ROVR: Reinforcment Optimized Video Reconstruction

Aaditya Prasad
Stanford University
aadityap@stanford.edu

Arnuv Tandon
Stanford University
arnuv@stanford.edu

Arjun Vikram
Stanford University
arjvik@stanford.edu

1. Extended Abstract

Video to Video translation is a common problem space in machine learning, where an input video is adjusted to match some specific output distribution or task. In this paper, we tackle a specific version of this problem: video reconstruction. In video reconstruction, a network is tasked with reconstructing a visually corrupted video and returning it to the original, pre-corruption distribution.

Prior solutions for video to video reconstruction fall into two general categories. The first of these are autoregressive strategies which perform inference on frames sequentially and condition on prior frames. This approach lacks generalizability, as many video reconstruction tasks require considering context from both neighboring and distant frames in order to infer accurately. The second category of strategies for video reconstruction are models such as Runway's GEN-1 and GEN-2 which leverage attention mechanisms to determine the optimal frames to condition on when performing inference on a given target frame. These models, however, are extraordinarily costly to apply to long sequences because of the quadratic time complexity of attention, rendering them infeasible for many use cases.

In this paper, we introduce the Reinforcement Optimized Video Reconstruction (or ROVR) platform. ROVR splits the reconstruction problem into two separate tasks at every time step. We first take a target frame (stepping through an input video autoregressively) and feed it to an actor-critic policy optimized with phasic policy gradient (PPG). Our policy conditions on a compressed, feature-extracted version of the entire video and produces two frames to condition on when reconstructing the given target frame. Note that our policy is able to produce these conditioning frames in **sub-quadratic time complexity** given that we do not leverage attention mechanisms.

The target frame and conditioning frame are then fed to a local network, which is able to condition on the derived conditioning frames to reconstruct the target frame in the pre-corruption distribution.

We select reinforcement learning for this task of video reconstruction given that we need to optimize with respect to two objectives: a) perceptual loss at each timestep of the video and b) optical flow of the reconstructed video. The former objective is local, while the latter objective is global – matching the strengths of reinforcement learning.

Throughout this paper, we evaluate the ROVR platform on the task of video in-painting across a masked video. This task is demanding as it requires consideration of context from both nearby and distant frames to ensure accurate in-painting and preservation of temporal consistency.

To promote training stability, we provide a warm-start to our policy via imitation learning, and pretrain our local network on random conditioning images – rendering the local network an unbiased estimator of the quality of conditioning frames.

Our results indicate three key points. First, our policy learned to optimize for the global objective of maintaining optical flow – validating the use of reinforcement learning for this problem. Second, our imitation learning warm-start positively contributes to downstream performance (perceptual loss and optical flow). Finally, ROVR beats the baseline of NVIDIA's Vid2Vid network, which operates on frames in an autoregressive manner and conditions sequentially on prior frames. We again emphasize that all of these results from ROVR were derived in sub-quadratic time complexity.

We see several avenues for future work. Conditioning on k frames, as opposed to two frames, substantially improves the complexity of tasks which we may tackle. More complex tasks which we plan to evaluate ROVR on include saturation and brightness perturbations, as well as various types of imaging noise.

Further, we plan to explore transitioning the choice of conditioning frames from a multi-label classification task to a ranking or retrieval task – replacing the binary cross-entropy loss with a contrastive loss or other adjacent objectives. This modification would improve imitation learning performance, thus improving the performance and stability of the entire ROVR architecture.

2. Introduction

Video to Video Translation is a common problem space in machine learning, where an input video is adjusted to match some specified output distribution or task. A specific version of this problem is video reconstruction, where a network is tasked with reconstructing a visually corrupted video and returning it to the original, pre-corruption distribution. Video translations are often evaluated using both a frame by frame translation loss, as would be used in a image to image translation task, as well as some global objective that ensures the video retains temporal consistency.

Solutions to video reconstruction and video to video translation more generally can be seen to fall into two general regimes. The first of these are auto-regressive strategies which work from the first video frame to the last, applying translations based on some fixed conditioning inputs (such as conditioning edit t on the frames at timesteps $t-1, t-2$). These strategies simplify the video reconstruction task but rely on the assumption that auto-regressive generation and conditioning are close to optimal, which is not always sound. As we explore in this paper, such an auto-regressive strategy struggles when the input video is corrupted with masks or occlusions, since these are often invariant over sequential frames.

