenProp's SCE fine-tuning, leaving room for performance improvements. Third, the method's performance is sensitive to mask quality, with imprecise segmentations causing boundary artifacts. Learned mask refinement or soft confidence masks could improve robustness.

Promising future extensions include:

1. Architectural Modifications: Mask-aware gating, input corruption, sparse-feature ControlNets, and connecting adapter layers to the decoder.
2. Multi-Keyframe Conditioning: For temporally-varying edits.
3. Transformer-Based SCE Architectures: For improved long-range modeling.

8. Conclusion

In this work, we present a reproduction and analysis of the Generative Video Propagation (GenProp) framework, adapted for a frozen Stable Video Diffusion (SVD) backbone. By implementing a custom Selective Content Encoder (SCE) and a Mask Prediction Decoder (MPD), we validated that generative priors from image-to-video models can be effectively repurposed for video object removal. Our implementation demonstrates that region-aware supervision and explicit temporal guidance are essential for maintaining visual consistency across dynamic scenes.

Despite limited training (17 epochs), the model shows consistent improvements across consecutive epochs on CLIP, LPIPS, and SSIM metrics, validating the approach. Experiments highlight the importance of architectural compatibility and careful design choices, such as zero-initialized injection layers and region-aware losses, for stable training and effective background preservation. We thus demonstrate an architectural baseline for future work on distillation for few-step inference and efficient training.

References

[1] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach. Stable video diffusion: Scaling latent video diffusion models to large datasets, 2023. 2, 4

[2] Haoxin Chen et al. Videocrafter: A toolkit for text-to-video generation and editing. *arXiv preprint arXiv:2309.03871*, 2023. 2

[3] Alexei A Efros and William T Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 341–346. ACM, 2001. 2

[4] Jonathan Ho et al. Video diffusion models. In *NeurIPS*, 2022. 2

[5] Alexander Kirillov et al. Segment anything. In *ICCV*, 2023. 2

[6] Vladimir Kulikov, Matan Kleiner, Inbar Huberman-Spiegelglas, and Tomer Michaeli. Flowedit:

Inversion-free text-based editing using pre-trained flow models, 2025. 2

[7] Jiachen Li, Weixi Feng, Tsu-Jui Fu, Xinyi Wang, Sugato Basu, Wenhu Chen, and William Yang Wang. T2v-turbo: Breaking the quality bottleneck of video consistency model with mixed reward feedback, 2024.

[8] Han Lin, Jaemin Cho, Abhay Zala, and Mohit Bansal. Ctrl-adapter: An efficient and versatile framework for adapting diverse controls to any diffusion model, 2024. 2

[9] Shanchuan Lin, Xin Xia, Yuxi Ren, Ceyuan Yang, Xuefeng Xiao, and Lu Jiang. Diffusion adversarial post-training for one-step video generation, 2025.

[10] Shaoteng Liu, Tianyu Wang, Jui-Hsien Wang, Qing Liu, Zhifei Zhang, Joon-Young Lee, Yijun Li, Bei Yu, Zhe Lin, Soo Ye Kim, and Jiaya Jia. Generative video propagation, 2024. 1, 2, 3

[11] Shaoteng Liu et al. Video-p2p: Video editing with cross-attention control. In *CVPR*, 2024. 2

[12] Hao Ouyang et al. Codef: Content deformation fields for temporally consistent video processing. In *CVPR*, 2024.

[13] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation, 2018. 3

[14] Richard Szeliski. A survey of optical flow techniques. *arXiv preprint arXiv:1409.5184*, 2014. 2

[15] Jianzong Wu, Xiangtai Li, Chenyang Si, Shangchen Zhou, Jingkang Yang, Jiangning Zhang, Yining Li, Kai Chen, Yunhai Tong, Ziwei Liu, and Chen Change Loy. Towards language-driven video inpainting via multimodal large language models, 2024. 3

[16] Jay Zhangjie Wu et al. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *ICCV*, 2023. 2

[17] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 2

[18] Bowen Zheng and Tianming Yang. Revisiting diffusion models: From generative pre-training to one-step generation, 2025.

[19] Shangchen Zhou, Chongyi Li, Kelvin C. K. Chan, and Chen Change Loy. Propainter: Improving propagation and transformer for video inpainting, 2023. 2