ProPainter

Improving **Propa**gation and **T**ransform**er** for Video Inpainting

Shangchen Zhou Chongyi Li Kelvin C.K. Chan Chen Change Loy

Giovanni Scialla n. 239181
Mattia Nardon n. 233707

Video Inpainting

Video inpainting (VI): fill gaps or missing regions in a video with visually consistent content while ensuring spatial and temporal coherence.

video of DAVIS dataset and ProPainter result

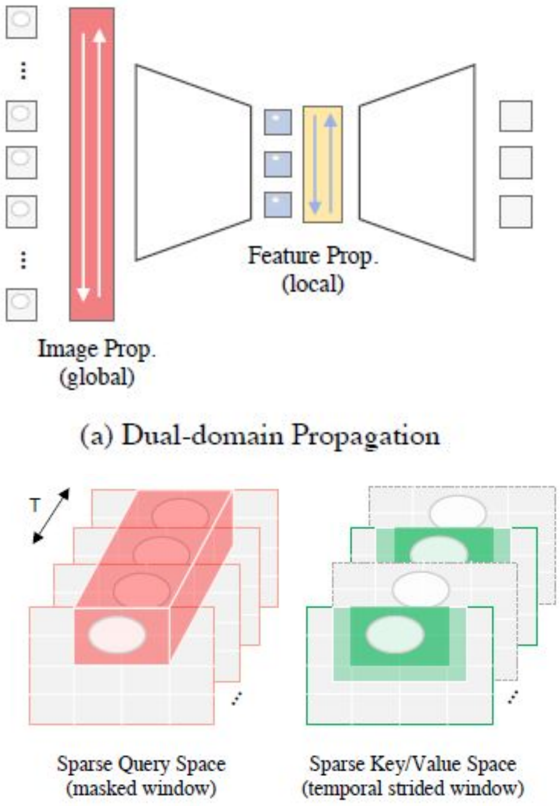
Main challenges

Challenge: Establish accurate correspondence across distant frames for information aggregation

1. **Memory** Constraint
2. **Computational** Constraint

Solutions:

1. Dual-domain Propagation
    - reduce computations
2. Mask-guided sparse video transformer
    - reduce memory



(a) Dual-domain Propagation

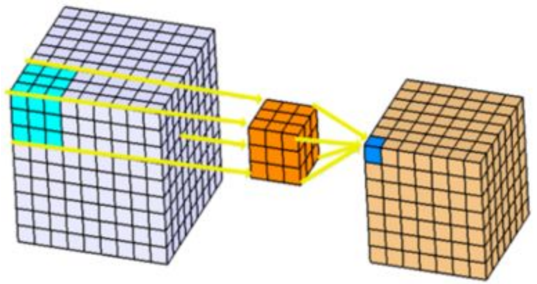
(b) Mask-guided Sparse Video Transformer

Less practical related works

3D CNNs:

Aggregate spatio-temporal information:

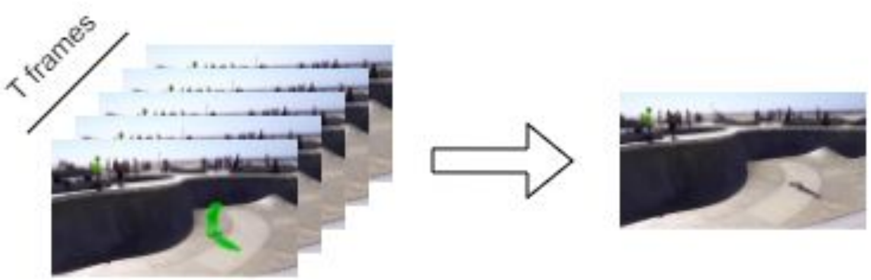
- Suffer from limited receptive fields in both temporal and spatial dimensions
- Less effective for exploring distant content



INTERNAL LEARNING:

Adopt internal learning to encode and memorize the appearance and motion of the video through deep networks:

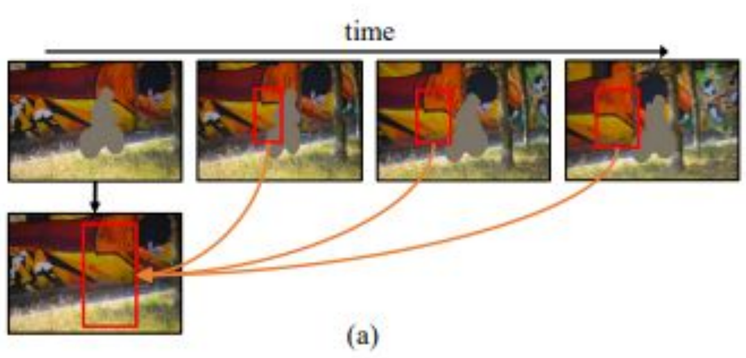
- Require individual training for each video.



Related works

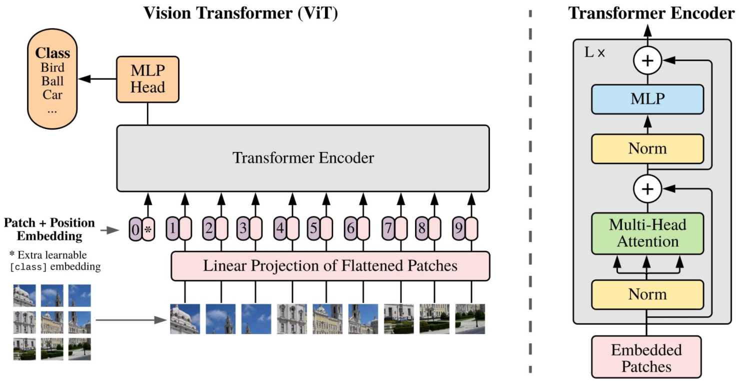
FLOW-GUIDED propagation:

Optical flow and **homography** to align neighboring reference frames to enhance temporal coherence and aggregation.