The second regime includes more expensive and often diffusion-based strategies such as Runway's GEN-1 and GEN-2. These strategies utilize deep latent diffusion networks that compute attention and convolutions over temporal dependencies. At scale, these strategies can produce amazing results but are costly to apply to long sequences due to the quadratic time complexity of attention.

In this paper, we propose a new video reconstruction platform that divides the reconstruction task into two separate tasks at every time step: choosing conditioning frames and reconstructing the current frame given the conditioning frames. The first key insight here is that, while both of these tasks require learning inter-frame dependencies, they operate at different range and fidelity requirements. Choosing conditioning frames requires learning long range dependencies across all frames in a video, but can be done with low-dim representations of the frames. Reconstructing a frame only requires learning dependencies between the chosen target and conditioning frames, but needs to be done at a higher fidelity to preserve the output.

Additionally, while the local reconstruction task is a supervised learning problem, the decision of which conditioning frames to use should be made with reinforcement learning. This is because a function that outputs the current best conditioning frames is necessarily dependent on the current video, including all reconstructions completed up until the current time step. Thus we teach our agent to pick conditioning frames using PPG and global reconstruction rewards, while our local network is trained to minimize local reconstruction loss at any given time step.

In Section 3, we summarize prior works that we build off of or compare ourselves with. In Section 4, we explain the task and dataset used in to test our system. In Section 5, we describe in detail our Markov Decision Process, implementation details for our networks, and justifications for system design choices. In Section 6, we explain how our networks are trained. In section 7, we provide and discuss results, and in Section 8 we conclucde.

3. Prior Work

State of the art approaches to **Video to Video Translation / Synthesis** employed by Blattmann et al. [1] incorporate attention-based transformers, converting input videos into sequences of patches and converting outputted sequences back into patches that can be concatenated and stacked to form videos. However, due to the quadratic nature of attention [7], these methods are incredibly computationally intensive and scale poorly with the length and resolution of the generated videos.

Alternate approaches in video to video translation attempt to resolve this issue of quadratic time complexity by introducing inductive biases in the method used to select the sequence in which to process frames, as well as contextual frames. Wang et. al. propose the Vid2Vid architecture [9], which employs perceptual-feature-preserving image translation techniques on a frame-by-frame basis, conditioning on the prediction of previous frames to ensure temporal consistency and proceeding sequentially to construct the output video in an autoregressive fashion. Further work by Wang et. al. attempts to reduce the training complexity of such methods [8] by introducing aggressive feature extraction to downsample inputs into smaller latent space.

Liang et. al. tackle the problem of style transfer in art by introducing a frame selection process which maximizes the amount of new information provided for style transfer at each step [6]. Despite these advances, the method suffers from limited generalizability due to its heavy reliance on inductive bias in the frame selection process. Existing methods attempt to avoid inductive biases in the frame selection process by utilizing attention on a low-dimensional version of the source video. Huang et. al. leverage an aggressive feature extraction network and attention [5] in order to perform few shot video to video translation. While this method is free of inductive biases, attention still runs in quadratic time

complexity – rendering the method infeasible for longer, higher resolution videos.

Our approach, **Reinforcement Optimized Video Reconstruction**, is inspired by several tangential works. Cao et. al. employ a similar RL based approach to sequencing patches and context for large image synthesis [2] in a patch-by-patch manner. This approach inspired us to employ an RL agent to sequence the reconstruction of video frames. Dosovitskiy et. al. propose a perceptual image loss [3] which we make use of to avoid regression-to-the-mean found with pixel-target-based losses. Finn et. al. train an RL agent to produce sequential video frames [4], inspiring our RL-based autoregressive approach to video reconstruction.

4. Task and Dataset

We chose the task of video in-painting across a masked video as a testbed to train and evaluate our proposed approach. This task is demanding as it requires consideration of context from both nearby and distant frames to ensure accurate in-painting and preservation of temporal consistency.

