Accelerating Video Inverse Problem Solvers with Autoregressive Diffusion Models

Taesung Kwon[1*] **Jonghyun Park**[1*] **Hyungjin Chung**[2†] **Jong Chul Ye**[1†]
[1]KAIST [2]Korea University

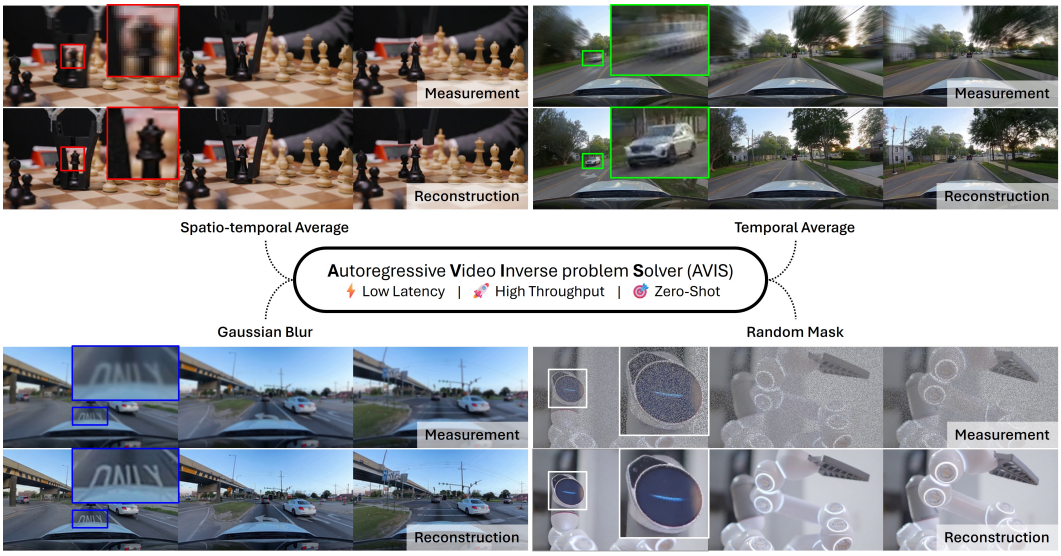

Figure 1: The AVIS framework leverages autoregressive video diffusion models to restore videos in a streaming manner, naturally eliminating latency bottlenecks. Examples show diverse video inverse problems that AVIS can solve in a zero-shot manner.

Abstract

Diffusion models provide powerful priors for zero-shot video inverse problems, but their real-time deployment is hindered by two inefficiencies: high initial latency caused by holistic video restoration, and low throughput resulting from multiple VAE passes to enforce measurement consistency in pixel space. To overcome these limitations, we propose Autoregressive Video Inverse problem Solver (AVIS). The AVIS framework leverages autoregressive video diffusion models to restore videos in a streaming manner, naturally eliminating latency bottlenecks. Specifically, AVIS initializes reverse diffusion with a measurement-consistent estimate, reducing the required sampling steps. Compared to leading non-autoregressive solvers, AVIS drastically reduces initial latency from 114s to 4s and increases throughput from 0.71 to 1.18 FPS while achieving superior restoration quality. We further introduce a highly accelerated variant, dubbed AVIS *Flash*, that enforces measurement consistency solely on the *first* chunk. AVIS *Flash* substantially boosts throughput to 5.91 FPS on a single RTX 4090 GPU while maintaining competitive performance and achieving a favorable efficiency–performance trade-off, paving the way toward real-time deployment.

[*]Equal contribution. [†]Corresponding authors.

Project page and code are available here.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

1 Introduction

Video inverse problems aim to reconstruct a clean video $\mathbf{x}$ from a degraded measurement $\mathbf{y}$ (*e.g.*, low-resolution, masked, or blurred). The degradation process is mathematically formulated as

$$\mathbf{y} = \mathcal{A}(\mathbf{x}) + \boldsymbol{n}, \tag{1}$$

where $\mathcal{A}$ is a degradation operator and $\boldsymbol{n} \sim \mathcal{N}(0, \sigma_n^2 I)$ denotes measurement noise. These video restoration tasks are ill-posed, as a single measurement $\mathbf{y}$ can correspond to many possible clean videos $\mathbf{x}$. To resolve this ambiguity, diffusion and flow-based models [1–9] have recently emerged as powerful priors for solving video inverse problems in a zero-shot manner [10–15], producing results that are both perceptually realistic and faithful to the measurements. Despite their successful restoration performance, these solvers remain far from real-time deployment due to two inefficiencies:

1. **High initial latency:** Recent Diffusion model-based Video Inverse problem Solvers (DVIS) [11–15] restore the entire video in a holistic manner, sampling the entire video sequence simultaneously. Consequently, the first frame cannot be displayed until the last frame has been restored, introducing latency equal to the time required to restore all frames.
2. **Low throughput:** Modern diffusion and flow-based video models [16–20] operate in the latent space of a variational autoencoder (VAE) [21] for computational efficiency. Recent DVIS based on such models [13, 15] require multiple VAE passes during restoration, restricting throughput to below 1 FPS. While some works [22, 23] bypass this with auxiliary latent operators, they require degradation-specific training, sacrificing zero-shot generality.

Here, we identify a promising yet underexplored direction toward the efficient deployment of DVIS: harnessing the potential of autoregressive (AR) video diffusion models [24–26]. Unlike non-autoregressive solvers that process the entire video jointly (Figure 2(a)), autoregressive solvers restore videos in a streaming manner, naturally eliminating the latency bottleneck. However, simply adopting an AR backbone does not automatically resolve the issue of low throughput, as it still requires multiple VAE passes for measurement consistency updates.

To overcome this limitation, we propose **A**utoregressive **V**ideo **I**nverse problem **S**olver (**AVIS**), which accelerates restoration by initializing the reverse diffusion process with a measurement-consistent estimate, thereby reducing the required sampling steps. We draw inspiration from CCDF [27], which demonstrates that initializing the reverse diffusion process from a coarse estimate, predicted by an auxiliary pretrained restorer, exploits stochastic contraction to significantly reduce the required sampling steps. Unlike CCDF, AVIS does not rely on an auxiliary pretrained restorer; instead, it obtains an initial restoration by directly minimizing the measurement consistency objective. By utilizing this estimate to initialize the reverse diffusion, AVIS begins the restoration process from a better starting point while enforcing measurement consistency for *every* video chunk (Figure 2(b)).

To push efficiency further, we introduce a highly accelerated variant, **AVIS *Flash***. Unlike AVIS, which applies iterative measurement guidance to every chunk, AVIS *Flash* enforces measurement consistency only on the *first* video chunk. Interestingly, we observe that this is sufficient to restore the entire sequence (Figure 2(c)). Once the first chunk is fully restored and accurately grounded to the measurement, subsequent chunks can be generated through autoregressive propagation from this corrected prefix, completely bypassing explicit measurement guidance. This eliminates iterative VAE passes for subsequent chunks, accelerating restoration by 5$\times$ while preserving zero-shot performance.