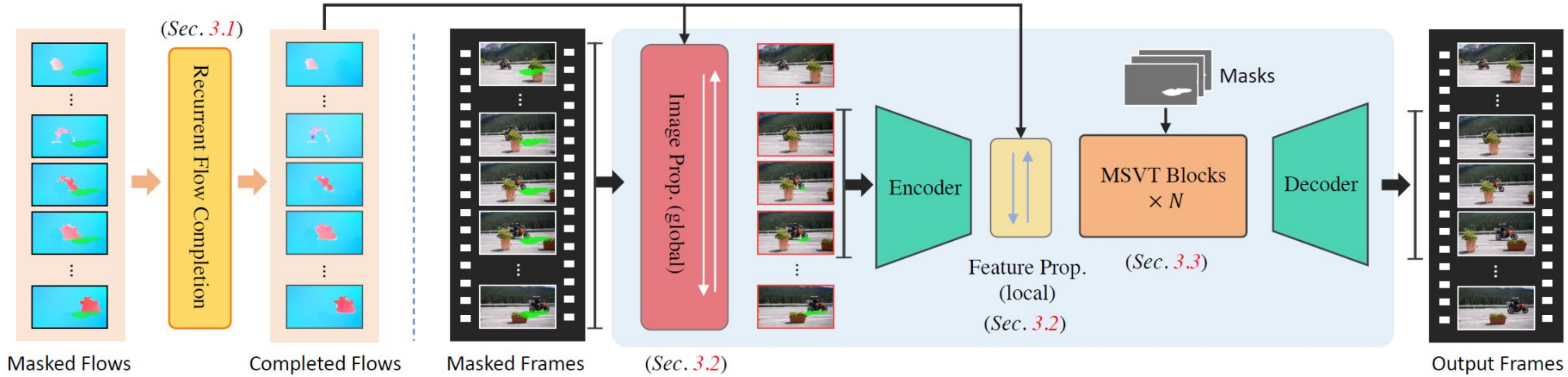


VIDEO TRANSFORMERS:

adopt spatio-temporal attention to explore recurrent textures in a video.



ProPainter: Architecture



Modules:

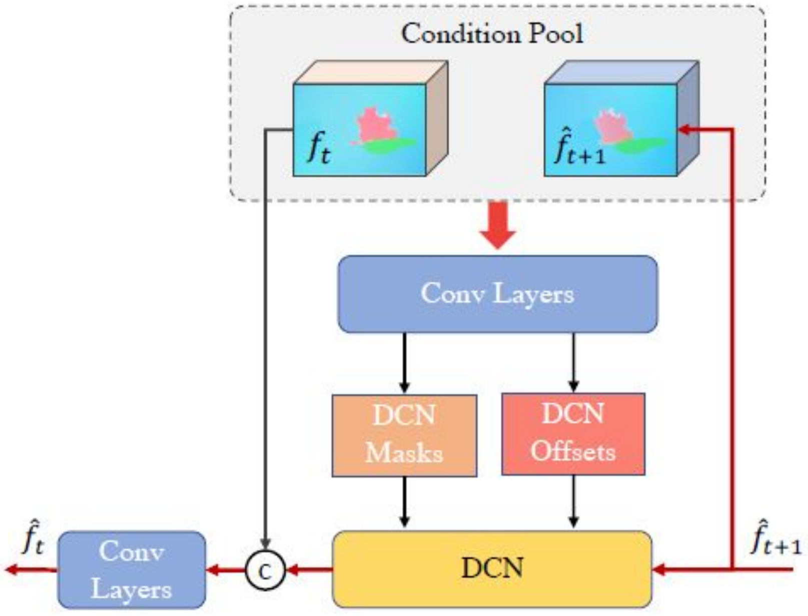
1. **Recurrent Flow Completion (RFC)** : flow field completion
2. **Dual-Domain Propagation (DDP)** : global image propagation and local feature propagation
3. **Mask-guided Sparse Video Transformer (MSVT)** : refine the propagation features and reconstruct the final video sequence

Recurrent Flow Completion (RFC)

use completed optical flow to propagate pixels and maintain temporal coherence.

- Recurrent network to compute forward and backward optical flows
- downsampled feature encoding
- **Deformable Convolution Network** (DCN) to bidirectionally propagate the flow information of adjacent frames
- Decoder to reconstruct the completed flows

$$\hat{f}_t = \mathcal{R}(\mathcal{D}(\hat{f}_{t+1}; o_{t\rightarrow t+1}, m_{t\rightarrow t+1}), f_t),$$



RFC: Flow-guided deformable alignment

1. spatial warping between optical flow of the previous frame and previous features