In dataset preparation, we apply four randomly sized and placed masks per frame (see Figure 1). While the random location of the masks in a frame is independent with respect to the position of the frame within the video, the random system is designed such that a limited number of frame pairs are valid conditioning options for any given target frame, while the majority of pair choices in the video do not provide enough information to reconstruct the target. The positions of these correct conditioning frames are also independent random variables with respect to the position of the original target frame, but importantly our masking algorithm allows us to recover the correct and incorrect solutions deterministically. This allows us to run training and testing in a self-supervised fashion.

Additionally, this design encourages our policy to select context frames that are not the immediate neighbors of the frame we have selected to edit at timestep t. This is desirable as our policy is forced to understand both long and short range dependencies.

We use the RealVSR dataset for videos. For simplicity, each video is 20 frames and has dimension 3 x 256 x 256.

5. Reinforcement Optimized Video Reconstruction

5.1. Markov Decision Process

We define a Markov decision process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ are state and action spaces, $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ is a state-transition probability function, $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is a reward function and γ is a discount factor. We use $\gamma = 1$ in this work since the trajectories are fixed length and the goal is the final product, rather than any intermediary success.

5.1.1 States

We denote the original, uncorrupted video as X and the corrupted video as $\bar{X}$. At any timestep $t \in \{1, ..., T\}$, the state s_t is an encoded representation of the entire video at the current timestep $\bar{X}_t$ (see Figure 2) as well as a downsampled representation of the target frame $\bar{x_t}$ which we are changing next. In this paper, we take $T = 20$ and $\bar{x_t} = \bar{X_{t_t}}$, i.e. we step sequentially through the video from start to finish. This state is provided to our Policy Network $\pi(\cdot)$.

5.1.2 Actions

$\pi(s_t)$ outputs the two frames c_{t_1}, c_{t_2} that should be used to reconstruct the target frame $\bar{x_t}$. The action space is every pair of frames that could be used to condition on, which is 19 choose 2 or 171. Of these, our limited task ensures that at most 16 specific pairs can achieve success.

5.1.3 Transitions

After our policy network decides on c_{t_1}, c_{t_2}, state to state transitions are modeled by our local network $\Gamma(\cdot, \cdot, \cdot)$, which takes the target frames and two conditioning frames and outputs $\Gamma(\bar{x_t}, c_{t_1}, c_{t_2}) = \hat{x_t}$, which is the t'th reconstructed frame. Then, s_{t+1} is constructed using the encoding of $\bar{X}_{t+1}$, which is made up of reconstructed frames up through timestep t and corrupted frames past then.

5.1.4 Reward

We calculate a local reconstruction reward at every timestep t as well as a global optical-flow reconstruction reward at the end of the trajectory. The local reward ensures that the model learns to predict useful frames at every timestep for the specific image to image translation task, while the global reward encourages π to optimize for temporal consistency in the final product. We will cover the specific reward formulations used in more detail in Section 6.

5.2. Networks

5.2.1 Agent

As stated previously, our agent is a policy network $\pi(\cdot)$. $\pi(\cdot)$ passes the encoded video and the target image, which together make up the current state, each through

Figure 1. Sample masked images from the RealVSR dataset. Our policy must learn to select context frames that provide information about masked-out regions.

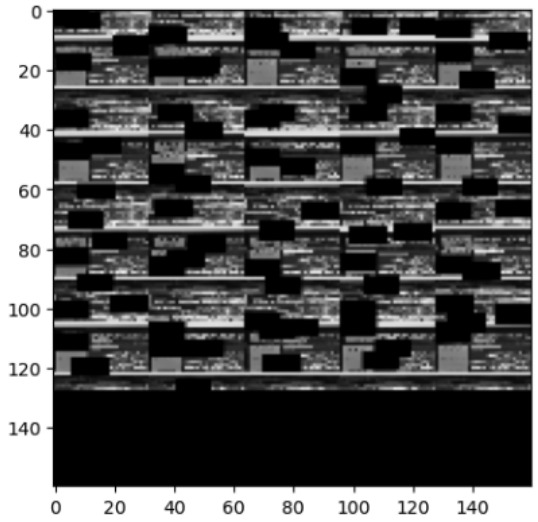


Figure 2. Downsampled representation of videos from the RealVSR dataset. We take 20 frames of shape (256, 256, 3), downsample each frame to (32, 32), and patch the downsampled frames into a new (160, 160) representation.

its own convolutional network. The outputs of those networks are stacked together and then passed through a fully connected network which output logits over 19 output classes, which represent the 19 possible conditioning frames. We use a Gumbel softmax with a temperature of .7 across these 19 conditioning frames and then choose the two largest predictions as c_{t_1}, c_{t_2}. The Gumbel softmax encourages the output values to be more uniform than a traditional softmax, which maintains stability when we pick the top two values.

