Video Inpainting with Region-Aware Propagation

Akshata Kumble
Amit Karanth Gurpur
Raghav Jadia
Rituraj Navindgikar

AGENDA

- Goal of project
- What are we trying to solve
- What was the approach
- What was needed
- What was SOTA/ what is novel
- How we plan on evaluating, metrics, datasets we plan to use
- Preliminary results
- Current Progress
- Future Scope

OUR PROJECT GOALS

Goal:
Develop a video object-removal system that propagates a single edited keyframe across an entire video with temporal consistency, inspired by GenProp with experiments for efficiency using adapters, and lightweight propagation modules.

Key Objectives:

- Enable object removal across full videos from one edited keyframe.
- Deliver temporally consistent results with no flicker or drift.
- Experiment with reducing training cost via SVD backbone + Control Adapters.

WHAT WE ATTEMPT TO SOLVE

The core problem:

Traditional video object removal is unreliable and slow.

Most tools use frame-by-frame inpainting or iterative diffusion, leading to:

- Temporal drift (object edges shift over time)
- Flicker (inconsistency between frames)
- High computational cost
- Handling occlusions, fast motion, and shape/depth changes.

We want to solve this by:

- Using the original video via SCE-style encoding+MPD.
- Applying region-aware loss to ensure clean, consistent propagation.
- Experiment with approaches for reducing computational cost for training/finetuning

PAPER RELEVANCE

GenProp introduced: Selective Content Encoder (SCE), Mask Prediction Decoder (MPD), and Region-Aware (RA) Loss for high-fidelity propagation.

Limitations of GenProp (why EfficientEdit must extend it):

- No official code release → hard to reproduce
- Some parameters, constraints, and training details are under-specified
- Multi-step sampling → not suited for fast or interactive workflows

We explored ways to extend GenProp by incorporating:

- Control Adapters (parameter-efficient conditioning).
- One-step generation process

I2V AND DATASETS

Image-to-Video Models:

1. SkyReels
2. VideoCrafter 1 & 2
3. Stable Video Diffusion - XT (Stability AI)

Datasets:

1. DAVIS (Densely Annotated Video Segmentation)
2. ROVI (Real-world Object Video Inpainting)

INITIAL CUSTOM 3D U-NET WITH SCE

Built a very small 3D U-Net that processes videos as 5D tensors [B, C, T, H, W]

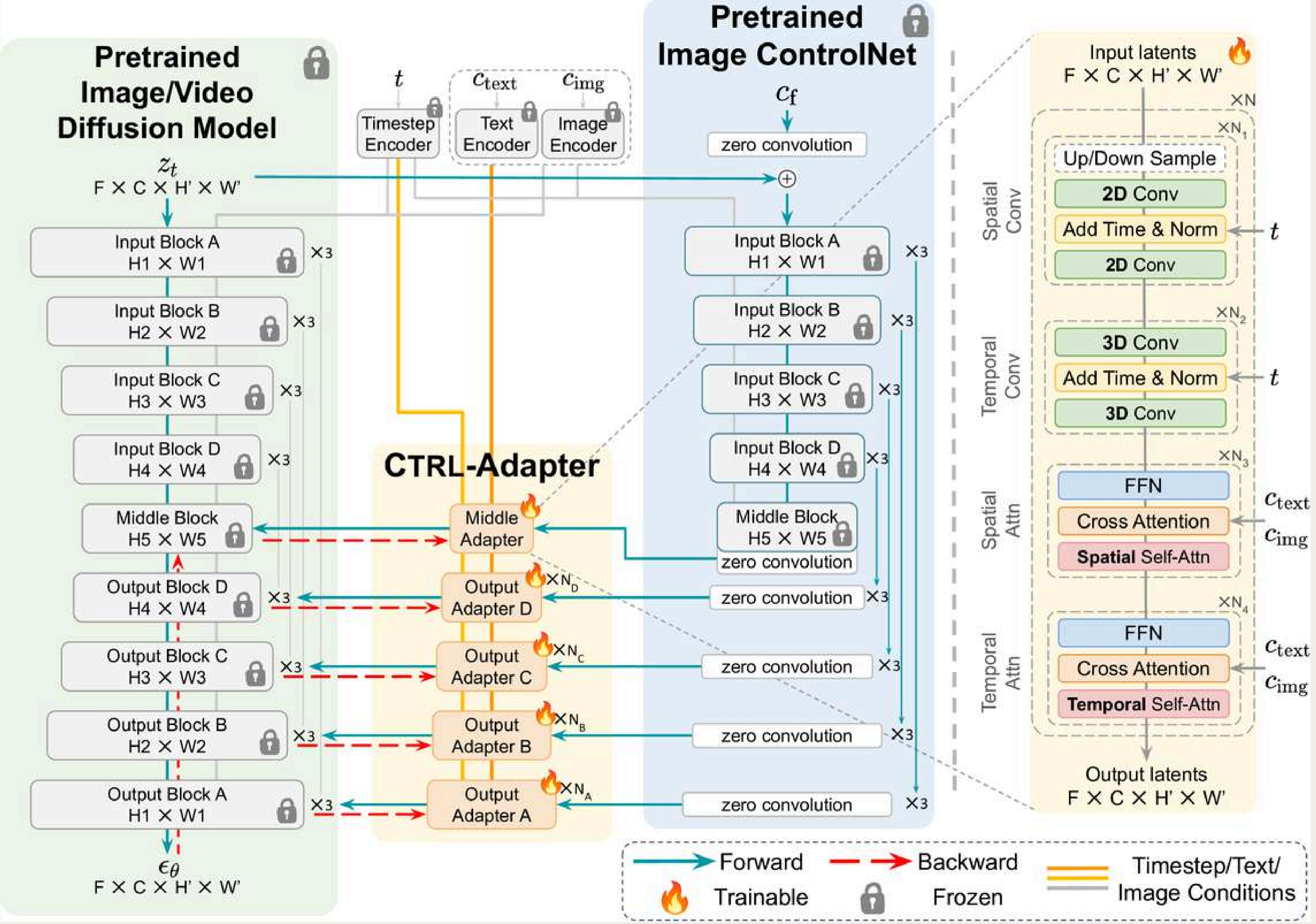
Added a Selective Content Encoder (SCE) using the first edited frame

Used region-aware loss to supervise mask vs non-mask areas

Experiments with Ctrl-Adapter

EXPERIMENTS WITH CTRL-ADAPTER

- GenProp’s SCE- inspired by ControlNet(copy of encoder layers of I2V model)- Heavy.
- Is there a more lightweight alternative?
- Ctrl-Adapter - let’s you ‘adapt’ pretrained ControlNets to any image/diffusion model.

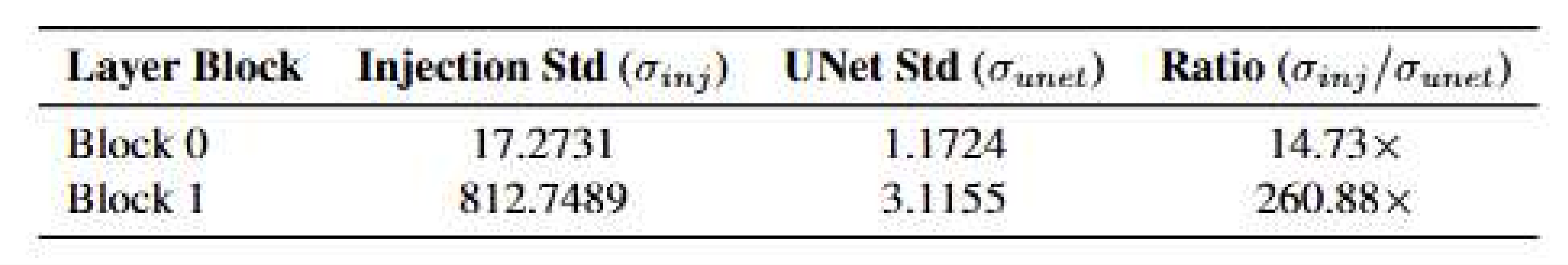


EXPERIMENTS WITH CTRL-ADAPTER

- Combine Ctrl-Adapter with GenProp-
  - Instead of training full ControlNet inspired SCE, why not base SCE on Ctrl-Adapter
  - Keep both I2V model and ControlNet backbone frozen, train only adapter layers+MPD with RA-loss → more computationally efficient and faster.
  - Implement inverse-timestep sampling- algorithm proposed by ctrl-adapter to map continuous timesteps of backbone with discrete timesteps expected by ControlNet.
  - Most ControlNets- for sparse features like canny, depth etc.
  - Chose ControlNet Tile- uses RGB images to increases resolution/reconstruct.