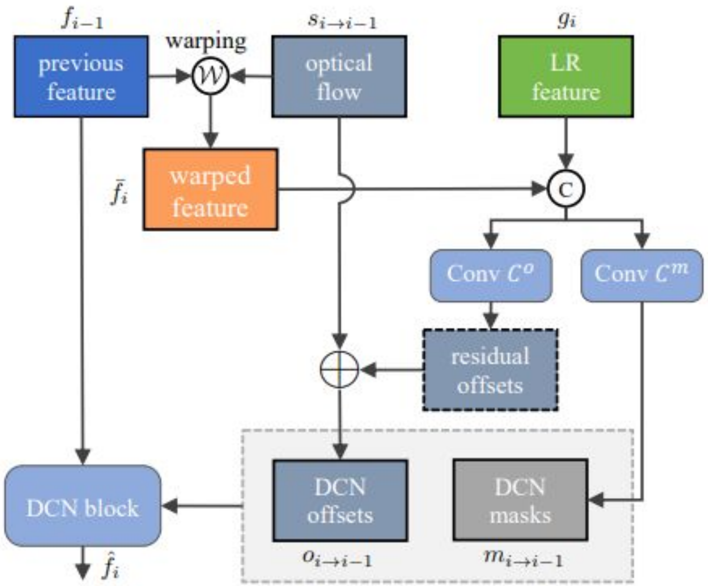
$$\bar{f}_{i-1} = \mathcal{W}(f_{i-1}, s_{i\rightarrow i-1})$$

2. compute the DCN offsets and modulation masks

$$o_{i\rightarrow i-1} = s_{i\rightarrow i-1} + C^{o}\left(c(g_i, \bar{f}_{i-1})\right)$$
$$m_{i\rightarrow i-1} = \sigma\left(C^{m}\left(c(g_i, \bar{f}_{i-1})\right)\right).$$

3. DCN applied to the unwarped feature

$$\hat{f}_i = \mathcal{D}\left(f_{i-1}; o_{i\rightarrow i-1}, m_{i\rightarrow i-1}\right)$$

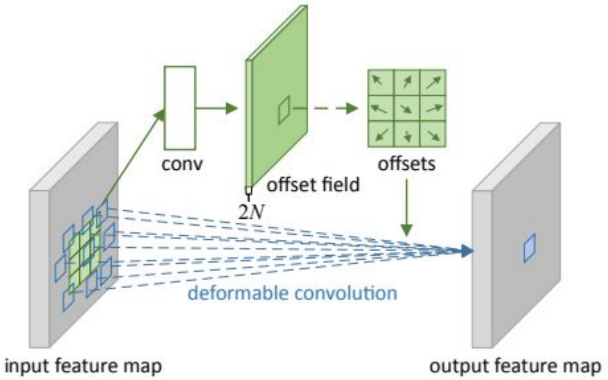


RFC: Deformable Convolution Network (DCN)

- **Deformable convolution**

augment the grid with optical flow offsets

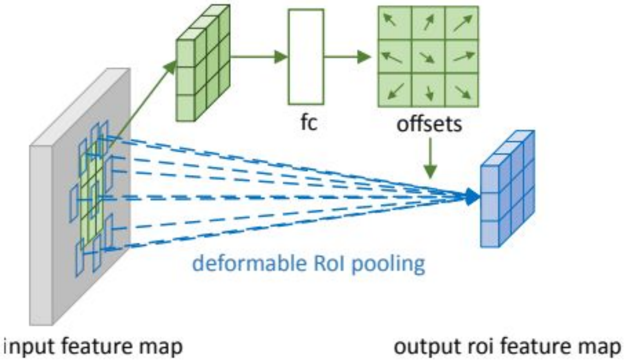
$$\mathbf{y}(\mathbf{p}_0) = \sum_{\mathbf{p}_n \in \mathcal{R}} \mathbf{w}(\mathbf{p}_n) \cdot \mathbf{x}(\mathbf{p}_0 + \mathbf{p}_n + \Delta \mathbf{p}_n).$$



- **Deformable RoI Pooling**

offsets are added to the spatial binning

$$\mathbf{y}(i,j) = \sum_{\mathbf{p} \in bin(i,j)} \mathbf{x}(\mathbf{p}_0 + \mathbf{p} + \Delta \mathbf{p}_{ij}) / n_{ij}.$$



RFC: Separate modules require separate losses

- **Reconstruction loss**

  applied to both valid and invalid regions

$$\mathcal{L}_{rec}^{flow} = \frac{\left\| M_t \odot (\hat{F}_t - F_t) \right\|_1}{\|M_t\|_1} + \frac{\left\| (1 - M_t) \odot (\hat{F}_t - F_t) \right\|_1}{\|1 - M_t\|_1},$$

- **Smooth loss**

  divergence operator to encourage the coherence of completer flow fields

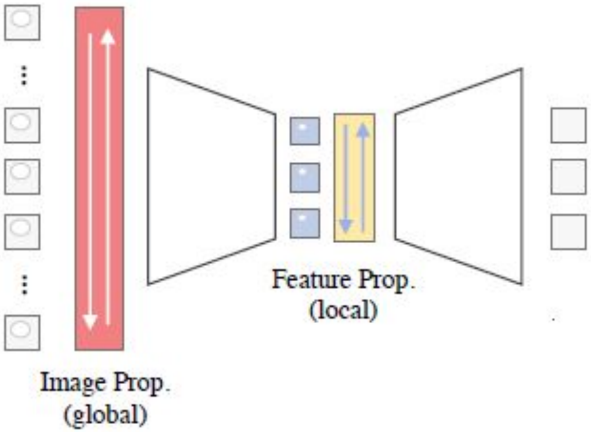
$$\mathcal{L}_{smooth}^{flow} = \left\| \Delta \hat{F}_t \right\|_1,$$

Dual-Domain Propagation (DDP)

- **GLOBAL** propagation in the **image domains**
- **LOCAL** propagation in the **feature domains**

Distinct alignment operations and strategies for each domain is employed.

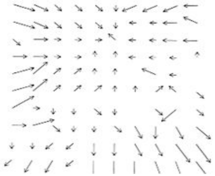
Both domains involve bidirectional propagation in the forward and backward directions.