This stability is important when we calculate the log probabilities of actions given states, which is required for the Reinforcement Learning algorithm we utilize, PPG. Additionally, through testing, we learned that using the geometric mean of the individual log-probs of c_{t_1} and c_{t_2} is more stable than using the more traditional sum of the log-probs, which corresponds to the log of the probability of choosing each conditioning frame independently.

5.2.2 Critic

Phasic Policy Gradient (PPG) is an actor-critic method, meaning it requires a parameterized critic network $V(\cdot)$ which takes in state s_t and predicts the reward to go at the given timestep r_t. Our implementation of $V(\cdot)$ is almost identical to our implementation of $\pi(\cdot)$, with the key difference being that the final fully connected layer outputs a single scalar rather than logits. This scalar is the predicted reward.

5.2.3 Local Network

Our local network $\Gamma(\cdot, \cdot, \cdot)$ is a UNet that stacks the input target and conditioning frames and uses convolutional layers to take them down to low-dimensional space and then back up to a generated image. We directly use the output of our local network as our predicted reconstructed image $\hat{x_t}$

6. Learning Algorithms and Procedure

To avoid the problem of reward propagation, we warm-start learning by using imitation learning to pretrain π and randomly sampled supervised learning to pretrain Γ. We then connect our whole system together as described in Section 5 and utilize PPG. Each of these strategies is described in detail here.

6.1. Local Network Pretraining

We aim to optimize the distance between the original ground truth image and the reconstructed image Γ outputs. We do this by combining simple supervised loss with a learned perceptual loss.

$$\mathcal{L}_R(x, \hat{x}) = \sum_l \frac{\lambda}{W_l H_l} \sum_{w,h} \left\| w_l \odot \left(\phi(x)^l_{wh} - \phi(\hat{x}^l_{wh}) \right) \right\|_2^2 + \frac{1-\lambda}{WH} \sum_{i=1}^{W} \sum_{j=1}^{H} (x_{i,j} - \hat{x}_{i,j})^2 \quad (1)$$

where ϕ is a pretrained model, in this case VGG, used for discerning perceptual features. λ is a hyperparameter that we grow exponentially from 0 to 0.9 throughout

pretraining of the local network, as MSE loss helps force early learning but introduces unwanted behaviors such as feature averaging, while perceptual loss optimizes for higher level feature similarity.

We train Γ using this loss and a simple random sampling strategy where the conditioning frames c_{t_1}, c_{t_2} are chosen randomly for any input target frame $\bar{x_t}$. This random strategy is necessary because it allows us to decouple pretraining from any assumption about π's performance. Random sampling does mean that sometimes Γ will be given an image it cannot reconstruct fully with the chosen context, but it still learns valuable information about the reconstruction task.

6.2. Policy Network Pretraining

To give our Policy Network a warm start, we pretrain it using Imitation Learning. Specifically, we optimize for Binary-Cross-Entropy loss between the class probabilities predicted by the actor network and ground truth class values y_t, as shown below.

$$\mathcal{L}_{IL}(y_t, \pi(s_t)) = -[y_t \log(\pi(s_t)) + (1-y_t)\log(1-\pi(s_t))] \tag{2}$$

We alternate between training on positive and negative class values since we can generate examples of both pairs that should be used and pairs that shouldn't be used.

It is important to note that while BCE is widely used in multi-label classification settings, it is usually used under the assumption that different label classifications are independent events. This is not entirely true in ROVR: some of choices of classes are independent with respect to each other but the entire action space is not independent. For example, there are some frames which are good choices with certain other frames, since between them they provide enough information to restore the original frame, but they are not good choices with other frames that do not provide the required marginal information gain. Further discussion of the effects of this is in the results section.