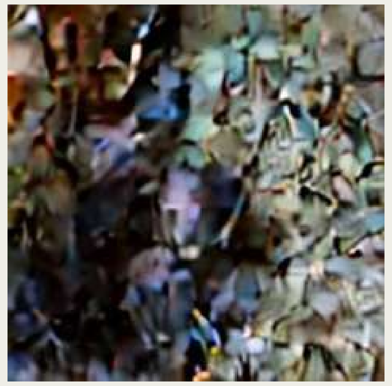
EXPERIMENTS WITH CTRL-ADAPTER

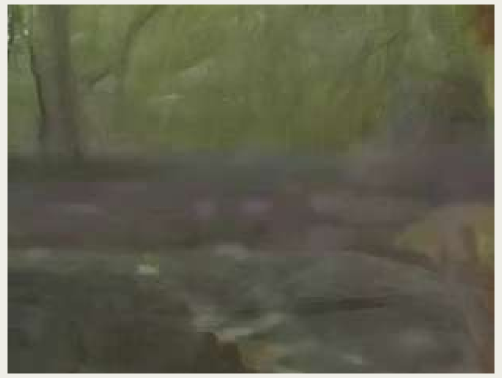
- Issues encountered-
    - Instability in the form of multicolor flashes/mosaics-
        - Variance explosion → VAE collapse
        - Small padding mismatches/Spatial Misalignment → another cause of mosaic patterns/garbled output.
        - Loss and their components- did not reveal failure on their own. Failure visible only in decoded video.
    - Task incompatibility-
        - Dense tile ControlNet features incompatible with SVD latent space.
        - ControlNet Tile → Trained to reconstruct videos.
        - SCE in GenProp → Has to ignore the same objects.



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×

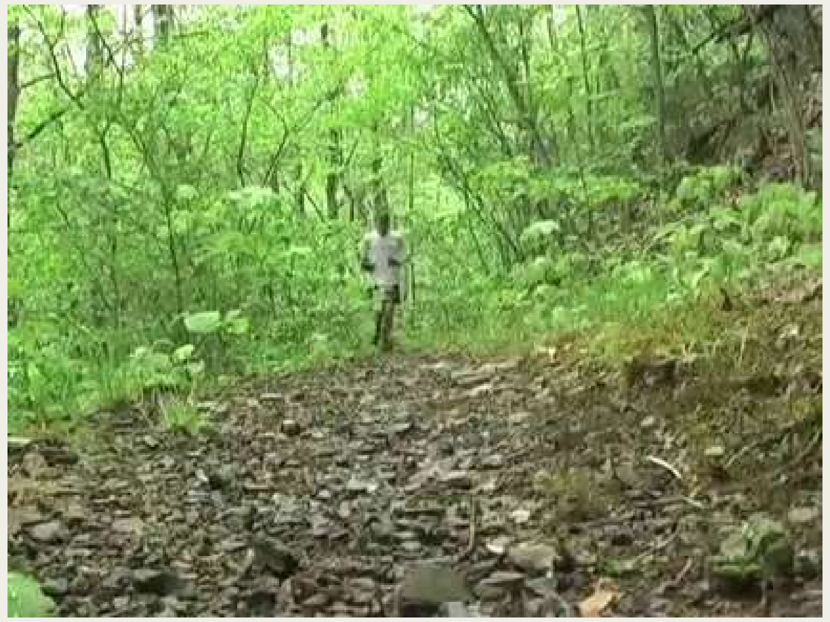
EXPERIMENTS WITH CTRL-ADAPTER



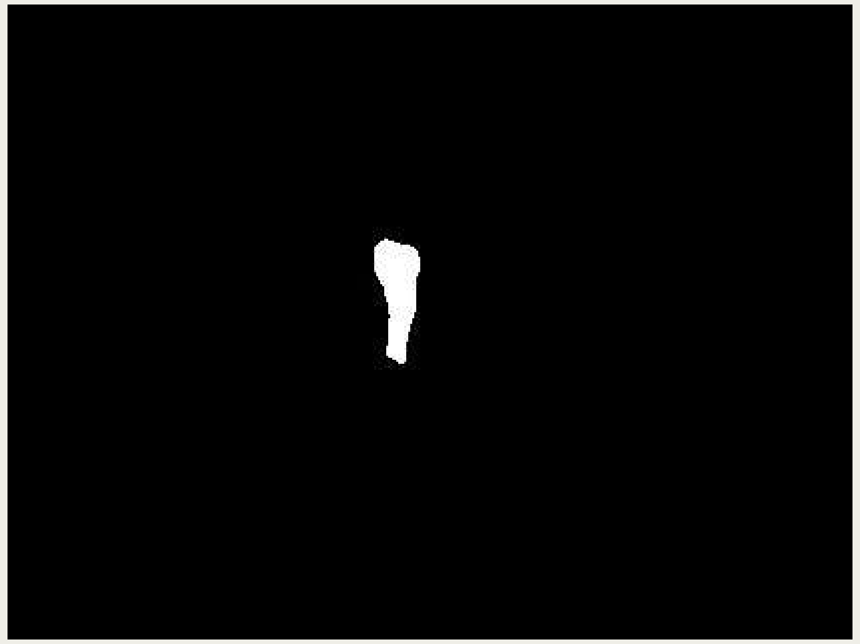


- What we tried to fix the issues-
  - Per-channel/per-pixel clamping
  - Latent-space range normalization
  - Lower LR
- Probable cause of failures-
  - Frozen ControlNet produces dense, high-energy features that causes latent explosions, mosaic like patterns etc.
  - Issue with connections- Ctrl-Adapter+encoder blocks- might be incompatible.
- What can be done for further experiments with this-
  - Mask-aware gating of the Ctrl-Adapter based SCE.
  - Input corruption- Blur masked region.
  - Switch to other pretrained ControlNets such as ControlNet Canny, ControlNet Depth
  - Switch to connecting adapter layers to decoder like in SVD.
- What we are currently doing- switch to normal ControlNet based SCE for GenProp.
- Simultaneously explore few-step/one-step editing.

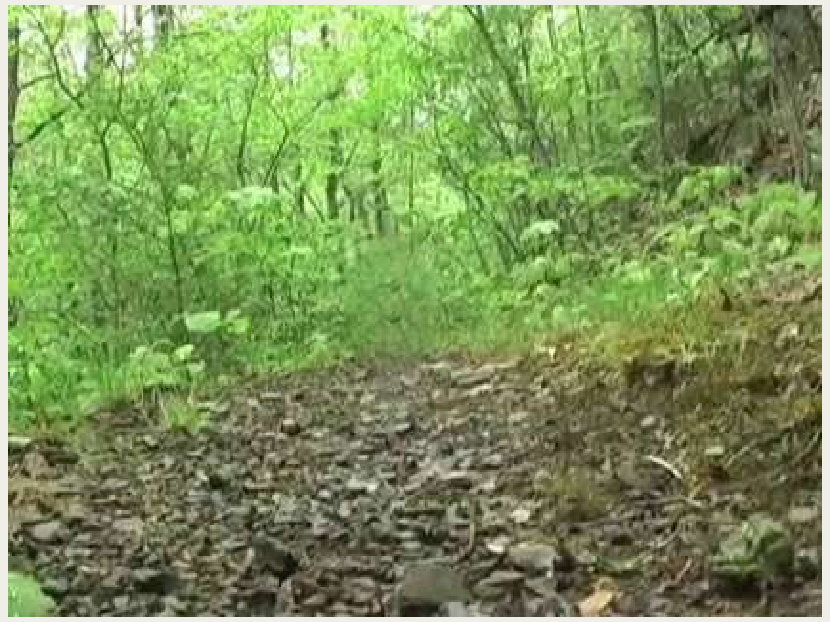
EXPERIMENTS WITH CTRL-ADAPTER



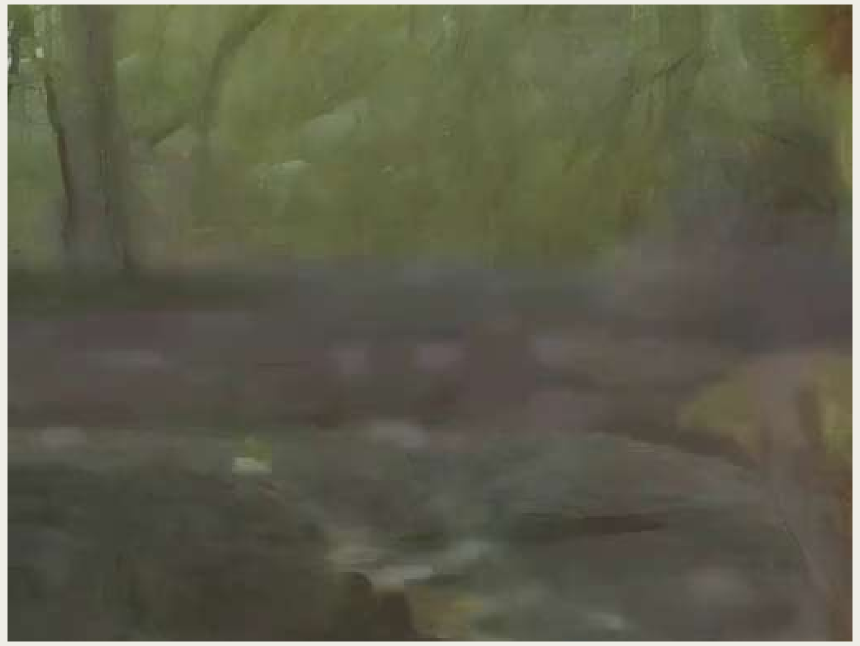
Frame from Original Video.
Original Video fed to SCE