(a) Dual-domain Propagation

DDP: image propagation with flow-based warping

INPUT: video sequence w/mask, completed flows F



The process start with the identification of the **reliable propagation area** Ar:

$$A_r(p) = \begin{cases} 1 & \text{if } p \in C_1 \cap C_2 \cap C_3, \\ 0 & \text{otherwise.} \end{cases}$$

where ***C1,C2,C3*** are constraints and ***p*** denotes pixel position of the current frame.

DDP: image propagation

$$A_r(p) = \begin{cases} 1 & \text{if } p \in C_1 \cap C_2 \cap C_3, \\ 0 & \text{otherwise.} \end{cases}$$

- **C1**: only pixels with a s**mall consistency error** will be propagated.

$$C_1 : \mathcal{E}_{t\to t+1}(p) < \epsilon \qquad \mathcal{E}_{t\to t+1}(p) = \left\| \hat{F}_{t\to t+1}(p) + \hat{F}_{t+1\to t}(p + \hat{F}_{t\to t+1}(p)) \right\|_2^2,$$

- **C2**: only consider **the masked areas** of the current frame Xt that needs to be filled

$$C_2 : M_t(p) = 1,$$

- **C3**: only propagate the **unmasked areas** from neighboring frame Xt+1

$$C_3 \ : \ M_{t+1}(p + \hat{F}_{t\to t+1}(p)) \ = \ 0.$$

The process of image propagation is expressed as:

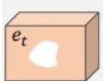
$$\hat{X}_t = \mathcal{W}(X_{t+1}, \hat{F}_{t\to t+1}) * A_r + X_t * (1 - A_r),$$

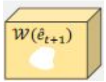
DDP: feature propagation

An **image encoder** is used for extracting the features.

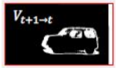
A **flow-guided deformable alignment module** for feature propagation is adopt, similar to the one in the RFC component.

It differs for the conditions for learning DCN offsets:

- current feature 

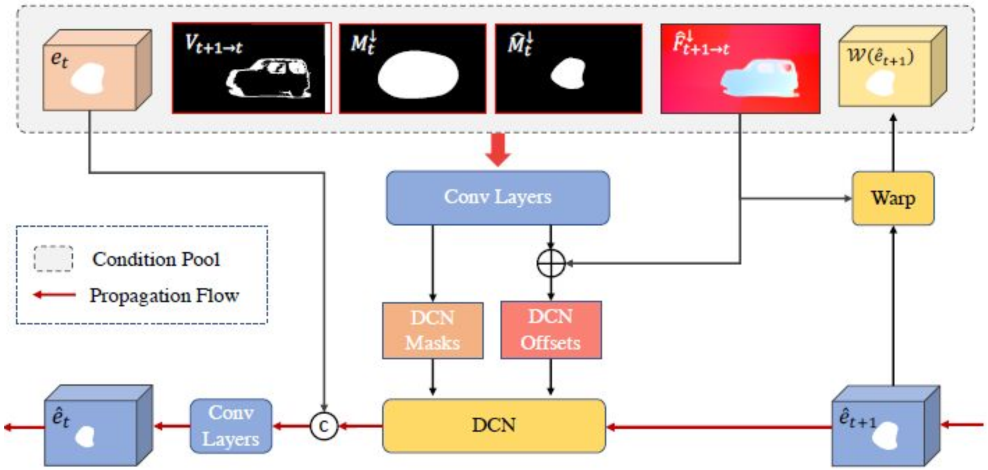
- warped propagation feature 

- completed flows 

- the flow valid map 

- original mask 

- updated mask 

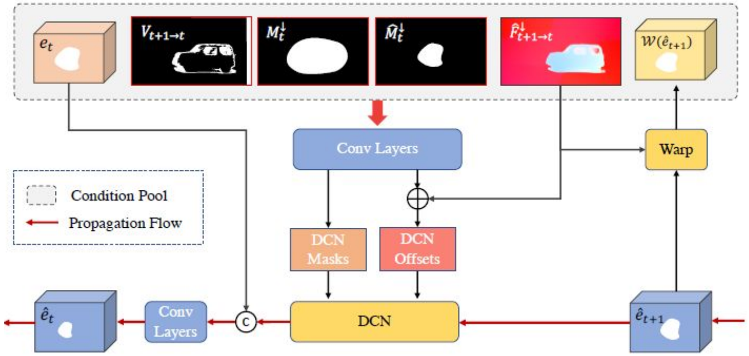


DDP: feature propagation

With these conditions, a stack of **convolutions** is employed to predict the **DCN offset residue** and **modulation masks**.

A **DCN** is then applied to **align** the propagation feature from the previous frame.
DCN alignment propagation is expressed, similar to the previous, as:

$$\hat{e}_t = \mathcal{R}(\mathcal{D}(\hat{e}_{t+1}; \hat{F}^{\downarrow}_{t\rightarrow t+1} + \widetilde{o}_{t\rightarrow t+1}, m_{t\rightarrow t+1}), f_t)$$



Finally, a **CNN block** is employed to fuse the current and aligned features, achieving **the propagation feature** of the current frame.

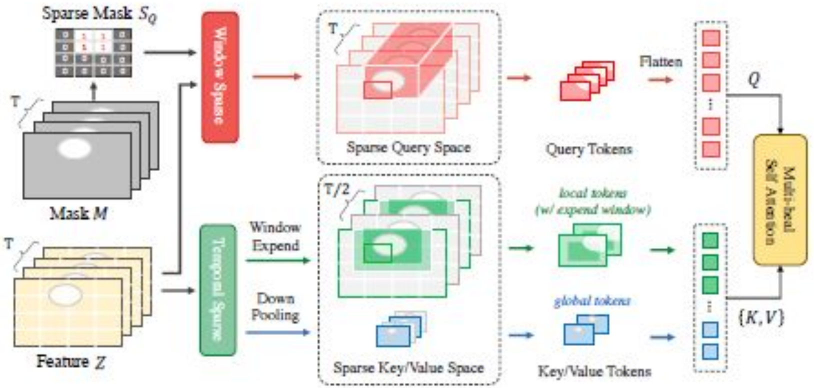
Mask-guided Sparse Video Transformer (MSVT)

Novel sparse video Transformer that builds on the window-based approach.