6.3. End to End Optimization

We optimize our entire network end to end by executing our MDP, training our local network on reconstruction loss $\mathcal{L}_{\mathcal{R}}$ at every time step in the trajectory, and optimizing our actor and critic networks as per PPG. We have already seen the local network's reconstruction loss, so this section focuses on PPG.

To calculate rewards, we start by calculating $\mathcal{L}_P$, which is just the perceptual loss component of $\mathcal{L}_{IL}$ unweighted by λ. We only use perceptual loss since the sliding weight scheme in $\mathcal{L}_{IL}$ destabilizes the reward signal and hinders training and perceptual loss is more important to optimize for reconstruction.

Then, we calculate the marginal reward at each time step as

$$\mathcal{R}_t = \mathcal{L}_P(\bar{x_t}, x_t) - \mathcal{L}_P(\hat{x_t}, x_t)$$

This is the improvement the local network made to the corrupted video during the last transition, or how much better s_{t+1} is than s_t.

Next, we add a global optical flow reward to the end of the sequence.

$$\mathcal{O} = 1 - \frac{|\varphi(\hat{X}) - \varphi(X)|}{|\varphi(\bar{X}) - \varphi(X)|} \tag{3}$$

This measures optical reconstruction, or how much of the original video's (video without masks) optical flow the reconstructed video recovered compared to the corrupted video. A score of 1 indicates that the optical flow of the reconstructed video is equal to the optical flow of the original video. A score of 0 indicates that the optical flow of the reconstructed video is equal to the optical flow of the masked video. In other words, a score of 0 indicates that the model has not improved the optical flow of the masked input video.

Finally, we compute rewards to go at each timestep based on these marginal rewards. Because we used $\gamma = 1$, this was a simple reversal and sum of the rewards sequence. $\mathcal{R}_t$ will be used to refer to rewards to go for the rest of the paper.

In simplistic terms, PPG turns policy gradient into an off-policy algorithm by clipping learning so that the current policy does not stray too far from the data we are using to calculate reward gradients. The difference between PPG and PPO is that in PPO, the actor and critic networks share parameters. We chose PPG because it empirically achieves higher sample efficiency.

PPG utilizes advantages, which we calculate using the formula $\hat{A}_t = V(s_t) - \mathcal{R}_t$. We then optimize the objective

$$\mathcal{L}^{CLIP}(\theta) = \hat{\mathbb{E}}_t \left[\min \left(r_t(\theta)\hat{A}_t, \text{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_t \right) \right]$$

The $r_t(\theta)$'s in the objective above denote the probability ratio $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$, where $\pi_{\theta_{\text{old}}}$ denotes the policy network used to compute the last rollout.

Finally, the critics are regressed to the true rewards-to-go by minimizing the following

$$\mathcal{L}_V = \frac{1}{N}\sum_{t=1}^{T}(\mathcal{R}_t - V(s_t))^2$$

Henceforth, we will refer to this full architecture containing all models outlined in Section 6 as the ROVR architecture.

Table 1. Local Net Performance vs. Toy Conditioning Policy

Conditioning Policy	Avg. MSE Per Frame	STDEV of MSE Per Frame
Best Contextual Frames	16,933	1,594
Worst Contextual Frames	20,406	1,257
Random Contextual Frames	19,827	1,655

Table 2. ROVR Performance vs Baselines

Conditioning Policy	Average Perceptual Loss	Optical Flow Recovery
ROVR Runs		
Optical Flow Reward & Cold Start	19,156	0.598
No Optical Flow Reward & Warm Start	19,348	0.434
Optical Flow Reward & Warm Start	**18,681**	**0.627**
Baseline Runs		
Imitation Learning	19,270	0.494
Sequential Conditioning	19,147	0.478

7. Results

7.1. Pre-Train Local-Net

We begin by evaluating the performance of our local network Γ, which is pretrained as described in Section 6.1.

Table 1 displays the results of our evaluation of the pre-trained local network Γ, where we can observe the influence of the quality of conditioning frames on the reconstruction error. Recall from Section 4 that our masking strategy allows us to deterministically recover the best and worst frames to condition on for a given video and target frame within that video. Thus, we feed the best conditioning frames to our local network Γ and observe that the network achieves much lower Perceptual Loss than the randomly chosen conditioning frames. When we feed the worst possible conditioning frames, Perceptual Loss exceeds that of Γ evaluated with randomly chosen conditioning frames and the best possible conditoining frames. Thus, we have demonstrated that local network is an unbiased estimator of the *quality* of conditioning frames chosen for a given target frame.