Compared to the leading non-autoregressive solver [15], AVIS reduces initial latency from 114s to 4s and improves throughput from 0.71 to 1.18 FPS while achieving superior restoration quality. Pushing efficiency further, AVIS *Flash* substantially boosts throughput to 5.91 FPS on a single RTX 4090 GPU while maintaining competitive performance, paving the way toward real-time deployment.

Our main contributions can be summarized as follows:

- **AVIS.** We investigate the potential of adopting autoregressive video diffusion models for zero-shot video restoration. AVIS accelerates restoration by initializing the reverse diffusion process with a measurement-consistent estimate, reducing the required sampling steps while enforcing measurement consistency for every video chunk.
- **AVIS *Flash*.** We introduce a highly accelerated variant of AVIS that retains the initialization while enforcing measurement consistency solely on the first video chunk. This substantially increases throughput, paving the way for real-time video inverse problem solvers.

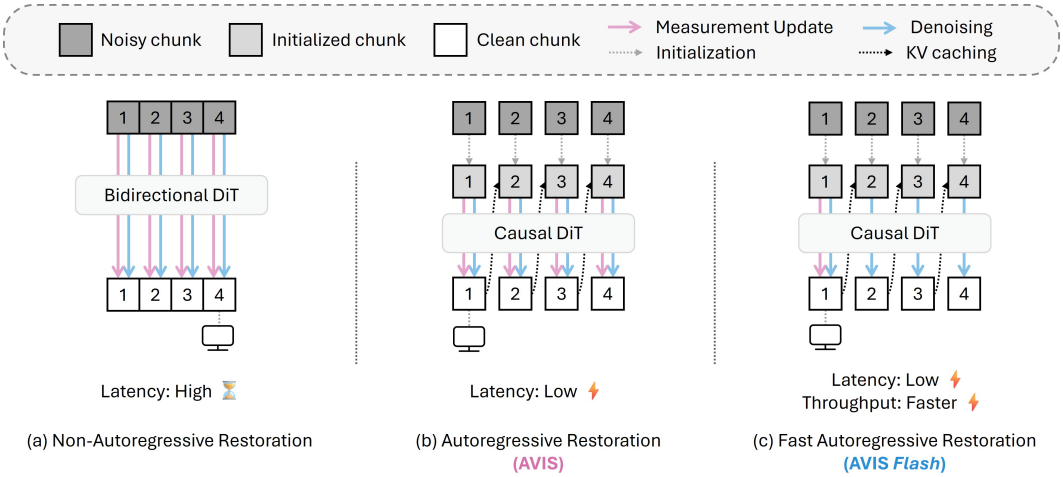

Figure 2: **Overview of our proposed AVIS and AVIS *Flash* framework.** (a) Non-autoregressive restoration processes the entire video holistically, suffering from high initial latency. (b) AVIS restores videos in a streaming manner and reduces sampling steps via measurement-consistent initialization while enforcing measurement updates for every video chunk. (c) AVIS *Flash* retains the same initialization as AVIS but applies measurement updates only to the first video chunk. Subsequent chunks ($n \geq 2$) are restored through autoregressive propagation from the corrected prefix, eliminating iterative VAE passes and dramatically accelerating throughput.

2 Related Work

2.1 Diffusion for Video Inverse Problems

Diffusion model-based Inverse problem Solvers (DIS) [28–37] enable repurposing the diffusion or flow-based models [1–4, 6–9] originally trained to model the data distribution $p(\mathbf{x})$ as powerful plug-and-play priors. By guiding the sampling trajectory toward the posterior $p(\mathbf{x}|\mathbf{y})$, these methods achieve strong zero-shot performance across a wide range of image restoration tasks, producing solutions that are perceptually natural and consistent with the measurement $\mathbf{y}$.

Motivated by this success, Diffusion model-based Video Inverse problem Solvers (DVIS) [11–15] extend this paradigm to the video modality. One line of work [11–13] repurposes image diffusion priors [3, 6] for video restoration by introducing explicit temporal conditioning. Warping-based approaches [10, 11] explicitly control temporal consistency using optical flow [38]. An alternative approach enforces temporal consistency without optical flow through batch-consistent sampling [12, 13], which efficiently synchronizes stochastic components across frames. More recently, methods leveraging native video diffusion priors [14, 15] have shown an improved ability to capture temporal correlations without requiring explicit temporal conditioning.

As mentioned earlier, all of these solvers restore the video holistically by sampling all frames simultaneously. As a result, the first frame cannot be displayed until the entire sequence is fully restored, introducing latency equal to the time required to restore all frames. We pinpoint this issue as a critical barrier for real-time deployment and propose a direction toward efficient DVIS.

2.2 Autoregressive Diffusion Models for Video Generation

In recent years, both closed- and open-source models [39, 40, 16–20] have demonstrated remarkable progress in foundational video generation. Currently, the dominant open-source video diffusion models [16–18, 20] typically adopt diffusion transformers [41] with bidirectional attention [42] that generate the entire video sequence simultaneously.

Recent works [24–26, 43–46] have explored autoregressive (AR) modeling, which generates the next frame by conditioning on previously generated ones, enabling low-latency generation. One representative direction is distilling diffusion transformers with bidirectional attention into student transformers with causal attention, allowing support for key-value (KV) caching during inference [25, 26]. Among these, Self Forcing [26] has emerged as a leading autoregressive video diffusion model.

It addresses the train-test gap by conditioning on its own previously generated outputs rather than ground-truth frames during training, preventing error accumulation (*i.e.*, exposure bias) at test time. Given its effectiveness in closing this gap and its KV caching efficiency, we adopt Self Forcing as the backbone of our framework.

3 Preliminaries: Autoregressive Video Diffusion Models

Autoregressive (AR) modeling. AR video diffusion models combine sequential generation along the temporal axis with denoising diffusion processes. Specifically, let $\mathbf{z} = [\mathbf{z}^1, \mathbf{z}^2, \ldots, \mathbf{z}^N]$ be a sequence of N video chunks, where each *chunk* refers to a block corresponding to a short sequence of consecutive latent frames. Rather than modeling the entire video at once, the model builds the sequence progressively, predicting each new video chunk conditioned on previously generated ones. Mathematically, the joint distribution of the sequence is factorized sequentially as follows:

$$p(\mathbf{z}) = \prod_{n=1}^{N} p(\mathbf{z}^n | \mathbf{z}^{<n}). \tag{2}$$

Each conditional distribution $p(\mathbf{z}^n|\mathbf{z}^{<n})$ is then parameterized by a diffusion model, where $\mathbf{z}^n$ is generated by gradually denoising the initial Gaussian noise, conditioned on the past context $\mathbf{z}^{<n}$, with the initial context $\mathbf{z}^0 = \emptyset$.