Ground truth mask.
Used for MPD during Training



Frame from ground truth inpainted video

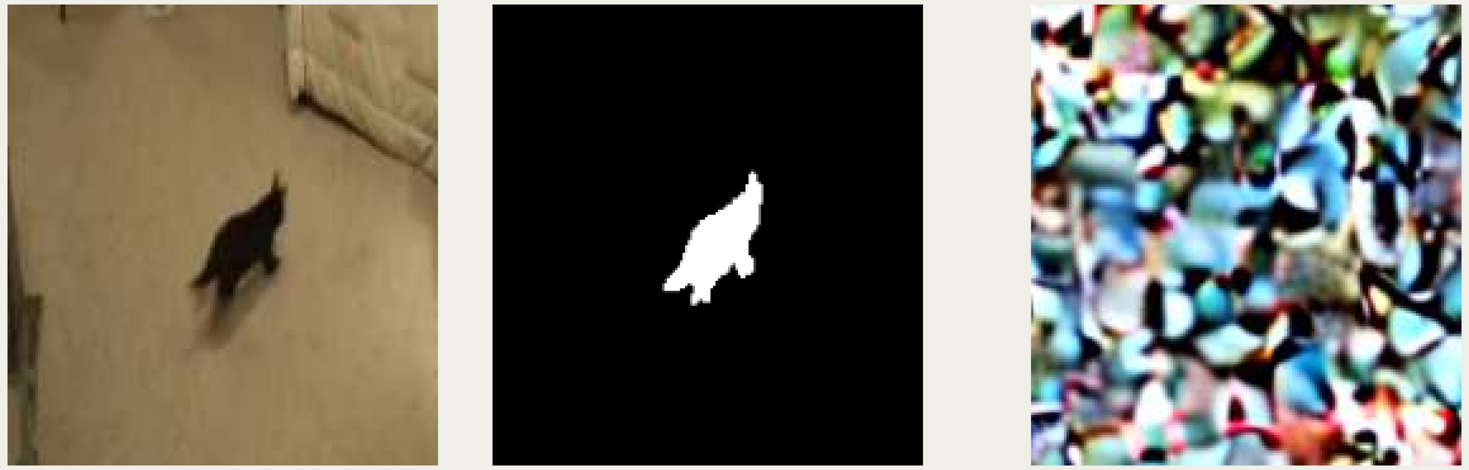


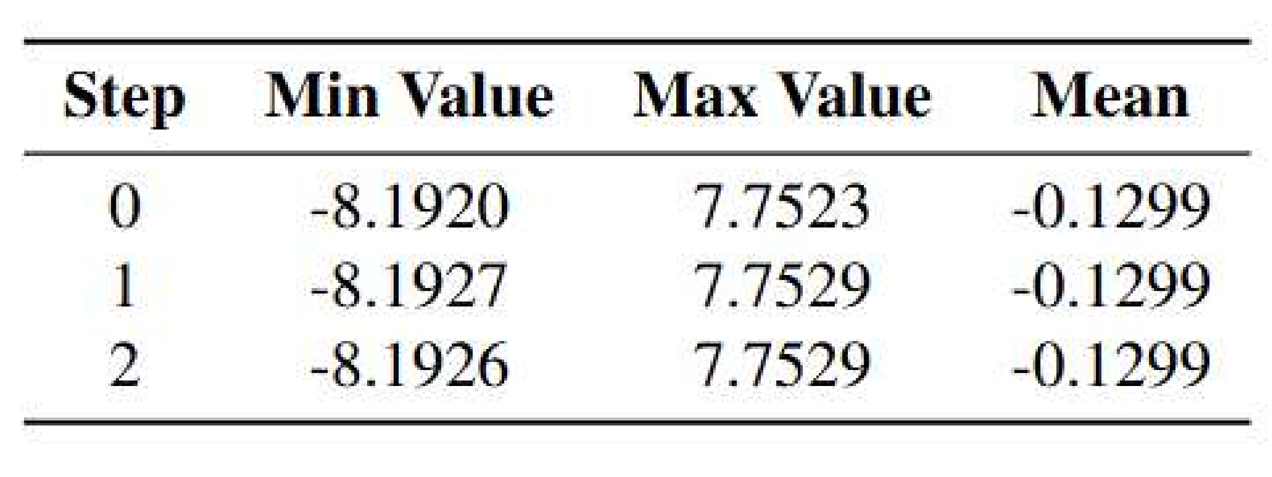
Current first frame result(mosaic/blurry) generated.
Used Ctrl-Adapter SCE(same adapter architecture as Ctrl-Adapter in Repo)

Control Adapter w/o attention layers

LATENT VIDEO PIPELINE

- Compress RGB frames into latent space, encoding into 4-channel latents using a VAE
- Spatial resolution drops significantly but the visual information is preserved
- The edited first frame is encoded separately using CLIP, giving us a semantic understanding of what the edit looks like at conceptual level
- Add edited video latent with noise and combine it with clean conditioning frame creating 8 channel input for UNet





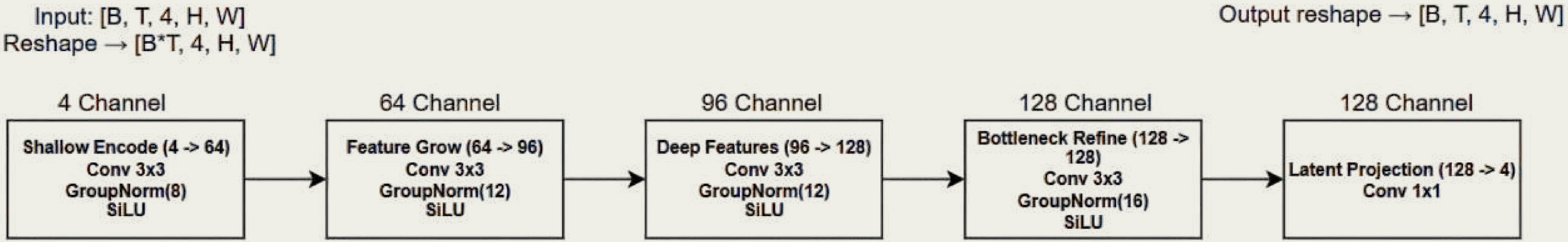
Step	Min Value	Max Value	Mean
0	-8.1920	7.7523	-0.1299
1	-8.1927	7.7529	-0.1299
2	-8.1926	7.7529	-0.1299

RGB videos + Mask Videos → VAE latents → noisy targets + conditioning latents → UNet input

[B, T, 3, 320, 192]

[B, T-1, 8, 40, 24]

SELECTIVE CONTENT ENCODER



Input: [B, T, 4, H, W]
Reshape → [B*T, 4, H, W]

Output reshape → [B, T, 4, H, W]

4 Channel

Shallow Encode (4 -> 64)
Conv 3x3
GroupNorm(8)
SiLU

64 Channel

Feature Grow (64 -> 96)
Conv 3x3
GroupNorm(12)
SiLU

96 Channel

Deep Features (96 -> 128)
Conv 3x3
GroupNorm(12)
SiLU

128 Channel

Bottleneck Refine (128 -> 128)
Conv 3x3
GroupNorm(16)
SiLU

128 Channel

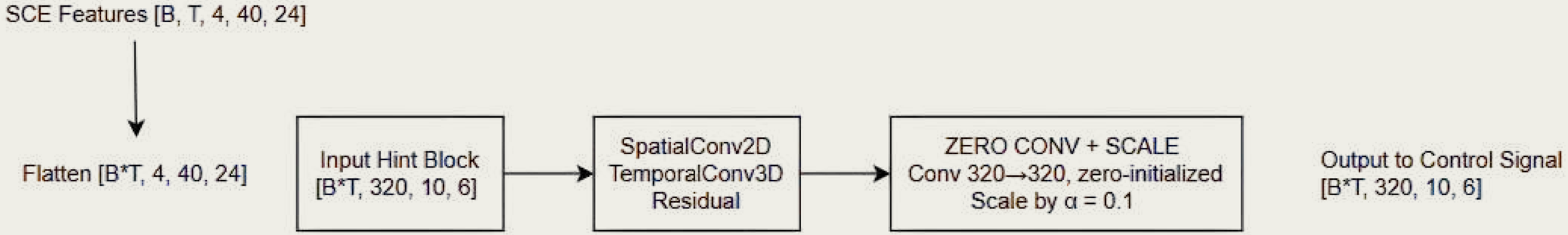
Latent Projection (128 -> 4)
Conv 1x1

- Lightweight Conv–GroupNorm–SiLU stack applied per frame to refine VAE latents
- Preserves shape [B, T, 4, H, W], but improves separation of foreground/background and structure
- Provides a clean content representation used by both the control branch and the mask head