7.2. Warm-Start Imitation Learning

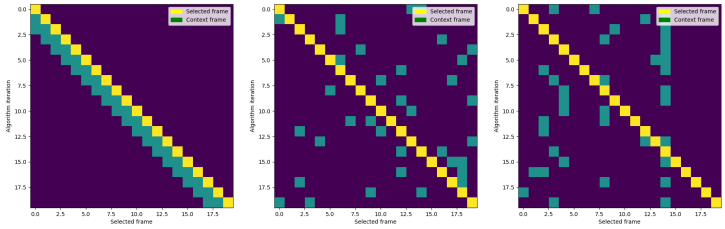
As detailed in Section 6.2, our policy network π is initially pre-trained using imitation learning. The performance of this policy is documented in Table 2. Comparing with Table 1, we see that at the end of imitation learning, the network's Perceptual Loss is marginally better than that of random conditioning frames. We discuss the implications of this warm-start in Section 8.

7.3. ROVR Architecture

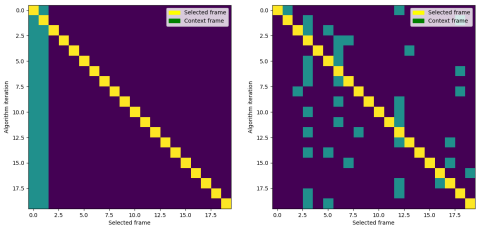
We perform training and testing runs with the full ROVR architecture. Specifically, we train three policies: a) π_1 with no warm-start and the global optical flow reconstruction reward detailed in Section 5 b) π_2 with a warm start and no global optical flow reconstruction reward, and c) π_3 with a warm start and the global optical flow reconstruction reward. Evidently, π_3 is the top-performing model, with the lowest average Perceptual Loss and highest optical flow reconstruction reconstruction score. Further, while all three policies are comparable on average Perceptual Loss, π_2 achieves a markedly worse optical flow reconstruction score – indicating that π_1 and π_3 were able to learn from the sparse, global optical flow reward. We discuss these results further in Section 8.

Figure 3 demonstrates the different conditioning frames selected for a test video across all five conditioning policies we evaluate (listed in Table 2). Interestingly, our policy trained with optical flow reward finds the unintuitive local-minimum of conditioning on the first two frames of the video throughout all timesteps. This behavior is not reflected in other testing videos, demonstrating ROVR's ability to discover unintuitive patterns specific to a given input video.

Figure 6 demonstrates some sample outputs of π optimized with simple imitation learning versus π optimized with the ROVR architecture. The rightmost column is most informative – displaying the relatively larger amount of context our policy optimized with the ROVR architecture is able to recover compared to the policy learnt with simple imitation learning. This trend is clear across all testing-images, especially the first frame.



(a) Sequential Conditioning (b) Imitation Learning (c) Warm Start



(d) Optical Flow Reward & No Warm Start (e) Optical Flow Reward & Warm Start

Figure 3. These heatmaps indicate which frames the policy conditioned on at each timestep. Yellow boxes indicate the target frame (which is diagonal given that we edit frames autoregressively) while green boxes indicate the selected context frames at each timestep.

8. Conclusions and Further Work

Our overall results show that our network beat the sequential Vid2Vid baseline (see Table 1 and Table 2). Our ablation studies suggest two additional conclusions related to this.

First, pretraining π with Imitation Learning seems to benefit the fully Reinforcement Learning section of training. It is interesting to note that the network hasn't necessarily learned much directly after pretraining, even though the expert actions it was shown continue to be valid choices during inference time. Instead, pretraining only seems to show much value later into the full end-to-end session, as we can see when we compare pi_3 and pi_1's results. This suggests that the model only had time to learn some low level features during imitation learning (which might mean that we should allow it to pretrain for longer/with more data) and/or there is something wrong with our IL paradigm. Specifically, we think that BCE loss and multi-label classification in general were not the right way to formulate this problem. As discussed in Section 6, optimizing IL with regards to BCE assumes that the output labels are independent events, which is untrue. This might be causing the poor performance seen during Imitation Learning, and lends credence to the idea that the network is not learning full decision-making strategies but is learning low level features that benefit it during Reinforcement Learning, when the independent-labels assumption is relaxed.