Diffusion process. To model the conditional distribution $p(\mathbf{z}^n|\mathbf{z}^{<n})$, we employ the flow matching framework [7–9], which offers a simple and deterministic ordinary differential equation (ODE) formulation for the diffusion process. Specifically, let $\mathbf{z}_0^n$ denote the clean n-th video chunk and $\mathbf{z}_1^n \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ be pure Gaussian noise. For a continuous timestep $t \in [0, 1]$, the intermediate noisy state $\mathbf{z}_t^n$ is formulated as a linear interpolation:

$$\mathbf{z}_t^n = (1-t)\mathbf{z}_0^n + t\mathbf{z}_1^n. \tag{3}$$

The corresponding ground-truth vector field driving this transformation is defined as $\mathbf{v}_t^n = \mathbf{z}_1^n - \mathbf{z}_0^n$.

During training, the model $\mathbf{v}_\theta$ is optimized through conditional flow matching to regress this vector field using a mean squared error (MSE) objective:

$$\mathcal{L}_{\text{CFM}} = \mathbb{E}_{t, \mathbf{z}_0^n \sim p_{\text{data}}, \mathbf{z}_t^n \sim p_t(\mathbf{z}_t^n|\mathbf{z}_0^n)} \|\mathbf{v}_\theta(\mathbf{z}_t^n, t; \mathbf{z}_0^{<n}) - \mathbf{v}_t^n\|_2^2, \tag{4}$$

where $p_t(\mathbf{z}_t^n|\mathbf{z}_0^n)$ denotes the time-dependent probability path induced by the linear interpolation in Eq. (3). Then, the minimizer of this objective corresponds to the conditional expectation:

$$\mathbf{v}_\theta(\mathbf{z}_t^n, t; \mathbf{z}_0^{<n}) = \mathbb{E}[\mathbf{z}_1^n - \mathbf{z}_0^n | \mathbf{z}_t^n, \mathbf{z}_0^{<n}]. \tag{5}$$

At inference, given a noisy video chunk $\mathbf{z}_t^n$ at timestep t, the model predicts the vector field $\mathbf{v}_\theta(\mathbf{z}_t^n, t; \mathbf{z}_0^{<n})$. By rearranging Eq. (3), we can obtain a clean estimate $\hat{\mathbf{z}}_{0|t}^n$, which approximates the posterior mean $\mathbb{E}[\mathbf{z}_0^n \mid \mathbf{z}_t^n, \mathbf{z}_0^{<n}]$:

$$\hat{\mathbf{z}}_{0|t}^n := \mathbb{E}[\mathbf{z}_0^n \mid \mathbf{z}_t^n, \mathbf{z}_0^{<n}] = \mathbf{z}_t^n - t\mathbf{v}_\theta(\mathbf{z}_t^n, t; \mathbf{z}_0^{<n}). \tag{6}$$

In practice, to avoid the computational overhead of processing the full past context at each step, the conditioning on $\mathbf{z}_0^{<n}$ is efficiently implemented using its KV cache [25, 26], denoted as $\text{KV}^{<n}$. Subsequently, following the specific sampling procedure of our backbone [26], the process continues to the next timestep $t - \Delta t$ by re-noising the clean estimate $\hat{\mathbf{z}}_{0|t}^n$ from Eq. (6):

$$\mathbf{z}_{t-\Delta t}^n = (1 - (t - \Delta t))\hat{\mathbf{z}}_{0|t}^n + (t - \Delta t)\mathbf{z}_1^n, \quad \mathbf{z}_1^n \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{7}$$

4 AVIS: Autoregressive Video Inverse Problem Solver

In this section, we detail the architecture of our Autoregressive Video Inverse problem Solver (AVIS) and its highly efficient variant, AVIS *Flash*, designed to harness the underexplored potential of AR video diffusion models for zero-shot restoration. We begin by introducing an efficient mechanism to enforce measurement consistency during the reverse diffusion process in Section 4.1. Building upon this, Section 4.2 presents our core framework, AVIS, which accelerates the process by initializing the reverse diffusion with a measurement-consistent estimate, thereby reducing the required sampling steps. Finally, in Section 4.3, we propose AVIS *Flash*, which pushes efficiency further by eliminating multiple VAE passes for subsequent video chunks.

4.1 Guidance for Measurement Consistency

During the reverse diffusion process of the n-th noisy video chunk $\mathbf{z}_t^n$, measurement consistency is enforced at each denoising step to guide the sampling trajectory toward the posterior. The conventional approach is to compute the gradient of the measurement consistency term [14, 28] (*e.g.*, $\nabla_{\mathbf{z}}\|\mathbf{y} - \mathcal{A}(\mathcal{D}(\mathbf{z}))\|^2$). However, this requires backpropagation through both the VAE decoder $\mathcal{D}$ and the model $\mathbf{v}_\theta$, which is computationally prohibitive for high-dimensional video data.

To ensure both efficiency and generalizability, we follow the common practice of recent DVIS [12, 13, 15] by adopting a multi-step optimization algorithm for the update. Specifically, we employ the Decomposed Diffusion Sampling (DDS) [31] framework, which utilizes the conjugate gradient (CG) method [47, 48] to efficiently enforce measurement consistency.

Formally, at each timestep t, we first obtain the clean latent estimate $\hat{\mathbf{z}}_{0|t}^n$ from $\mathbf{z}_t^n$ using Eq. (6) and decode it into the pixel space:

$$\hat{\mathbf{x}}_{0|t}^n = \mathcal{D}(\hat{\mathbf{z}}_{0|t}^n). \tag{8}$$

Next, we enforce data consistency by solving the following proximal optimization problem through the aforementioned CG method:

$$\tilde{\mathbf{x}}_{0|t}^n := \arg\min_{\mathbf{x}} \frac{\gamma}{2}\|\mathbf{y} - \mathcal{A}(\mathbf{x})\|^2 + \frac{1}{2}\|\mathbf{x} - \hat{\mathbf{x}}_{0|t}^n\|^2. \tag{9}$$

To avoid complex parameter tuning, we fix $\gamma = 1$ throughout this paper. We then re-encode the updated pixel-space estimate $\tilde{\mathbf{x}}_{0|t}^n$ back into the latent space:

$$\tilde{\mathbf{z}}_{0|t}^n = \mathcal{E}(\tilde{\mathbf{x}}_{0|t}^n). \tag{10}$$

By doing so, we enforce measurement consistency using only forward passes of $\mathcal{D}$ and $\mathcal{E}$, entirely avoiding the expensive neural network Jacobian computation. Finally, the updated latent estimate $\tilde{\mathbf{z}}_{0|t}^n$ is renoised back to the next timestep $t - \Delta t$ using Eq. (7),

$$\mathbf{z}_{t-\Delta t}^n = (1 - (t - \Delta t))\tilde{\mathbf{z}}_{0|t}^n + (t - \Delta t)\mathbf{z}_1^n, \quad \mathbf{z}_1^n \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{11}$$

to continue the reverse process.