B - Batch size
T - Time steps
4 - Latent channels
H - Latent height
W - Latent Width

CONTROL ADAPTER (SPATIAL + TEMPORAL CONVOLUTION)

- Takes SCE output → generates control signal
- Architecture:
  - Input Hint Block (4→320 channels, downsample 4x)
  - 2× ControlNet Blocks (SpatialConv2D + TemporalConv3D)
  - Zero Conv output
- Output: [B*T, 320, H/4, W/4]
- NO attention layers - pure convolutions



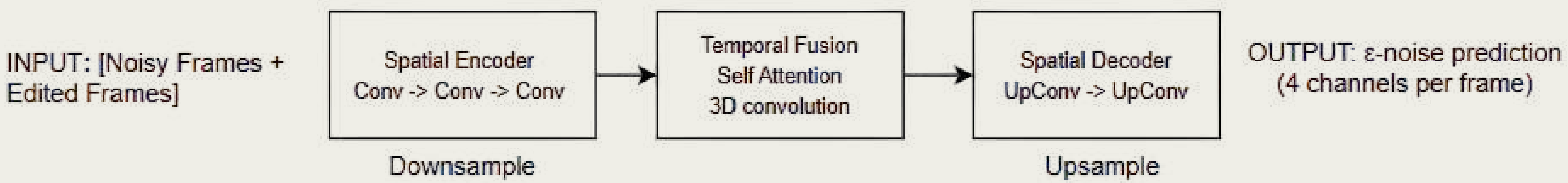
CONTROL INJECTION LAYER

The control signal is injected into the UNet through three steps:

a. Spatial Upsampling: Our Control signal is at reduced resolution, so we use bilinear interpolation to sample it back to match the UNet input size
b. Channel Projection: The control signal has 320 channels, but UNet input only has 8 channels, we use 1x1 convolution to project 320 down to 8 channels
c. Scale and Add: We scale the control signal by 0.1 and add it as a residual to UNet input

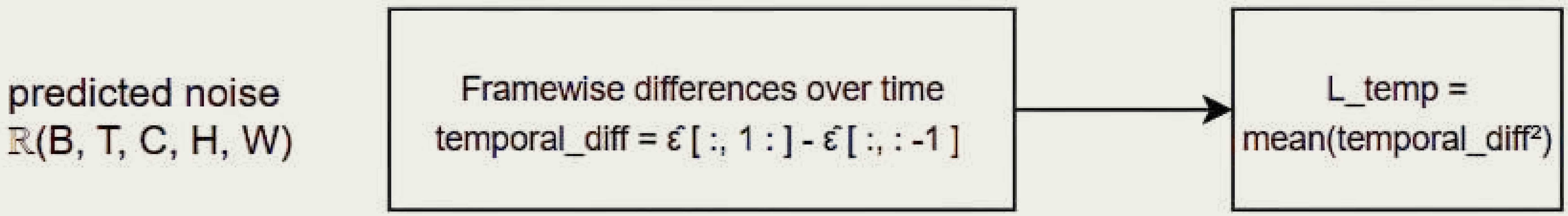
SPATIO-TEMPORAL UNET BACKBONE

- A Stable-Video-Diffusion spatio-temporal UNet denoises latent video sequences.
- Operates on concatenated [noisy targets ⊕ edited first frame] → 8 latent channels, with temporal fusion in mid-blocks.
- Predicts ε-noise for all frames, which is used by the diffusion sampler to reconstruct the edited video.



TRAINING LOSSES

- Temporal Consistency Loss
  - Since the ROVI dataset had all video frames edited so while training we could check for another loss for each frame
  - Penalizes changes in predicted noise between consecutive frames → encourages temporal smoothness in the video



predicted noise $\mathbb{R}(B, T, C, H, W)$

Framewise differences over time

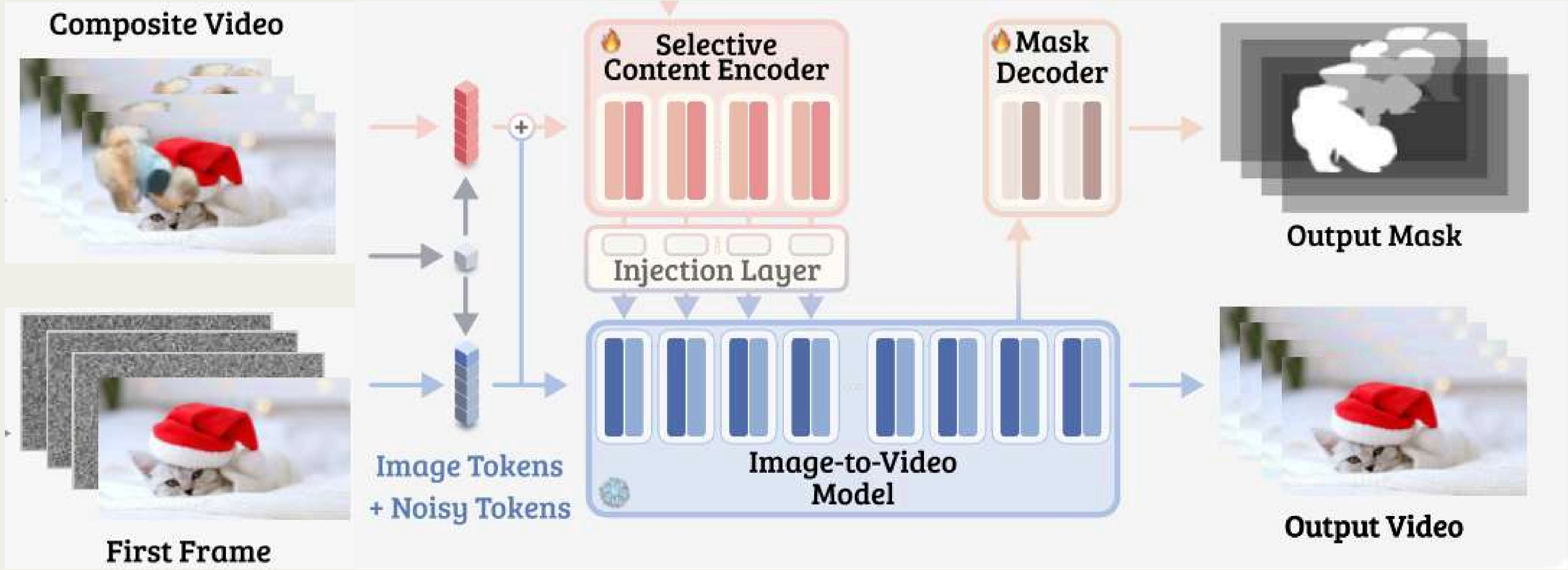
temporal_diff = $\hat{\epsilon}[:, 1:] - \hat{\epsilon}[:, :-1]$

L_temp = mean(temporal_diff^2)