Our second main conclusion is that this network architecture can learn to optimize for a global metric, in this case optical flow reconstruction. As discussed in our results, the inclusion of this metric directly led to far greater performance on it, which means that the policy network isn't regressing to simple supervised learning at every time step but is actually learning temporal dependencies across the trajectory. Another interesting point is that the full ROVR network performs poorly on **Perceptual Loss** with the warm start and no optical flow objective, but outperforms the optical flow baseline when both optical flow and warm start are included. This seems to suggest that a global objective is required for the warm start to be useful, perhaps because the BCE-based imitation learning learns some incorrect decisions that require extra reward signal to break, or simply because the model learns important details about local reconstruction when it prioritizes global reconstruction (which makes some sense, given that the optical flow reconstruction is another way of incentivizing approaching the original video).

There are several directions for future work.

First, the choice of conditioning frames c_{t_1}, c_{t_2} should be changed from a multi-label classification task to a ranking or retrieval task. This will let us get rid of BCE and its onerous independent label assumption and replace it with contrastive loss or some similar objective. This should lead to much better imitation learning performance, as the model will be able to properly learn from the positive/negative labels we assign to pairs.

Second, other global rewards such as spatio-temporal objective should be tested. The NVIDIA Vid2Vid

paper trained a spatio-temporal observer on ground truth videos before predicting consistency on translated videos. This is a much more sophisticated objective than our own Optical Flow Reconstruction, which uses a pre-trained RAFT network included in Pytorch.

Third, and related to the first direction, is reformatting the goal of choosing 2 conditioning frames into choosing k conditioning frames where $k < n$ for some constant n and then reducing the reward given at a time step by some function $f(k)$. This would give the policy network more agency in collecting information for the local network to work with, and might improve the ability of the model to get reward signal if it can properly balance these objectives. This would also be a generally interesting research direction since it would demonstrate new capabilities.

Finally, we were originally planning on adding more corruption strategies such as random noising and brightness fluctuations, to match more potential real world use cases (one can imagine wishing to automatically fix bad HDR or blur in a consumer's video). We did not end up testing these strategies since the masking strategy proved hard enough for the network to learn, but it would be interesting to append these to the task description.

9. Contributions

We (Aaditya, Arnuv, Arjun) contributed equally towards coming up with and implementing the ideas in this paper. In determining our experimental approach, Aaditya found prior work on video translation using both attention-based and convolution-based mechanisms, Arnuv identified the opportunity for RL guidance and codified the MDP, and Arjun researched frame-level style transfer via U-Nets and attention-based mechanisms. With respect to developing the project, Aaditya built out code for PPG training, dataset handling, and expert trajectory generation for immitation learning; Arnuv architectured the policy networks, wrote the video corruption strategy, and produced visualizations from the performance of our models, and Arjun developed and trained the local inpainting network, built out and ran an immitation learning-based warm start for our policy network, and managed MLOps infrastructure for experiment and data logging. Together, we wrote, debugged, and fixed the training of our RL model through PPG.

We owe many thanks to Ansh Khurana for his invaluable mentorship and advice throughout our work.

References

[1] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models, 2023.

[2] Qingxing Cao, Liang Lin, Yukai Shi, Xiaodan Liang, and Guanbin Li. Attention-aware face hallucination via deep reinforcement learning, 2017.

[3] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[4] Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[5] Feiyan Huang, Shangyou Zeng, Jie Ke, Songtong Lei, and JinJin Wang. A video description model with improved attention mechanism. *Journal of Physics: Conference Series*, 2384(1):012015, Dec. 2022.