4.2 AVIS: Accelerating DVIS with a Better Starting Point

Diffusion models are inherently slow, as they require numerous steps to generate data from pure Gaussian noise. To accelerate solving inverse problems, CCDF [27] proposes initializing the reverse diffusion process from a *better* starting point. Specifically, it obtains a coarse estimate from an auxiliary pre-trained network, diffuses it to an intermediate timestep $t_0 < 1$, and starts the reverse sampling from there. By starting at t_0, CCDF effectively reduces the total number of sampling steps, thereby significantly reducing inference time while leveraging the stochastic contraction property of the reverse diffusion path.

However, its reliance on a task-specific auxiliary network for the initial correction fundamentally sacrifices the zero-shot generality of the solver. To overcome this limitation, we propose a training-free, multi-step optimization strategy to obtain a better starting point, preserving the acceleration benefits of the CCDF principle.

Specifically, we obtain the starting point of the reverse sampling as follows. We first compute a measurement-consistent initial estimate $\mathbf{x}_{\text{init}}^\lambda$ in pixel space using a multi-step optimization algorithm (*e.g.*, CG) that minimizes the measurement consistency objective:

$$\mathbf{x}_{\text{init}}^\lambda = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{y} - \mathcal{A}(\mathbf{x})\|_2^2 + \frac{\lambda}{2}\|\mathbf{x} - \mathbf{x}_{\text{ref}}\|_2^2, \tag{12}$$

where $\mathbf{x}_{\text{ref}}$ denotes the dimension-aligned measurement, *i.e.*, $\mathcal{A}^\top(\mathbf{y})$. Next, we encode $\mathbf{x}_{\text{init}}^\lambda$ to get $\mathbf{z}_{\text{init}}$ and diffuse it to timestep t_0:

$$\mathbf{z}_{t_0} = [\mathbf{z}_{t_0}^1, \mathbf{z}_{t_0}^2, \ldots, \mathbf{z}_{t_0}^N] = (1 - t_0)\mathbf{z}_{\text{init}} + t_0\mathbf{z}_1, \quad \mathbf{z}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{13}$$

While this initial estimate $\mathbf{x}_{\text{init}}^\lambda$ satisfies measurement consistency, it is only a coarse estimate and may lack perceptual realism. The role of the reverse diffusion process is to transport this estimate

onto the high-density region of the video data manifold. To this end, we perform K sampling steps starting from t_0, yielding a step size of $\Delta t = t_0/K$ for each reverse diffusion process. During this process, we continuously apply DDS guidance at each step to ensure the trajectory remains strictly faithful to the given measurement. By integrating the guidance and the accelerated sampling strategy into the autoregressive diffusion process, we establish our complete framework: the **A**utoregressive **V**ideo **I**nverse problem **S**olver (**AVIS**).

4.3 AVIS *Flash*: Accelerating AVIS with Autoregressive Propagation

Here, we propose AVIS *Flash*, a highly accelerated variant that fully unlocks the potential of autoregressive propagation. While it shares the same initialization at t_0 with AVIS, it fundamentally eliminates the need for iterative measurement guidance after the first chunk. Although the guidance introduced in Section 4.1 avoids expensive neural network Jacobian computations, applying it to every video chunk requires iterative VAE passes at each reverse step, which remains a bottleneck for high-throughput video restoration.

Empirically, we discover a strikingly simple yet effective solution: bypassing measurement guidance for all subsequent chunks ($n \geq 2$) dramatically accelerates throughput with competitive restoration performance. We attribute this success to the temporal context propagation induced by the autoregressive formulation, combined with the initialization. Once the first video chunk ($n = 1$) is grounded to the measurements through guidance, its restored features are extracted and stored as a prefix cache KV^1. Furthermore, since the initialization in Section 4.2 provides a measurement-consistent estimate that is then re-noised to an intermediate timestep t_0, the model does not need to generate subsequent chunks ($n \geq 2$) from scratch; it simply refines this estimate through the guidance-free sampling steps in Eq. (6) and Eq. (7), conditioned on the accumulated prefix cache $\mathrm{KV}^{<n}$.

To clarify the intuition behind AVIS *Flash*, we provide an error decomposition within our framework. Specifically, Proposition 1 decomposes the final error into two terms: the initial error from the measurement-consistent estimate, and the context error propagated from previously restored chunks. Detailed derivations are provided in Appendix A.

Proposition 1. *Let ϵ_0^n denote the initial error of the current chunk, and let δ_n denote the context error propagated from previously restored chunks. Under local Lipschitz assumptions on the autoregressive vector field $\mathbf{v}_\theta$, the final error ϵ_K^n of the n-th chunk after K sampling steps satisfies*

$$\epsilon_K^n \leq \Lambda_K \epsilon_0^n + B_K \delta_n, \tag{14}$$

where Λ_K and B_K depend on the sampling schedule and local Lipschitz constants.

This decomposition provides an interpretive view for understanding the design of AVIS *Flash*. The measurement-consistent initialization is designed to reduce the per-chunk initial error ϵ_0^n before autoregressive sampling. Meanwhile, the context error δ_n contributes to the bound as an additive perturbation term. This indicates that the context error propagated from previous chunks affects the current chunk through incremental additions rather than through multiplicative amplification of the initial error.

AVIS *Flash* can also be naturally extended to handle long video sequences. By periodically re-injecting measurement consistency (*e.g.*, applying guidance every N chunks), we can further suppress error accumulation. This periodic guidance effectively mitigates drift, allowing for more stable restoration over extended video sequences, as demonstrated in Appendix C. The complete AVIS *Flash* pipeline is detailed in Algorithm 1.

5 Experiments

5.1 Experimental Setup

Dataset & Inverse Problems. We evaluate our method on 100 high-resolution videos sourced from Pexels [49]. Each video is spatially resized to 480×832 and temporally cropped to the initial 81 frames. We comprehensively evaluate our method on five video restoration tasks: (1) *Super-Resolution* at a $4\times$ scaling factor, (2) *Random Inpainting* with 50% of the video masked, (3) *Gaussian Deblur* with a kernel size of 61 and a σ of 3.0, (4) *Temporal Average* using $7\times$ averaging over a

Algorithm 1 Autoregressive Video Inverse problem Solver (AVIS and AVIS *Flash*)

Require: Measurement $\mathbf{y}$, operator $\mathcal{A}$, diffusion model $\mathbf{v}_\theta$, number of chunks N, start time t_0, encoder $\mathcal{E}$, decoder $\mathcal{D}$, and mode $\in \{\text{AVIS}, \text{AVIS } \textit{Flash}\}$.