ControlNet Implementation

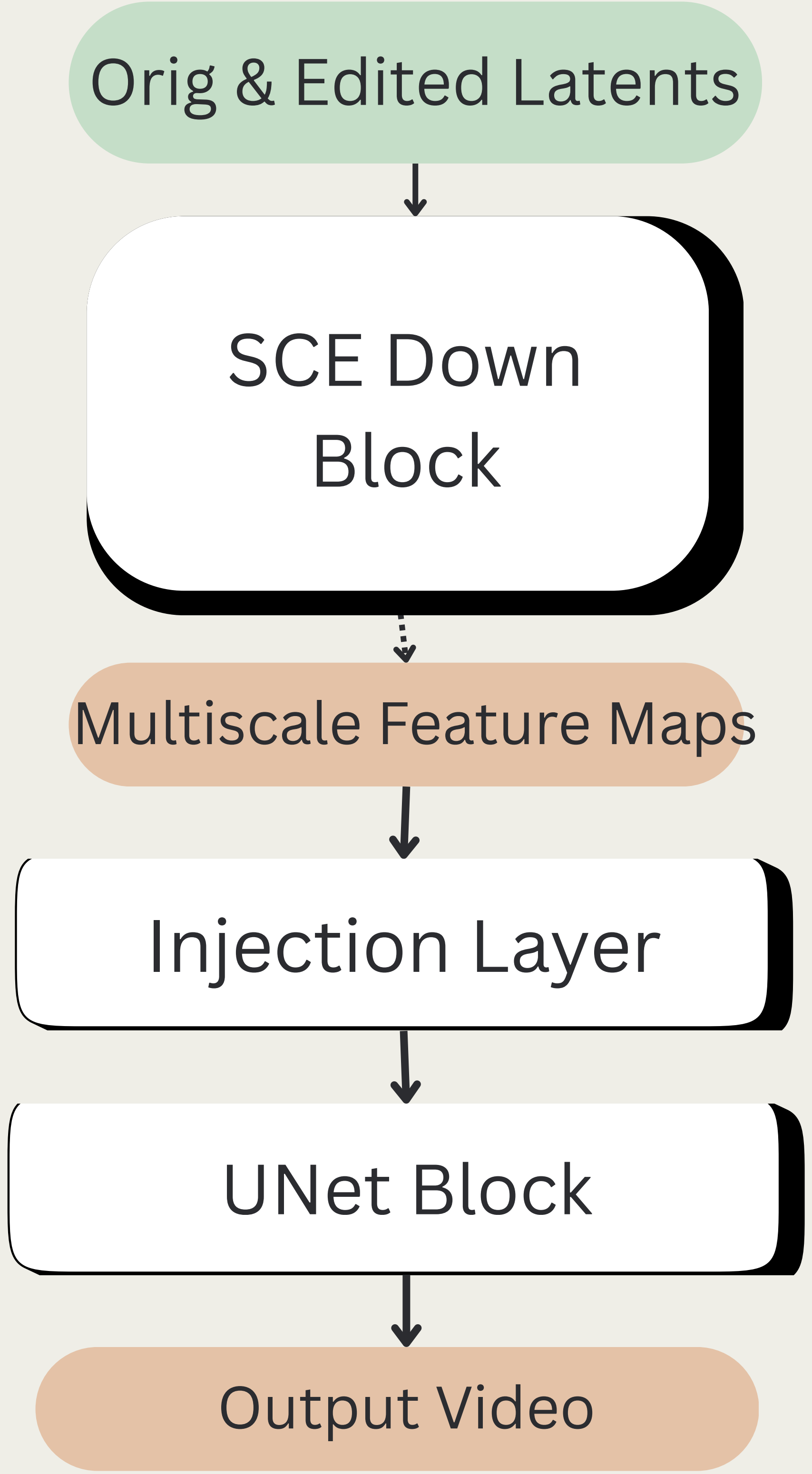
SYSTEM OVERVIEW

- Edit propogation using SVD-XT
- Conditioning using SCE, Injection Layers, MPD
- SCE extracts edit-aware features from (orig || edited) latents.
- MPD predicts edit mask over time.
- Supervised by Region-Aware Loss



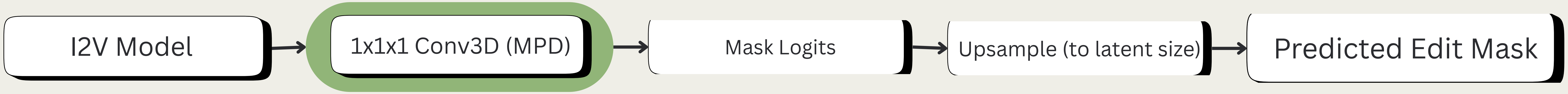
SELECTIVE CONTENT ENCODER

- SCE reuses UNet down blocks
- Produces multi-scale edit-focused features.
- Injection = zero-init 1×1×1 Conv3D Adapter.
- Injects SCE features into matching UNet blocks.
- Enables edit-conditioned denoising while UNet stays frozen.



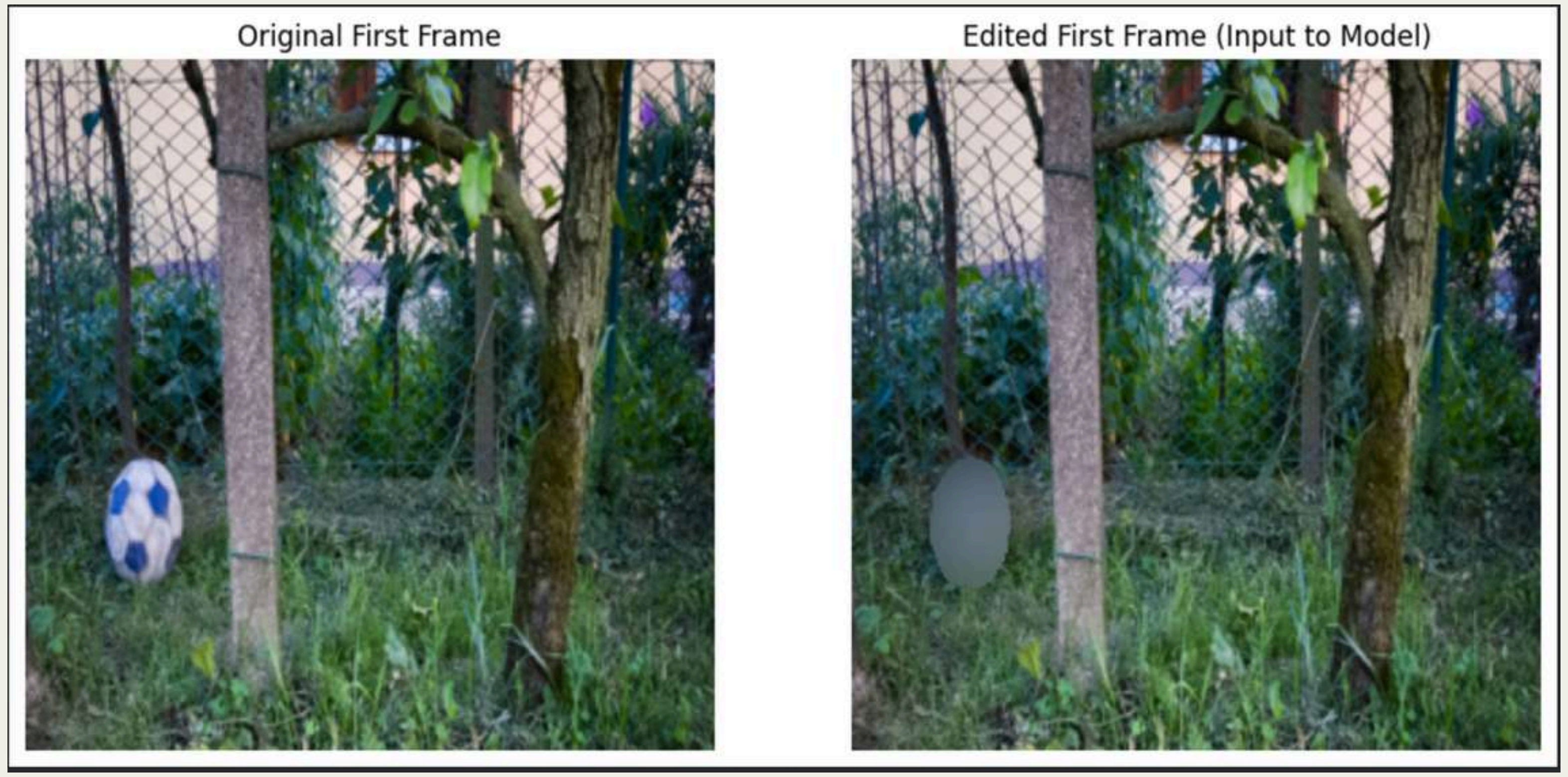
MASK PREDICTION DECODER

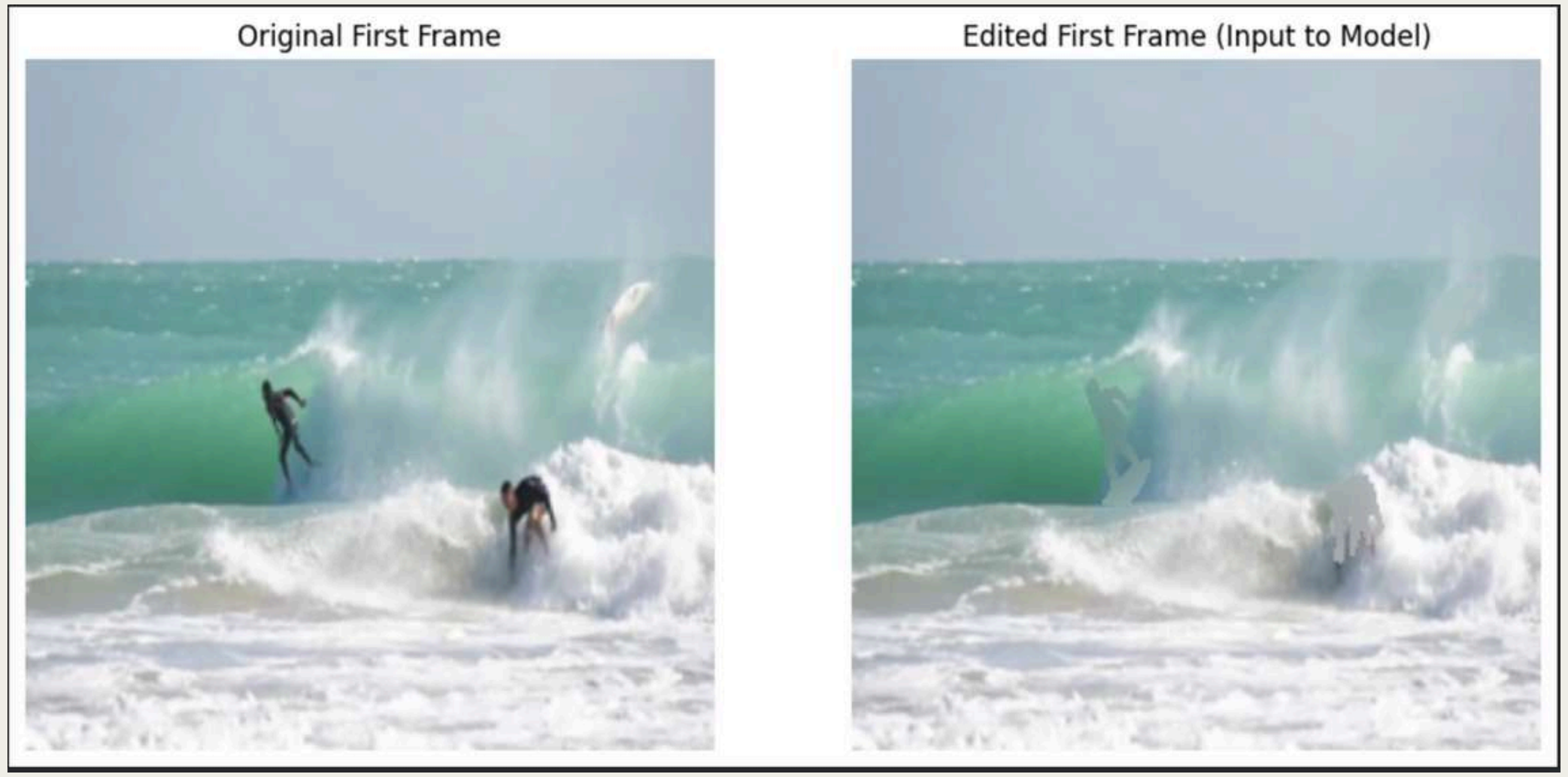
- MPD consumes deep spatiotemporal features from the I2V backbone.
- Predicts a per-frame edit mask to guide region-aware training.
- Provides explicit spatial & temporal supervision to the diffusion model.
- BCE loss between predicted mask & ground-truth masks.