[6] Hao Liang, Shuai Yang, Wenjing Wang, and Jiaying Liu. Instance-aware coherent video style transfer for chinese ink wash painting. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 823–829. International Joint Conferences on Artificial Intelligence Organization, 8 2021. Main Track.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[8] Ting-Chun Wang, Ming-Yu Liu, Andrew Tao, Guilin Liu, Bryan Catanzaro, and Jan Kautz. Few-shot video-to-video synthesis. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[9] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

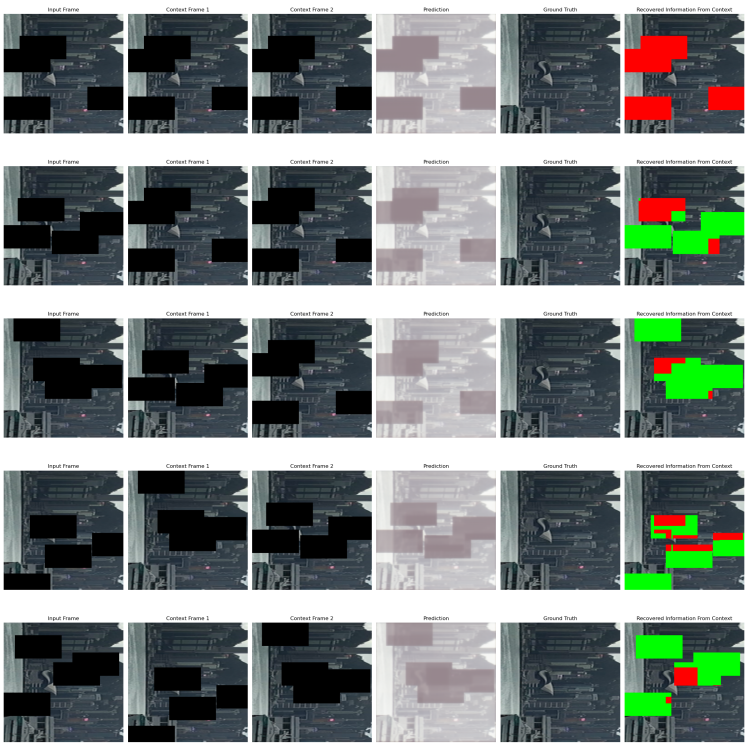


Figure 4. Imitation Learning

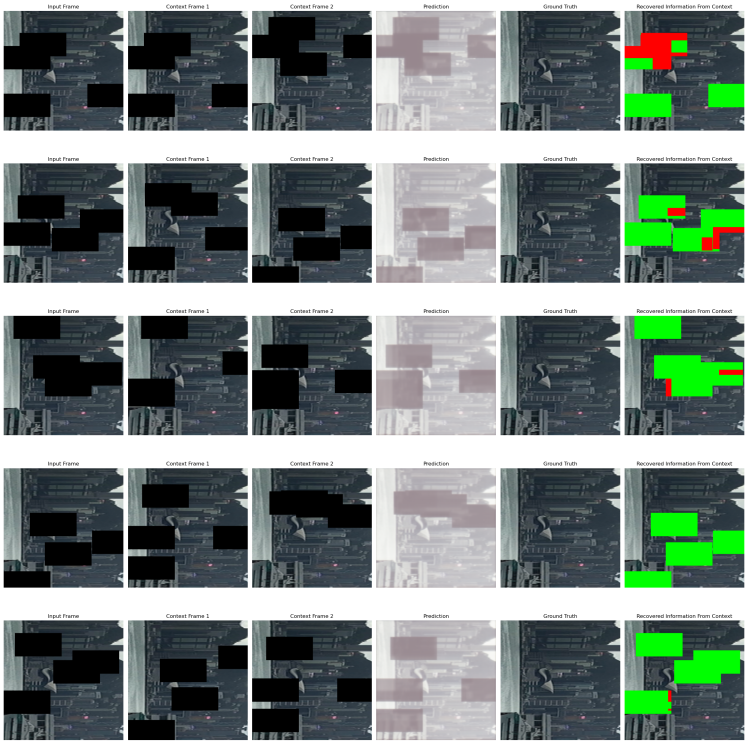


Figure 5. ROVR (Optical Flow Reward & Warm Start)

Figure 6. Performance of policy learned via imitation learning vs policy learned via ROVR against five testing frames. Green indicates areas that are masked out in the input frame, but not masked out in one (or both) context frames. Red indicates areas that are masked out in the input image and both context frames. Green, therefore, represents useful context derived from the context frames for in-painting the mask.