$\mathbf{x}_{\text{init}}^{\lambda} \leftarrow$ Solve Eq. (12) **and** $\mathbf{z}_{\text{init}} \leftarrow \mathcal{E}(\mathbf{x}_{\text{init}}^{\lambda})$ ▷ Pre-restoration
$\text{KV}^0 \leftarrow \emptyset$
for $n = 1 : N$ **do**
  $\mathbf{z}_{t_0}^n \leftarrow (1 - t_0)\mathbf{z}_{\text{init}}^n + t_0\mathbf{z}_1, \quad \mathbf{z}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ▷ Initialization
  for $t : t_0 \to 0$ **do**
    $\hat{\mathbf{z}}_{0|t}^n \leftarrow \mathbf{z}_t^n - t\mathbf{v}_\theta(\mathbf{z}_t^n, t; \text{KV}^{<n})$ ▷ Obtain clean estimate
    if mode **is** AVIS **or** (mode **is** AVIS *Flash* and $n = 1$) **then**
      $\hat{\mathbf{z}}_{0|t}^n \leftarrow$ Solve Eq. (9) via CG ▷ Measurement consistency update
    end if
    $\mathbf{z}_{t-\Delta t}^n \leftarrow (1 - (t - \Delta t))\hat{\mathbf{z}}_{0|t}^n + (t - \Delta t)\mathbf{z}_1, \quad \mathbf{z}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ▷ Re-noise
  end for
  Display $\mathcal{D}(\mathbf{z}_0^n)$ **and** $\text{KV}^{\leq n} \leftarrow$ Update $\text{KV}(\mathbf{z}_0^n, \text{KV}^{<n})$
end for
return $[\mathbf{z}_0^1, \ldots, \mathbf{z}_0^N]$

7-frame causal window, and (5) *Spatio-Temporal Average* using 4× spatial average pooling and 4× temporal averaging over a 4-frame causal window.

Baselines. We compare our methods (AVIS and AVIS *Flash*) against other diffusion model-based solvers for high-resolution video inverse problems: DiffIR2VR-Zero [50], VISION-XL [13], and LVTINO [15]. We include DiffIR2VR-Zero specifically for super-resolution (SR), as it only supports SR among the various inverse problems addressed in this work. For completeness, detailed descriptions of each baseline method are provided in Appendix B.

Metrics & Implementation Details. To assess restoration quality, we report five fidelity and perceptual metrics: PSNR, SSIM [51], LPIPS [52], FID [53], and FVD [54]. We further evaluate video quality using VBench [55], specifically focusing on Subject Consistency (Sub. Con.), Background Consistency (Bg. Con.), Motion Smoothness (M. Smooth.), Aesthetic Quality (Aesth.), and Imaging Quality (Imag.). Computational efficiency is measured in terms of latency (time to restore the first frame in seconds), total restoration time, and throughput (FPS). Unless otherwise specified, all experiments are conducted on a single NVIDIA RTX 4090 GPU. Based on our ablation studies, we employ $t_0 = 0.1$ and $K = 2$ as our default settings. Please refer to Appendix B for more information.

5.2 Results

Computational Efficiency. The primary advantage of our framework is its practical efficiency. As shown in Table 1, both AVIS and AVIS *Flash* reduce the initial latency to only 4 seconds, whereas the baselines suffer from severe delays (167 seconds for VISION-XL and 114 seconds for LVTINO). Beyond latency improvements, our framework significantly enhances overall throughput. While the baselines fail to surpass 1 FPS, AVIS successfully achieves over 1 FPS. Notably, AVIS *Flash* achieves 5.91 FPS, making it over 12× and 8× faster than VISION-XL and LVTINO, respectively. Furthermore, we found that deploying AVIS *Flash* on a single NVIDIA H100 GPU pushes the performance even further to an initial latency of 1.85 seconds and a throughput of 10.2 FPS, paving the way toward real-time deployment.

Restoration Performance. Table 2 compares AVIS and AVIS *Flash* against baselines across five restoration tasks. Overall, AVIS achieves the best restoration performance, outperforming or

Method	Latency (s) ↓	Time (s) ↓	FPS (frame/s) ↑
DiffIR2VR-Zero	1300	1300	0.06
VISION-XL	167	167	0.49
LVTINO	<u>114</u>	114	0.71
AVIS	**4**	<u>68.5</u>	<u>1.18</u>
AVIS *Flash*	**4**	**13.7**	**5.91**

Table 1: **Efficiency comparison for 4× video super-resolution.** Our framework (AVIS and AVIS *Flash*) achieves significant improvements across all efficiency metrics. **Bold** and <u>underline</u> indicate the best and second-best results, respectively.

Method	PSNR↑	SSIM↑	LPIPS↓	FVD↓	FID↓	Sub. Con.↑	Bg. Con.↑	M. Smooth.↑	Aesth.↑	Imag.↑
Super Resolution										
DiffIR2VR-Zero	27.46	0.722	0.157	225.6	24.49	95.92	95.22	98.14	**56.00**	**69.09**
VISION-XL	29.89	0.800	0.115	76.24	22.51	96.10	95.75	99.05	51.96	53.73
LVTINO	30.04	0.824	0.102	59.42	**18.32**	**96.36**	96.09	99.11	52.63	59.05
AVIS	**30.38**	**0.826**	**0.101**	**40.36**	23.94	96.30	**96.21**	**99.24**	52.19	55.13
AVIS *Flash*	29.95	0.818	0.109	62.98	25.46	96.29	96.14	99.15	51.80	54.59
Inpainting										
VISION-XL	29.39	0.801	0.123	144.8	18.92	96.09	94.51	98.95	52.82	60.37
LVTINO	26.36	0.737	0.187	322.0	35.75	95.95	**95.56**	98.60	51.94	**66.86**
AVIS	**31.27**	**0.870**	**0.075**	**68.90**	**10.21**	**96.24**	94.61	**99.20**	**54.06**	64.15
AVIS *Flash*	30.29	0.842	0.091	81.75	13.96	96.18	94.84	99.14	53.22	62.90
Gaussian Deblur										
VISION-XL	**31.10**	**0.835**	**0.093**	41.52	**16.38**	96.27	96.23	99.08	**52.52**	**58.01**
LVTINO	30.64	**0.835**	0.109	47.45	20.37	**96.43**	96.18	99.20	51.65	57.17
AVIS	30.69	0.831	0.110	**36.07**	24.40	96.30	**96.41**	**99.25**	51.54	54.63
AVIS *Flash*	30.41	0.827	0.115	49.66	25.29	96.31	96.33	99.23	51.24	54.39
Temporal Average										
VISION-XL	30.46	0.825	0.095	110.5	12.15	95.87	95.00	98.67	54.27	63.08
LVTINO	31.68	0.878	0.069	77.76	8.015	**96.34**	95.11	99.01	**55.73**	66.36
AVIS	**32.10**	**0.887**	**0.063**	**54.85**	**6.803**	96.26	95.58	**99.12**	55.29	67.07
AVIS *Flash*	31.57	0.878	0.067	57.92	6.829	96.21	**95.65**	99.05	55.16	**67.54**
Spatio-Temporal Average										
VISION-XL	29.62	0.794	0.124	99.54	24.13	95.96	95.41	98.96	51.44	52.73
LVTINO	29.85	**0.818**	**0.110**	95.35	**19.73**	**96.27**	95.98	99.03	**52.42**	**58.02**
AVIS	**29.99**	**0.818**	0.112	**62.11**	25.61	96.22	**96.05**	**99.20**	51.62	54.28
AVIS *Flash*	29.81	0.815	0.117	73.23	25.94	96.22	95.97	99.13	51.32	54.20

Table 2: **Quantitative comparison across five video restoration tasks.** Our proposed AVIS achieves superior performance, and its highly efficient variant, AVIS *Flash*, maintains competitive performance. **Bold** and underline indicate the best and second-best results, respectively.

remaining highly competitive with all baselines on the majority of metrics. In particular, AVIS consistently attains the best or near-best PSNR, SSIM, LPIPS, and FVD across most tasks, validating the effectiveness of combining autoregressive video diffusion priors with our initialization. While LVTINO occasionally achieves favorable perceptual metrics, it relies on both a video diffusion prior [25] and an additional image diffusion prior [56]. In contrast, AVIS achieves better overall results with only a single backbone.