TRAINING PIPELINE

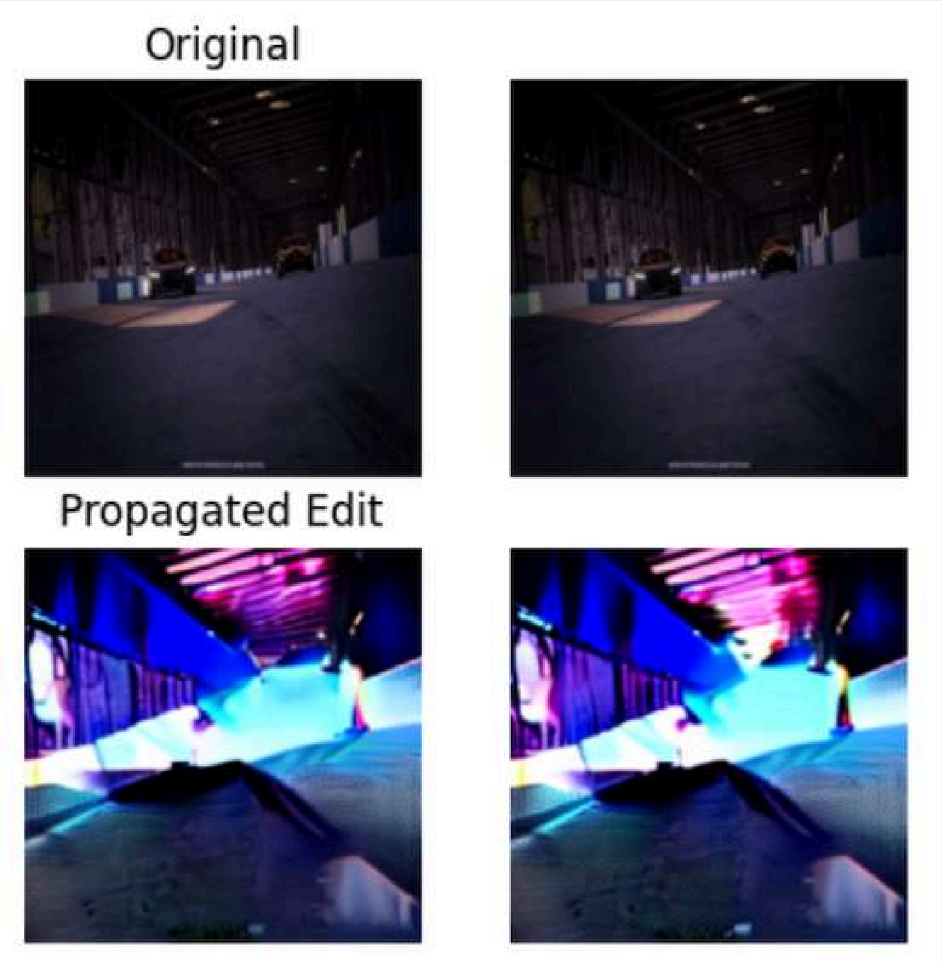
- Frozen: UNet, VAE
- Trainable: SCE, Injection Layers, MPD.
- Input edit: Masked Gaussian blur.
- Loss: Region-Aware (inside/outside) + MPD BCE.
- SCE learns edit structure; injection learns feature modulation.

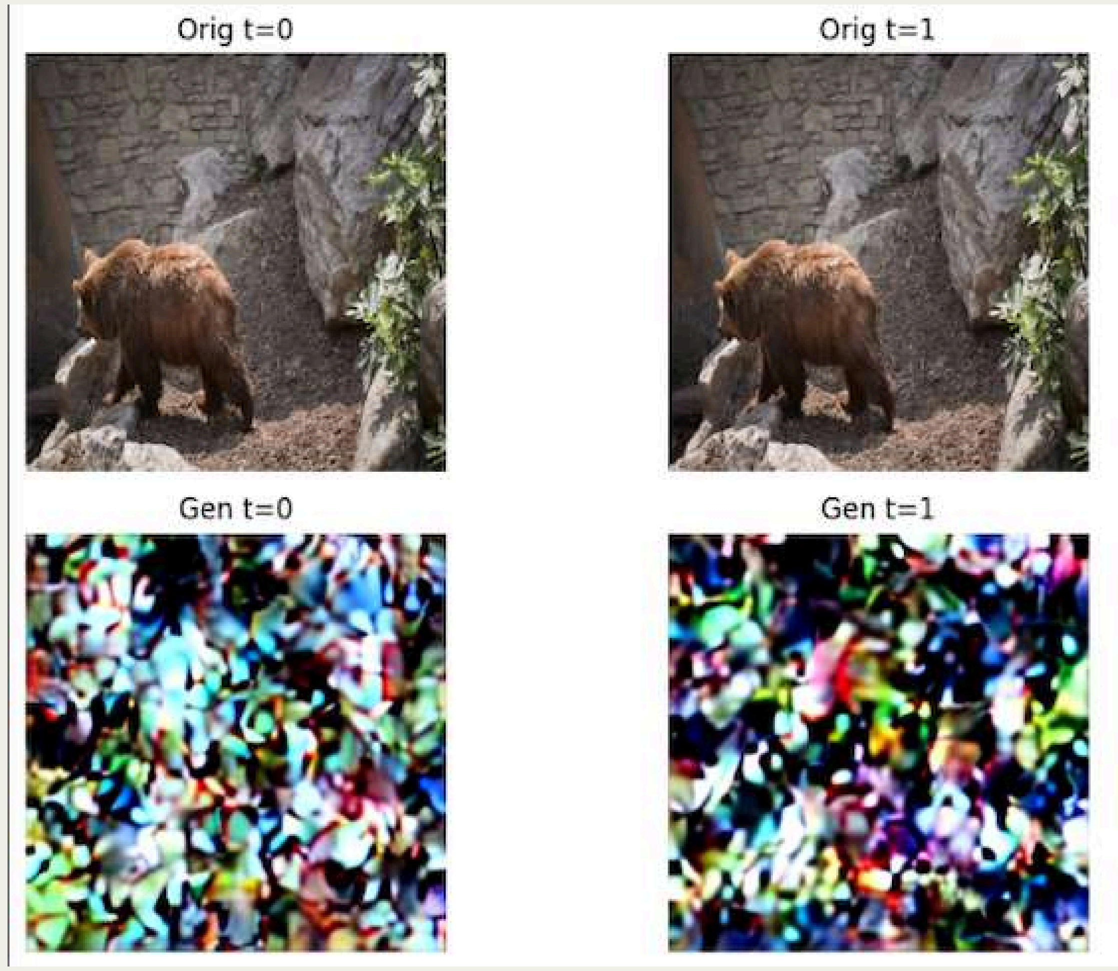




CHALLENGES FACED DURING IMPLEMENTATION

- MPD collapse: solved via stronger BCE weighting
- Clean DAVIS masks: simulated blur-removal edits for controllable supervision.
- Backbone frozen: dependant on the conditioning blocks.