From video sequence feature use **soft split operator** for having many overlapping patches.

The **query Q**, **key K**, and **value V** are obtained through **linear layers**, from the patches.

Sparse strategies designed to reduce memory and computation



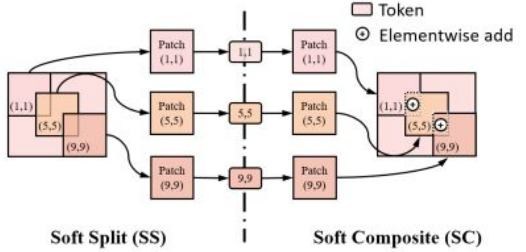


Figure 3. The illustration of Soft Split (SS) and Soft Composite (SC) module.

MSVT: SPARSE QUERY SPACE

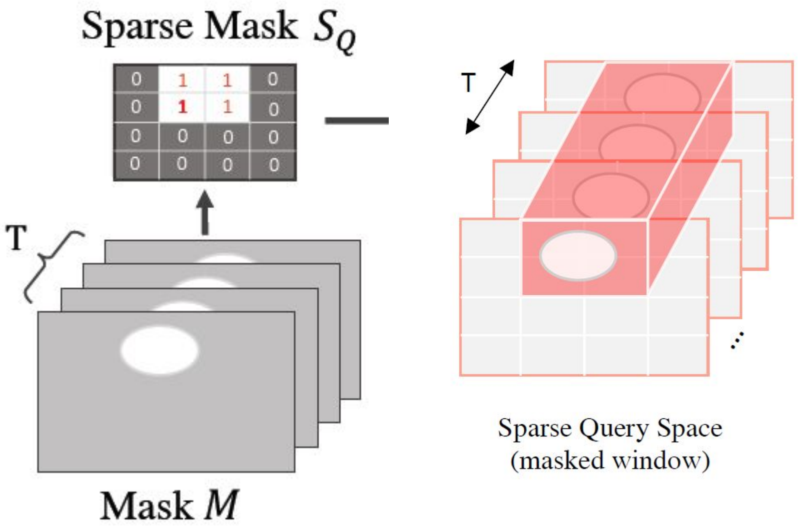
The **mask** is usually **small** in reference to the full video frame, and **spatiotemporal attention** may not be necessary for all query windows

Apply attention only to query windows that intersects with mask regions.

- the masks are downsampled and partitioned
- sum it up in the temporal dimension

To obtain sparse mask S_Q

$$S_Q = Clip\left(\sum_{t=1}^{T_l} M_t^{\downarrow}, 1\right),$$

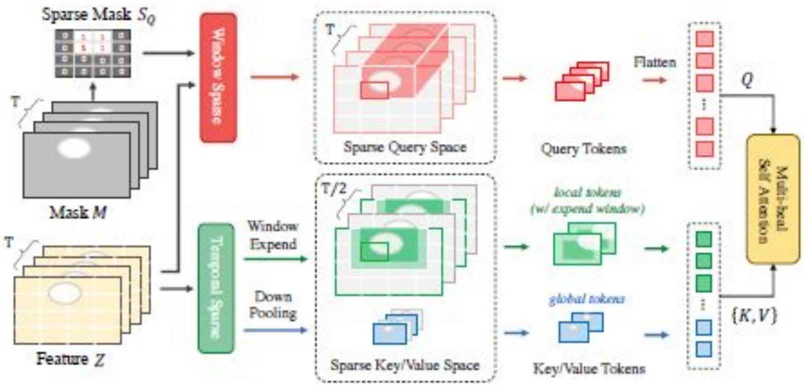


MSVT: SPARSE KEY/VALUE SPACE

Due to the highly redundant and repetitive textures in adjacent frames, it is unnecessary to include all frames as key/value tokens in each Transformer block.

Only include strided temporal frames alternately, with a temporal stride of 2 in the design.

By doing so, the key and value space is reduced by half, effectively reducing the computation and memory cost of the Transformer module.

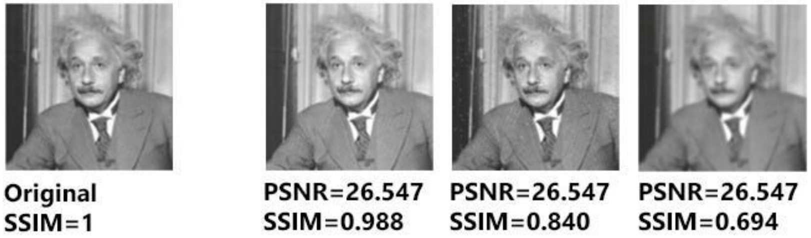


Metrics

Metrics:

- **Peak Signal-to-Noise Ratio (PSNR)**
  - the ratio between the maximum possible value of a signal and the value of distorting noise that affects the quality of its representation

- **Structural Similarity Index Measure (SSIM)**
  - It considers luminance, contrast, and structure, comparing the local patterns of pixel intensities in the original and distorted images.