AVIS *Flash* targets a faster operating point. By bypassing iterative measurement updates for subsequent chunks, AVIS *Flash* experiences a slight drop in certain metrics compared to AVIS. However, it remains competitive with baselines such as VISION-XL and LVTINO, while reducing initial latency by over 100 seconds and boosting throughput by over $12\times$ and $8\times$, respectively (Table 1). These results highlight the central benefit of our framework: AVIS establishes a high-quality autoregressive video inverse problem solver, while AVIS *Flash* offers a favorable efficiency–performance trade-off for practical deployment.

Robustness to Measurement Noise. Beyond the noise-free setting (σ_n=0, λ=0) in Table 2, we further evaluate AVIS and AVIS *Flash* under additive Gaussian measurement noise (σ_n=0.01) on all five tasks in Table 3. We fix λ=0.1 in Eq. (12) across all tasks without per-task tuning. AVIS achieves the best PSNR on every task and the best results on the majority of the remaining metrics, while AVIS *Flash* remains competitive with LVTINO. These results demonstrate that our framework remains robust to measurement noise.

Further Experiments. In Appendix C, we provide further results, including long video restoration (§C.1), the robustness of AVIS *Flash* on long videos (§C.2), novel view synthesis formulated as another type of inpainting problem (§C.3), spatio-temporal super-resolution on GoPro240 and Adobe240 (§C.4), chunk-wise measurement consistency (§C.5), a matched-backbone comparison with LVTINO (§C.6), and the detailed runtime for each degradation (§C.7).

5.3 Ablation Study

To isolate the contributions of our core components, we report the ablation study results for video inpainting on the initialization for AVIS in Table 4, and on AVIS *Flash* in Table 5. For ablation results on the other tasks, please refer to Appendix C.8.

Method	PSNR↑	SSIM↑	LPIPS↓	FID↓	FVD↓
Super Resolution					
LVTINO	<u>29.77</u>	<u>0.813</u>	<u>0.125</u>	**23.63**	<u>67.00</u>
AVIS	**30.02**	**0.817**	**0.123**	<u>28.20</u>	**57.70**
AVIS *Flash*	29.57	0.808	0.133	29.13	83.81
Inpainting					
LVTINO	26.39	0.741	0.188	34.36	382.3
AVIS	**31.09**	**0.869**	**0.076**	**9.850**	**65.70**
AVIS *Flash*	<u>29.31</u>	<u>0.819</u>	<u>0.099</u>	<u>12.03</u>	<u>97.16</u>
Gaussian Deblur					
LVTINO	<u>26.19</u>	**0.694**	**0.267**	<u>55.89</u>	520.7
AVIS	**26.80**	<u>0.602</u>	<u>0.287</u>	**55.75**	**215.3**
AVIS *Flash*	25.46	0.563	0.317	61.76	<u>321.7</u>
Temporal Average					
LVTINO	29.04	0.788	0.144	**18.42**	**121.6**
AVIS	**29.90**	**0.843**	**0.124**	<u>22.48</u>	<u>182.1</u>
AVIS *Flash*	<u>29.15</u>	<u>0.826</u>	<u>0.136</u>	23.96	194.5
Spatio-Temporal Average					
LVTINO	27.10	0.701	0.255	51.76	745.7
AVIS	**28.84**	**0.789**	**0.163**	**36.40**	**211.3**
AVIS *Flash*	<u>28.65</u>	<u>0.786</u>	<u>0.167</u>	<u>37.17</u>	<u>250.9</u>

Table 3: **Robustness to measurement noise** (σ_n=0.01, λ=0.1). AVIS and AVIS *Flash* remain stable under measurement noise. **Bold** and <u>underline</u> indicate the best and second-best results, respectively.

Setting	PSNR↑	SSIM↑	LPIPS↓	FID↓	FVD↓
$\mathbf{x}_{\text{init}}^{\lambda}$	<u>30.32</u>	<u>0.826</u>	0.145	26.57	136.8
$\mathcal{D}(\mathcal{E}(\mathbf{x}_{\text{init}}^{\lambda}))$	30.11	0.818	<u>0.136</u>	<u>24.31</u>	130.3
AVIS w/o DDS	29.04	0.789	0.150	25.42	<u>129.5</u>
AVIS	**30.35**	**0.834**	**0.121**	**18.52**	**100.8**

Table 4: **Effect of the initialization on video inpainting** (σ_n=0.05, λ=0.3). AVIS consistently improves over the initialization and its VAE reconstruction.

Initialization. To separate the contribution of the solver from the quality of the initialization, we compare AVIS with the initial estimate $\mathbf{x}_{\text{init}}^{\lambda}$, its VAE reconstruction $\mathcal{D}(\mathcal{E}(\mathbf{x}_{\text{init}}^{\lambda}))$, and AVIS without DDS guidance. We conduct this experiment under the strong noise setting (σ_n=0.05, λ=0.3). As shown in Table 4, AVIS achieves the best results on all metrics. The VAE reconstruction yields only marginal changes over $\mathbf{x}_{\text{init}}^{\lambda}$, indicating that the gain does not come from the encoder $\mathcal{E}$ and decoder $\mathcal{D}$ alone. Without DDS guidance, the fidelity metrics fall even below those of $\mathbf{x}_{\text{init}}^{\lambda}$, as the generative prior alone drifts away from the measurements. These results demonstrate that the final restoration cannot be explained by the initialization alone, and that the generative prior and measurement guidance synergistically refine the initial estimate.

Autoregressive Propagation. To isolate the contribution of autoregressive context propagation, we remove the KV cache for subsequent chunks while keeping the rest of AVIS *Flash* unchanged. Under this setting, each subsequent chunk starts from the same initialized noisy state as in AVIS *Flash*, but is restored without access to the previously generated prefix. As shown in Table 5(a), removing the KV cache consistently degrades all metrics. This highlights the benefit of autoregressive propagation, which provides performance gains complementary to the initialization. In other words, the autoregressive propagation of the restored prefix further improves subsequent chunk restoration and maintains temporal coherence across the video.

To further isolate the role of autoregressive propagation, we visualize the case where subsequent chunks are generated with KV cache conditioning but without the initialization (*i.e.*, starting from pure noise). As shown in the middle row of Figure 3, autoregressive propagation alone effectively preserves the context from previous chunks, but gradually drifts from the desired restoration. This drift is mitigated by our initialization, as shown in the bottom row of Figure 3. Starting the reverse process from the initialized anchor keeps each chunk closer to the desired restoration trajectory, while periodic re-injection of measurement consistency further prevents long-range drift for longer videos, as described in Section 4.3.