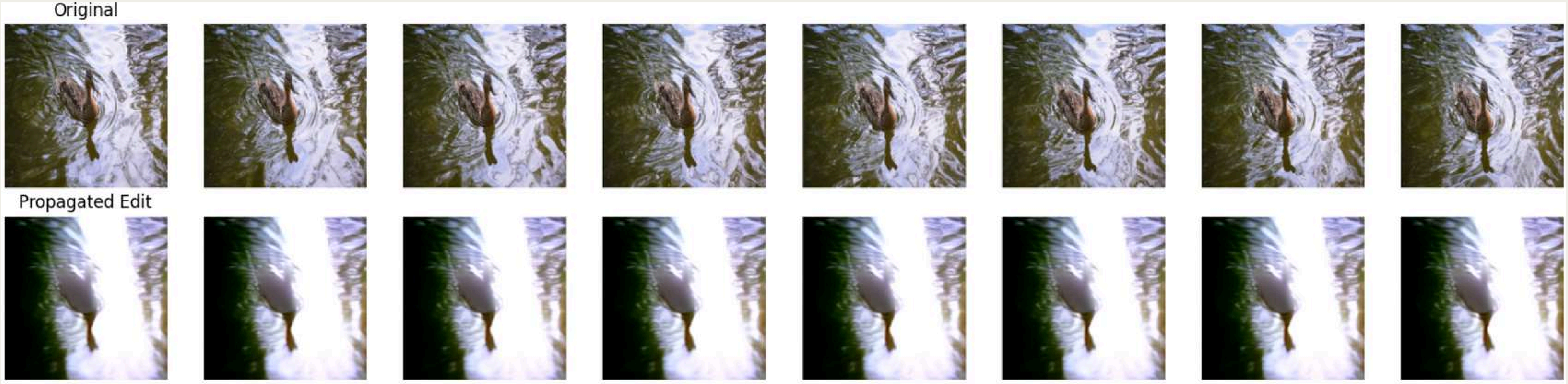


OVERCOMING CHALLENGES

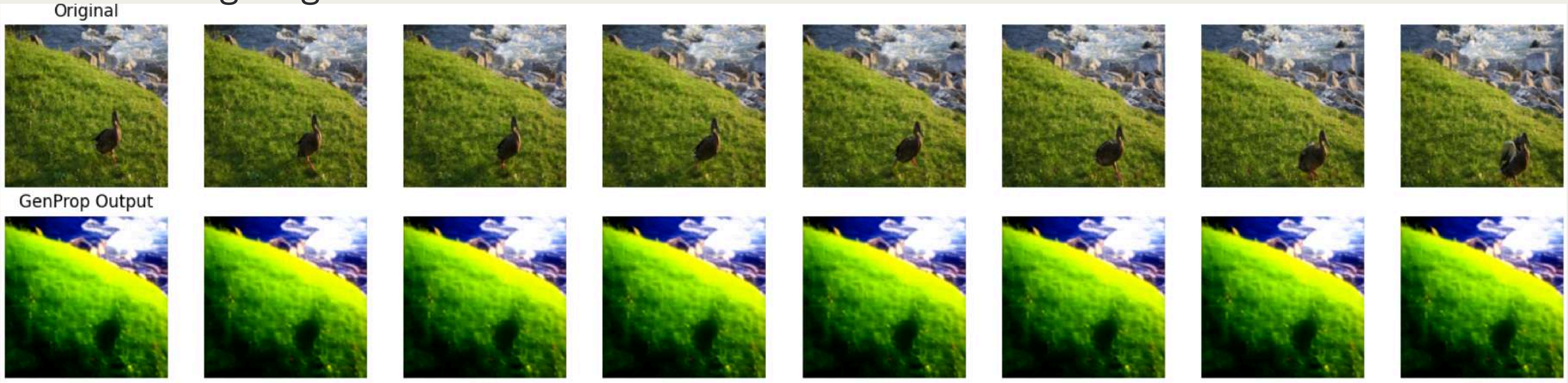
- Use GroupNorm to prevent the latent explosion
- Add clamping before computation of L_grad
- Proper mapping between noise & velocity of SVD-XT

OUTPUT AT THE TIME OF PRESENTATION (2ND DEC)

A duck swimming in a water body

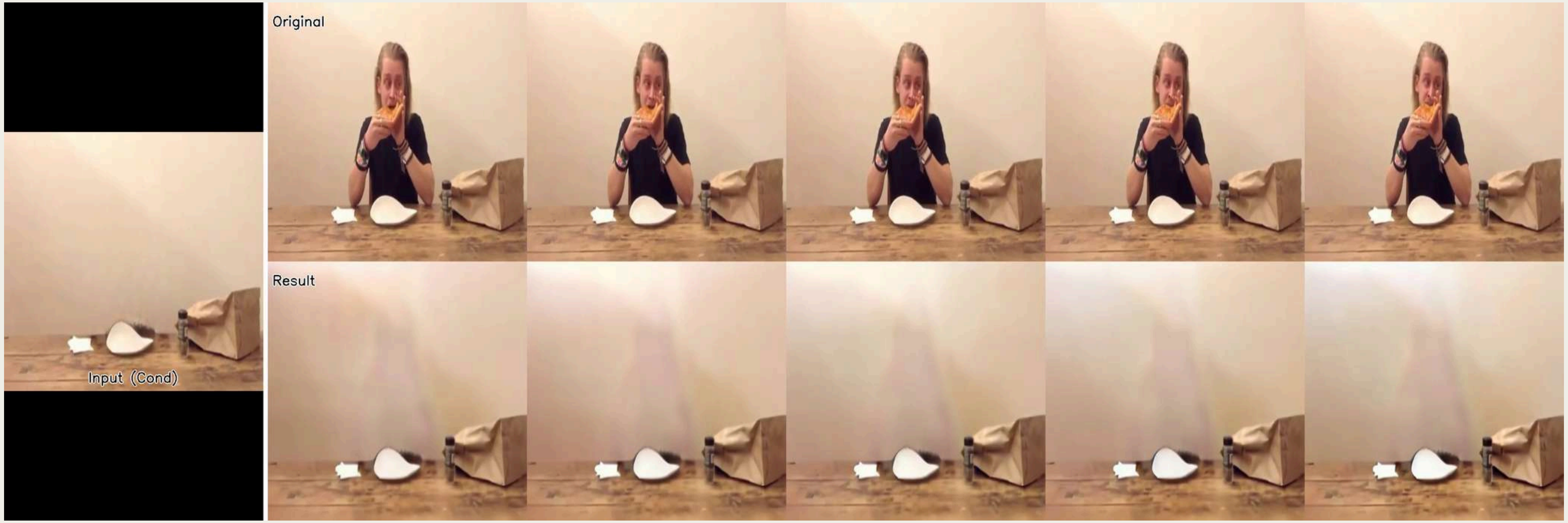


A bird walking on grass



Final Results

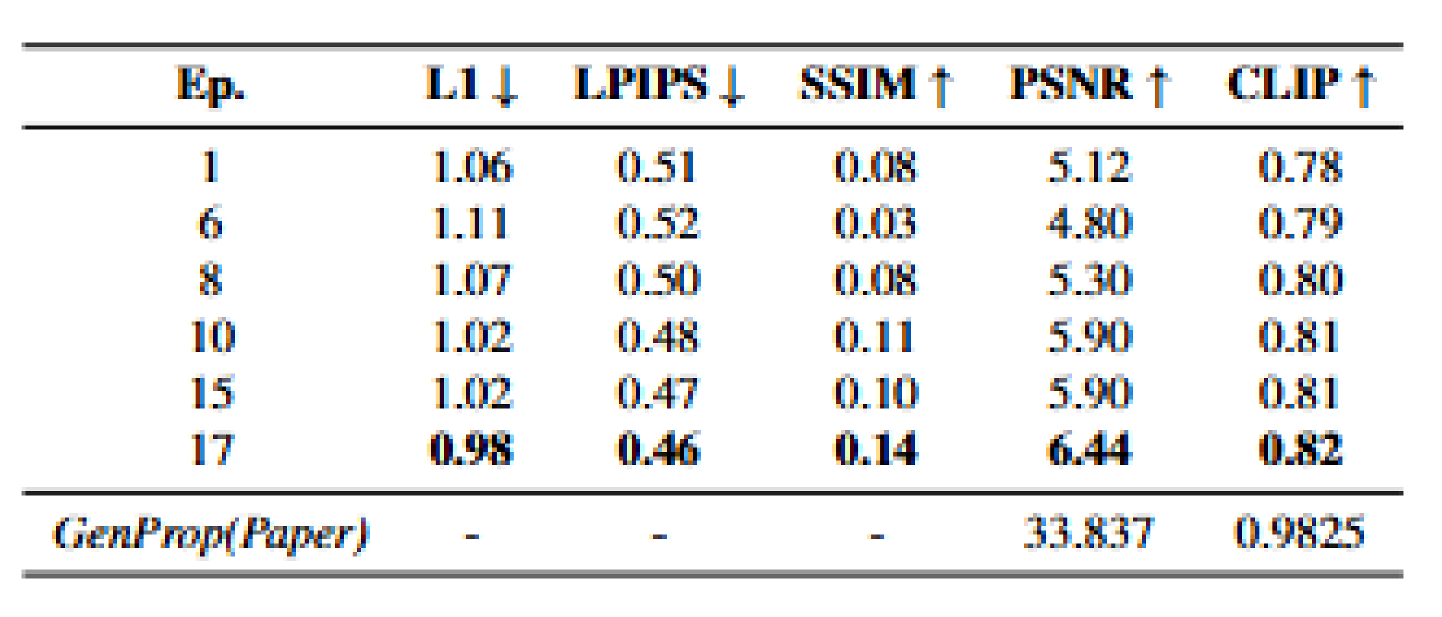
QUALITATIVE FINAL RESULTS(AFTER 17 EPOCHS)