Results

Paper results

	ProPainter
PSNR ↑	34.47
SSIM ↑	0.9776

50 video clips,
from DAVIS dataset

Our results

	ProPainter
PSNR ↑	23.536
SSIM ↑	0.912

90 video clips 854 × 480,
from DAVIS dataset with half precision inference.

psnr = 13.520
ssim = 0.8523

psnr = 32.312
ssim =0.9905

↑ higher value is better

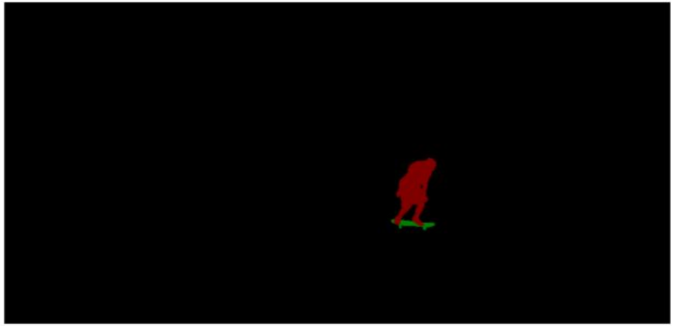
PROPOSED IMPROVEMENTS

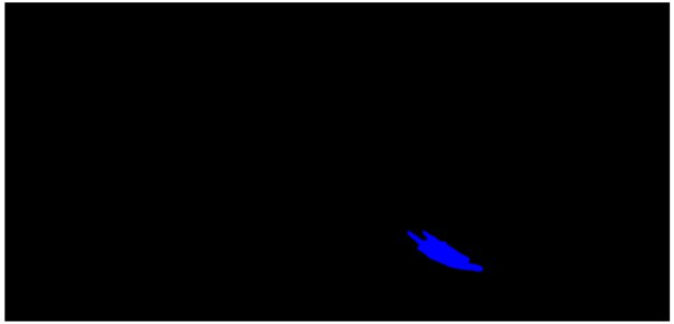
- qualitative experiments:
  - shadow detection and removal
  - single/multi-prompt detection and removal
- new sparse strategy
- improving video inpainting with diffusion model (?)

Shadow detection and removal

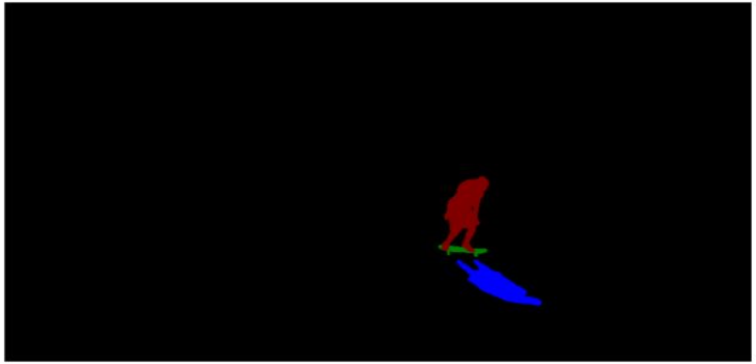
Using Segment Anything Model (SAM) to segment shadow masks in each frame

Shadow detection and removal





- Add up the segmented shadow mask with the original one



- Use propainter for the impainting

Shadow detection and removal: results

Multi-stage approach for object removal by textual prompts

4 Models in a cascade in order to remove objects in video sequences:

- **Yolov8** - Object Detection in the scene
- **Clip** - Select object based on the input textual prompt
- **SAM** - Segmentation of the objects
- **ProPainter** - Object removal given the segmented masks

spatial information between frames has been added during the object detection phase in order to be consistent with the video sequences.

Results: Paragliding

Textual prompt = “The white and red glider”

Results: Stroller

Textual Prompt: “The person with a white shirt”

Multi-stage approach for object removal: Drawbacks

- **Multi-stage curse**: Since the models are used in a cascade approach, if one of them fails into doing the inference , the whole network will fail
- **Limited number of classes**: the models are pre-trained on a fixed number of classes (expecially yolov8 has in total 80 classes) , so i can't remove objects out of those classes.

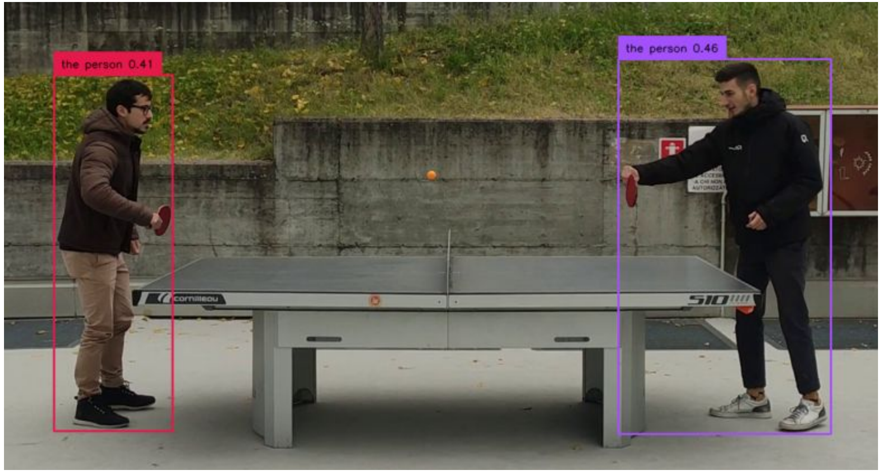
Textual Prompts Object Removal with GroundingDINO

- **Single Prompt**

Text = 'The small orange ball'

- **Multi Prompt**

Text = 'The person on the right and the person on the left'