Start Time (t_0). We study the effect of the start time $t_0 \in \{0.1, 0.2, 0.5\}$. As shown in Table 5(b), a smaller t_0 consistently improves the majority of metrics, with $t_0 = 0.1$ yielding the best overall performance. This is consistent with the results in Figure 3: as t_0 increases, the starting point

Setting	PSNR↑	SSIM↑	LPIPS↓	FVD↓	FID↓	Sub. Con.↑	Bg. Con.↑	M. Smooth.↑	Aesth.↑	Imag.↑
(a) ***Autoregressive Propagation***										
w/ AR Prop (default)	**30.29**	**0.842**	**0.091**	**81.75**	**13.96**	**96.18**	**94.84**	**99.14**	**53.22**	**62.90**
w/o AR Prop	29.75	0.830	0.103	108.3	17.67	95.99	94.82	99.11	52.97	62.36
(b) ***Start Time*** (t_0)										
$t_0 = 0.1$ (default)	**30.29**	**0.842**	**0.091**	**81.75**	**13.96**	**96.18**	**94.84**	99.14	53.22	62.90
$t_0 = 0.2$	28.98	0.810	0.112	112.0	16.48	96.13	94.53	99.14	53.28	63.45
$t_0 = 0.5$	25.50	0.716	0.192	283.3	30.50	95.92	94.11	**99.15**	**53.31**	**65.21**
(c) ***Sampling Steps*** (K)										
$K = 1$	**30.40**	**0.843**	**0.091**	**81.73**	14.64	96.18	94.73	99.13	53.15	62.47
$K = 2$ (default)	30.29	0.842	**0.091**	81.75	13.96	96.18	94.84	99.14	53.22	62.90
$K = 4$	29.95	0.837	0.094	81.78	**13.69**	**96.23**	**95.17**	**99.15**	**53.34**	**63.46**

Table 5: **Ablation studies on AVIS *Flash* for the video inpainting task.** (a) Effect of autoregressive propagation with KV cache. (b) Impact of the start time t_0. (c) Impact of the number of diffusion sampling steps K. **Bold** and underline indicate the best and second-best results, respectively.

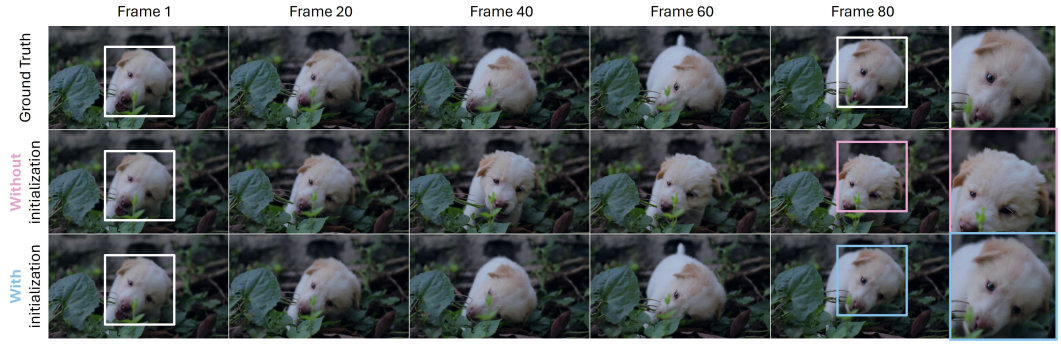

Figure 3: **Effect of AR propagation and initialization.** (Top) Ground-truth video. (Middle) While AR propagation preserves the preceding context, it exhibits gradual error accumulation over time. (Bottom) Our initialization for the reverse diffusion effectively mitigates this temporal drift.

approaches pure noise, and the reverse process increasingly relies on pure autoregressive propagation, which gradually drifts away from the desired restoration. Although a larger t_0 slightly improves some perceptual VBench metrics, it substantially degrades fidelity metrics. We therefore adopt $t_0 = 0.1$ as our default setting.

Sampling Steps (K). We also evaluate the impact of the number of sampling steps ($K \in \{1, 2, 4\}$). Table 5(c) reveals a clear trade-off: as the number of steps increases, certain perceptual metrics (*e.g.*, FID) improve, whereas pixel-level fidelity metrics (PSNR, SSIM) slightly decrease. Since larger K increases inference time with diminishing returns in overall quality, we adopt $K = 2$ as the default setting to achieve the best balance between speed and quality.

6 Conclusion

In this paper, we proposed Autoregressive Video Inverse problem Solver (AVIS) and its highly efficient variant, AVIS *Flash*, for practical deployment of diffusion model-based video inverse problem solvers. Extensive experiments show that the proposed methods improve efficiency over existing non-autoregressive solvers by large margins, with AVIS achieving superior restoration performance and AVIS *Flash* providing a dramatic boost in throughput while maintaining competitive restoration quality. We believe our framework establishes a strong foundation for future research, bridging the gap between generative priors and real-time deployment of video restoration.

Limitations and Broader Impacts. While AVIS paves the way for the practical, real-time deployment of diffusion-based video inverse problem solvers, it still requires multiple VAE passes for the initial pixel-space guidance. Incorporating measurement consistency updates directly in the latent space with auxiliary latent operators, similar to [22, 23], represents an interesting direction toward further acceleration. Furthermore, the capability to restore high-fidelity videos from severely degraded inputs raises potential ethical concerns regarding misinformation and the unauthorized reconstruction of sensitive visual data.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grants funded by the Korea government (MSIT) (RS-2024-00336454, RS-2026-25468886), by the Advanced GPU Utilization Support Program funded by the Government of the Republic of Korea (Ministry of Science and ICT) (02-26-01-0404), by the AI Computing Infrastructure Enhancement (GPU Rental Support) User Support Program funded by the Ministry of Science and ICT (MSIT), Republic of Korea (RQT-25-120217), by the IITP(Institute of Information & Communications Technology Planning & Evaluation)-ITRC(Information Technology Research Center) grant funded by the Korea government(Ministry of Science and ICT)(IITP-2026-RS-2020-II201461), and by a Korea University Grant (K2626081).

References

[1] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[2] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2020.

[3] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[4] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[5] Jonathan Ho, Tim Salimans, Alexey A Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. In *ICLR Workshop on Deep Generative Models for Highly Structured Data*, 2022.

[6] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=di52zR8xgf`.

[7] Xingchao Liu, Chengyue Gong, and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://arxiv.org/abs/2209.03003`.

[8] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://arxiv.org/abs/2210.02747`.

[9] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

[10] Pascal Chang, Jingwei Tang, Markus Gross, and Vinicius C. Azevedo. How i warped your noise: a temporally-correlated noise prior for diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=pzElnMrgSD`.