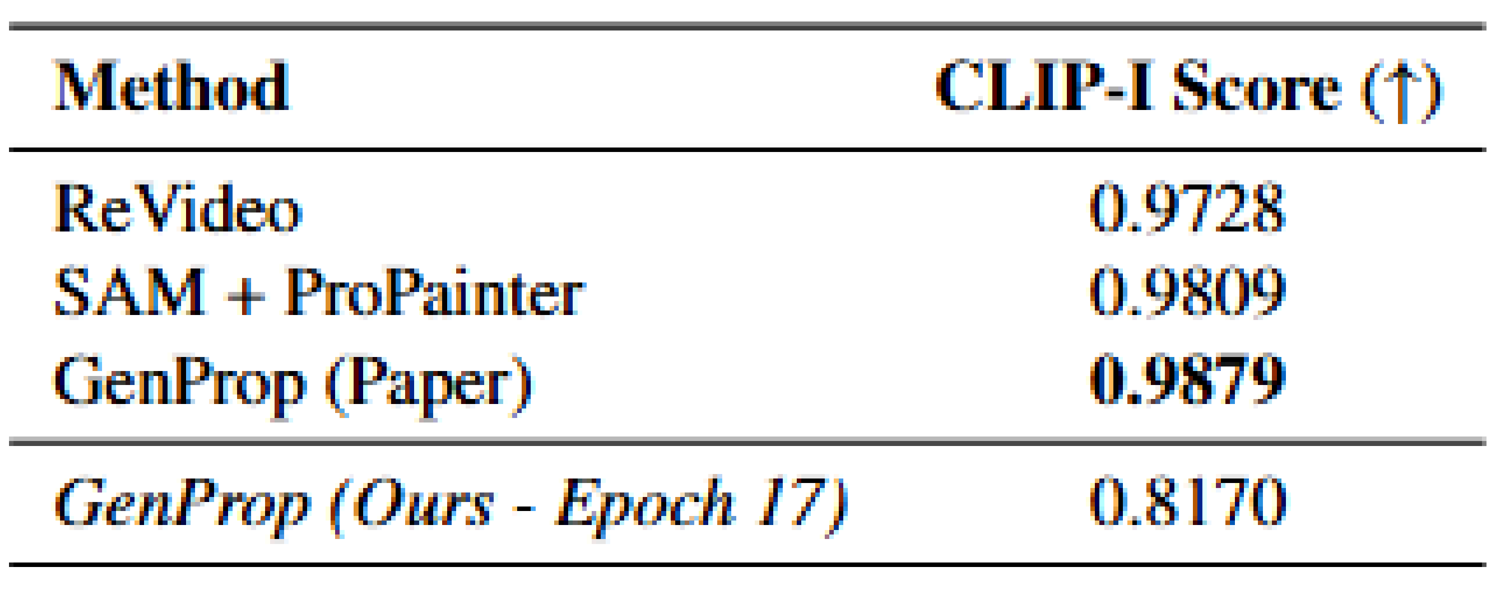
The figure displays the Edited First Frame (Left), which serves as the condition for the GenProp model. The Original Video frames (Top Right) show the ground truth sequence, while the Propagated Output (Bottom Right) shows our model's generation.

While the edit is propagated semantically, the background exhibits artifacts due to limited fine-tuning.

QUANTITATIVE RESULTS



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



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

Semantic alignment improves steadily:

- CLIP score increases from 0.78 → 0.82, showing the model is learning the intended edit meaningfully.

Perceptual quality gets better:

- LPIPS decreases from 0.51 → 0.46, indicating more realistic inpainted textures.

Background consistency starts to emerge late in training:

- SSIM rises from 0.08 → 0.14 (largest jump at Epoch 17, ~30% improvement).
- PSNR increases from 5.12 → 6.44 dB, though still far from convergence.

Masked-region accuracy improves modestly:

- L1 slightly reduces to 0.98, the best value at Epoch 17.

Overall improvement trend is clear despite only 17 epochs, showing a strong learning signal even without full convergence.

Our CLIP-I is lower (0.817 vs. 0.97–0.99 SOTA), largely due to limited training time rather than architectural limitations.

FUTURE PLAN-ACCELERATE GENPROP INFERENCE TIME

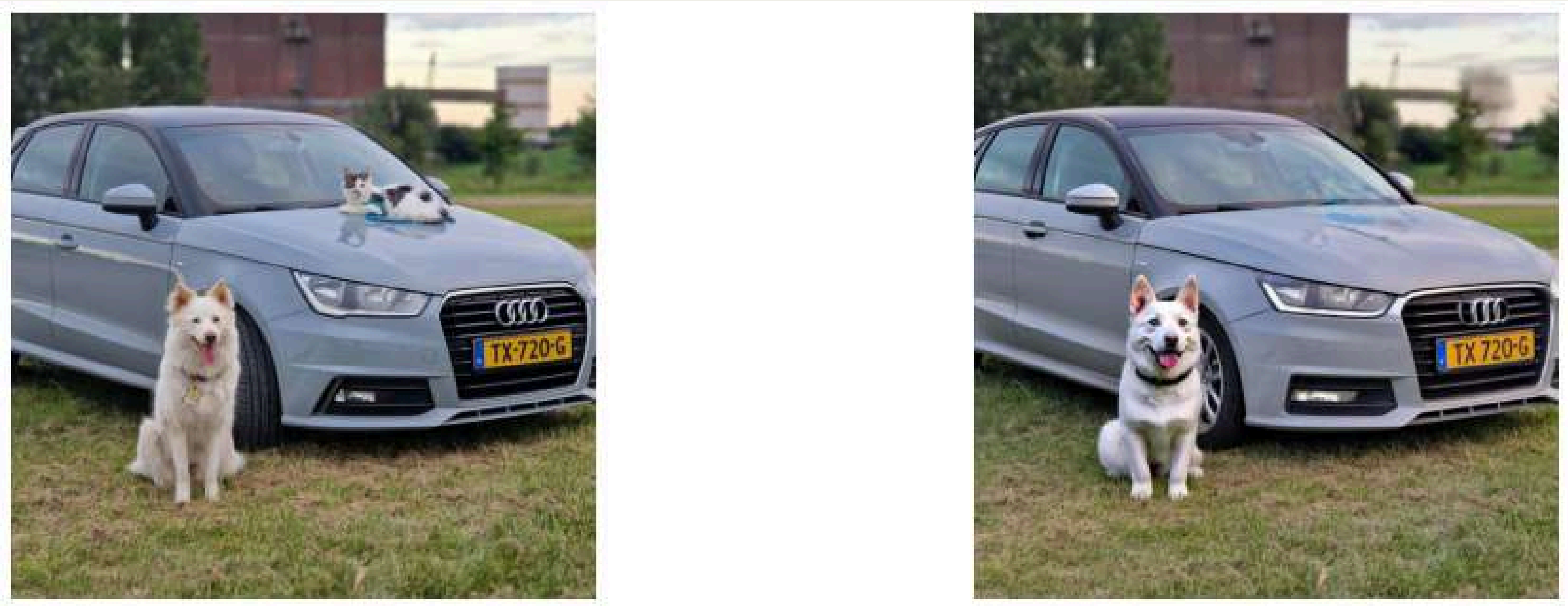
- Accelerate inference to a few-step process.

- Clone the fully trained GenProp model(with SCE and MPD intact) to serve as a student. And use the frozen GenProp model to serve as the teacher.
  - Use a hybrid objective-
    - Instead of standard diffusion noise prediction, minimize the consistency loss wrt. teacher model's multi-step output.
    - Combine RA loss with quality rewards from T2V-Turbo(so J_img(using HPSv2), and J_vid(using ViCLIP)

FLOWEDIT (EVALUATION ASKED BY DR. CAMPS)

FlowEdit is a text-based image editing method for flow models

- **Inversion-free editing:** FlowEdit modifies an image by constructing a direct ODE path from the source image to a target prompt without reconstructing noise.
- **Structure-preserving**: It maintains scene layout, lighting, and geometry while applying text-driven edits.
- **Model-agnostic:** Works with flow models like SD3 and FLUX.



Thank you!