GroundingDINO : Results

Original Video

Single Prompt Inpainting

GroundingDINO : Results

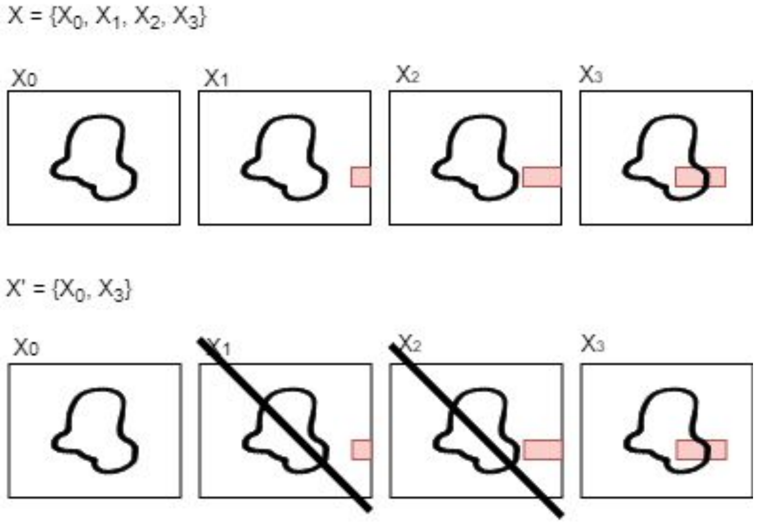
Original Video

Multi Prompt Inpainting

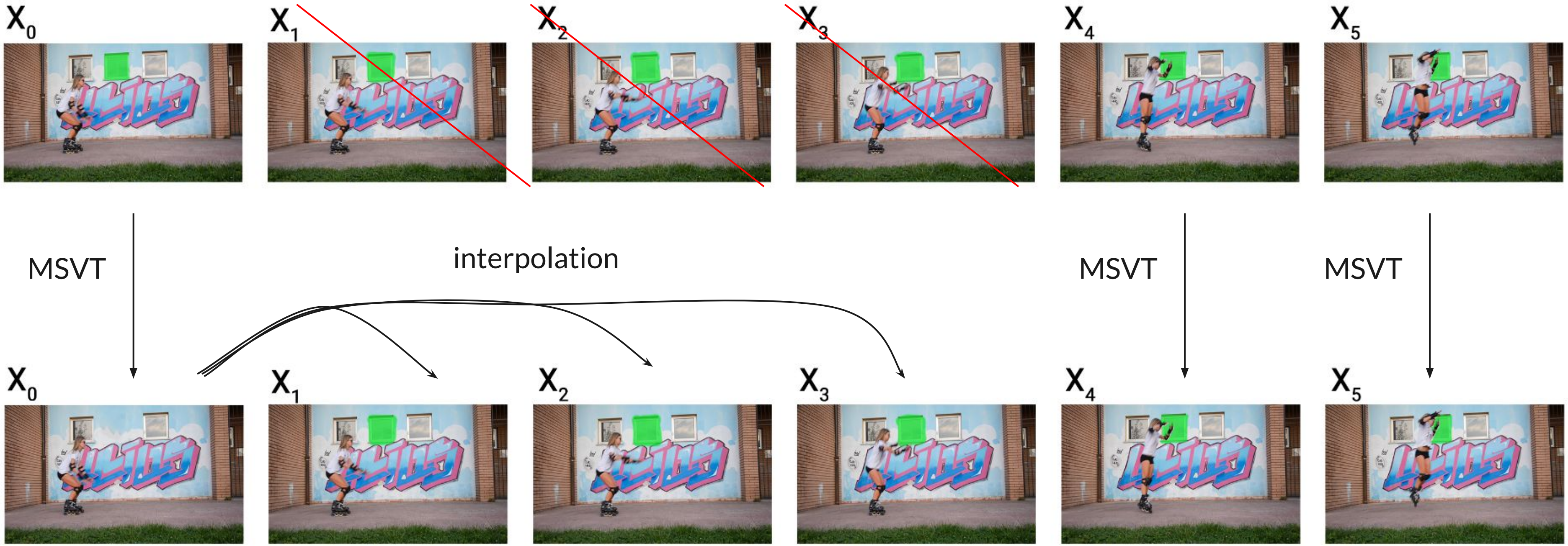
NEW SPARSE STRATEGY: adaptive sparse video transformer for fast computing during static frames

GOAL: leveraging the temporal coherence of low-motion regions.

As input for the MSVT we reduce the sequences X by $$X' = X - \sum_{t=1}^{T} Match_t(|O_{t-1} - O_t| \leq \gamma, X_t) \quad \gamma \simeq 0$$

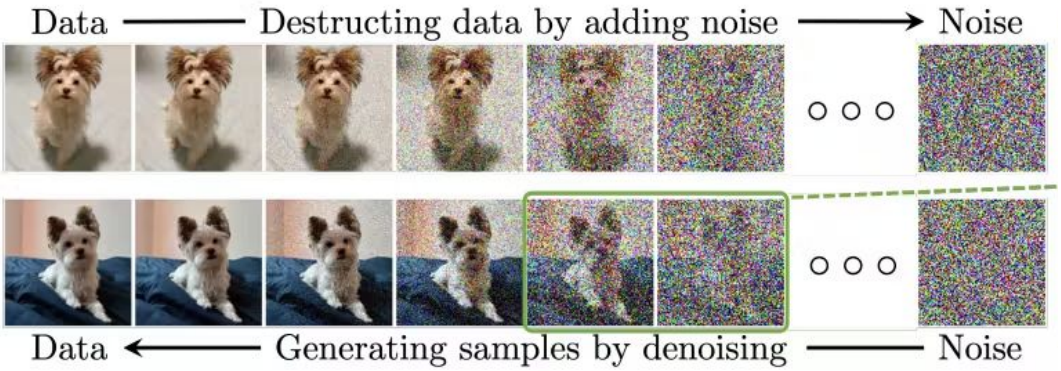


$X' = X_0, X_4, X_5$



Improving video inpainting with diffusion model (?)

Diffusion models are **generative models**, that generate new data from a noisy space using denoising techniques and conditioning to achieve desired results.

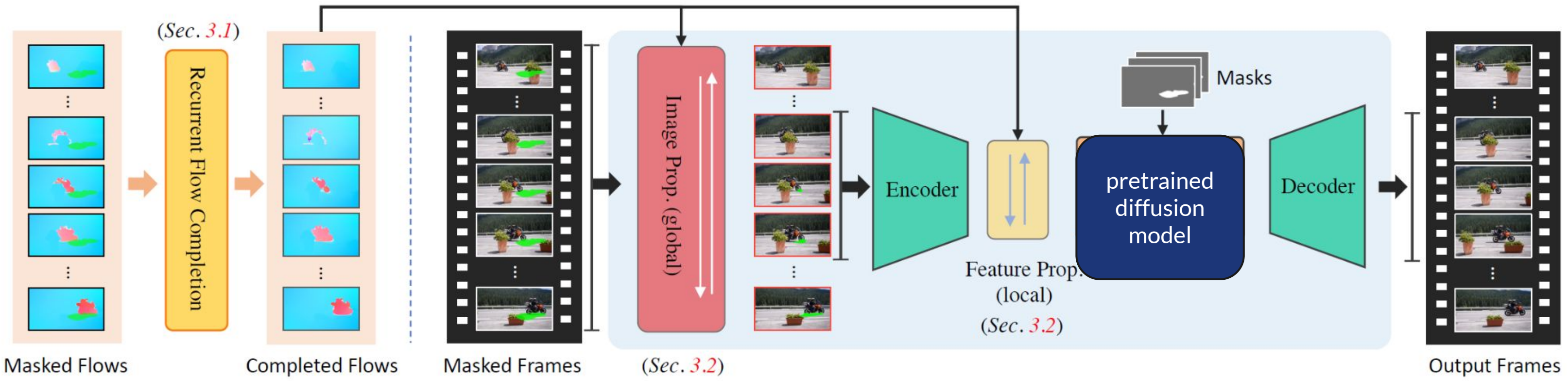


Video inpainting with diffusion models is often treated as an **internal learning** task, demonstrating high accuracy but at the expense of a slower computational speed.

Improving video inpainting with diffusion model

OUR IDEA:

- substitute the MSVT block with a **pretrained diffusion model** fine tuned for video
- use features and optical flow as **conditioning**



Pros

Some **advantages** of using diffusion models are:

- better **quality**
- better **results** in non conventional cases

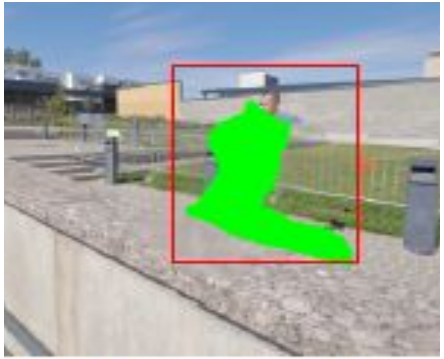


image with mask

Propainter

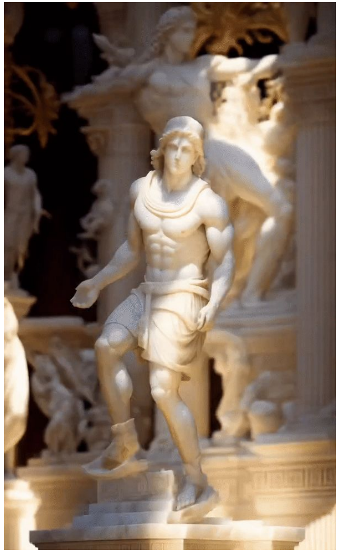
Dall-e2, prompt:”infill according to the background”

Cons

SLOW INFERENCE

possible solutions:

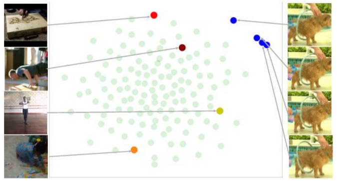
- use lower quality models,
- sparse strategy with interpolation



TEMPORAL CONSISTENCY (flickering problem)

possible solutions:

- **sampling of noise for better correlation**
  Preserve Your Own Correlation: A Noise Prior for Video Diffusion Models (ICCV'23)
- **temporal layers**
  Stable Video Diffusion: Scaling Latent Video Diffusion Models to Large Datasets



Thank you for the attention

Bibliography

- Shangchen Zhou, Chongyi Li, Kelvin C.K. Chan, Chen Change Loy. ProPainter: Improving Propagation and Transformer for Video Inpainting. In ICCV, 2023
- Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. 2015
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. 2021
- Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, Ross Girshick. Segment Anything. 2023
- Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, Lei Zhang. Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection. 2023
- Songwei Ge, Seungjun Nah, Guilin Liu, Tyler Poon, Andrew Tao, Bryan Catanzaro, David Jacobs, Jia-Bin Huang, Ming-Yu Liu, Yogesh Balaji. PYoCo: A Noise Prior for Video Diffusion Models. NVIDIA, University of Maryland, College Park, University of Chicago. In ICCV, 2023
- Blattmann et al. Stable Video Diffusion: Scaling Latent Video Diffusion Models to Large Datasets. 2023