[11] Giannis Daras, Weili Nie, Karsten Kreis, Alexandros G Dimakis, Morteza Mardani, Nikola B Kovachki, and Arash Vahdat. Warped diffusion: Solving video inverse problems with image diffusion models. *Advances in Neural Information Processing Systems*, 37:101116–101143, 2024.

[12] Taesung Kwon and Jong Chul Ye. Solving video inverse problems using image diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=TRWxFUzK9K`.

[13] Taesung Kwon and Jong Chul Ye. Vision-xl: High definition video inverse problem solver using latent image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10465–10474, 2025.

[14] Taesung Kwon, Gookho Song, Yoosun Kim, Jeongsol Kim, Jong Chul Ye, and Mooseok Jang. Video diffusion posterior sampling for seeing beyond dynamic scattering layers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[15] Alessio Spagnoletti, Andres Almansa, and Marcelo Pereyra. LVTINO: LAtent video consistency INverse solver for high definition video restoration. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=8SyEcWVe10.

[16] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

[17] Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, et al. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024.

[18] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

[19] Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

[20] Arslan Ali, Junjie Bai, Maciej Bala, Yogesh Balaji, Aaron Blakeman, Tiffany Cai, Jiaxin Cao, Tianshi Cao, Elizabeth Cha, Yu-Wei Chao, et al. World simulation with video foundation models for physical ai. *arXiv preprint arXiv:2511.00062*, 2025.

[21] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[22] Ron Raphaeli, Sean Man, and Michael Elad. Silo: Solving inverse problems with latent operators. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10570–10580, 2025.

[23] Yeobin Hong, Suhyeon Lee, Hyungjin Chung, and Jong Chul Ye. Inversecrafter: Efficient video recapture as a latent domain inverse problem. *arXiv preprint arXiv:2512.05672*, 2025.

[24] Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024.

[25] Tianwei Yin, Qiang Zhang, Richard Zhang, William T Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22963–22974, 2025.

[26] Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[27] Hyungjin Chung, Byeongsu Sim, and Jong Chul Ye. Come-closer-diffuse-faster: Accelerating conditional diffusion models for inverse problems through stochastic contraction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12413–12422, 2022.

[28] Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=OnD9zGAGT0k.

[29] Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=9_gsMA8MRKQ.

[30] Yinhuai Wang, Jiwen Yu, and Jian Zhang. Zero-shot image restoration using denoising diffusion null-space model. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=mRieQgMtNTQ.

[31] Hyungjin Chung, Suhyeon Lee, and Jong Chul Ye. Decomposed diffusion sampler for accelerating large-scale inverse problems. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=DsEhqQtfAG.

[32] Morteza Mardani, Jiaming Song, Jan Kautz, and Arash Vahdat. A variational perspective on solving inverse problems with diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=1YO4EE3SPB.

[33] Bowen Song, Soo Min Kwon, Zecheng Zhang, Xinyu Hu, Qing Qu, and Liyue Shen. Solving inverse problems with latent diffusion models via hard data consistency. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=j8hdRqOUhN.

[34] Litu Rout, Negin Raoof, Giannis Daras, Constantine Caramanis, Alex Dimakis, and Sanjay Shakkottai. Solving linear inverse problems provably via posterior sampling with latent diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

[35] Bingliang Zhang, Wenda Chu, Julius Berner, Chenlin Meng, Anima Anandkumar, and Yang Song. Improving diffusion inverse problem solving with decoupled noise annealing. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 20895–20905, 2025.

[36] Jeongsol Kim, Bryan Sangwoo Kim, and Jong Chul Ye. Flowdps: Flow-driven posterior sampling for inverse problems. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12328–12337, 2025.

[37] Jonghyun Park and Jong Chul Ye. Flowlps: Langevin-proximal sampling for flow-based inverse problem solvers. *arXiv preprint arXiv:2512.07150*, 2025.

[38] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020.

[39] OpenAI. Sora. https://openai.com/sora/, 2024.

[40] Google DeepMind. Veo 3. https://deepmind.google/models/veo/, 2025.

[41] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[43] David Ruhe, Jonathan Heek, Tim Salimans, and Emiel Hoogeboom. Rolling diffusion models. In *International Conference on Machine Learning*, pages 42818–42835. PMLR, 2024.

[44] Jihwan Kim, Junoh Kang, Jinyoung Choi, and Bohyung Han. Fifo-diffusion: Generating infinite videos from text without training. *Advances in Neural Information Processing Systems*, 37:89834–89868, 2024.

[45] Lvmin Zhang, Shengqu Cai, Muyang Li, Gordon Wetzstein, and Maneesh Agrawala. Frame context packing and drift prevention in next-frame-prediction video diffusion models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[46] Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong MU, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=66NzcRQuOq.

[47] Magnus R Hestenes, Eduard Stiefel, et al. Methods of conjugate gradients for solving linear systems. *Journal of research of the National Bureau of Standards*, 49(6):409–436, 1952.

[48] Jörg Liesen and Zdenek Strakos. *Krylov subspace methods: principles and analysis*. Numerical Mathematics and Scie, 2013.

[49] Pexels. Pexels. https://www.pexels.com/, 2024.

[50] Chang-Han Yeh, Chin-Yang Lin, Zhixiang Wang, Chi-Wei Hsiao, Ting-Hsuan Chen, and Yu-Lun Liu. Diffir2vr-zero: Zero-shot video restoration with diffusion-based image restoration models. *arXiv preprint arXiv:2407.01519*, 2024.

[51] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[52] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[53] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[54] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. FVD: A new metric for video generation, 2019. URL `https://openreview.net/forum?id=rylgEULtdN`.

[55] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21807–21818, 2024.

[56] Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T Freeman. Improved distribution matching distillation for fast image synthesis. *Advances in neural information processing systems*, 37:47455–47487, 2024.

[57] Saurabh Saxena, Charles Herrmann, Junhwa Hur, Abhishek Kar, Mohammad Norouzi, Deqing Sun, and David J Fleet. The surprising effectiveness of diffusion models for optical flow and monocular depth estimation. *Advances in Neural Information Processing Systems*, 36:39443–39469, 2023.

[58] Xinqi Lin, Jingwen He, Ziyan Chen, Zhaoyang Lyu, Bo Dai, Fanghua Yu, Yu Qiao, Wanli Ouyang, and Chao Dong. Diffbir: Toward blind image restoration with generative diffusion prior. In *European conference on computer vision*, pages 430–448. Springer, 2024.

[59] Weimin Bai, Suzhe Xu, Yiwei Ren, Jinhua Hao, Ming Sun, Wenzheng Chen, and He Sun. Instantvir: Real-time video inverse problem solver with distilled diffusion prior. *arXiv preprint arXiv:2511.14208*, 2025.

[60] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[61] Songwei Ge, Aniruddha Mahapatra, Gaurav Parmar, Jun-Yan Zhu, and Jia-Bin Huang. On the content bias in fréchet video distance. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7277–7288, 2024.

[62] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[63] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[64] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[65] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[66] Zhen Li, Zuo-Liang Zhu, Ling-Hao Han, Qibin Hou, Chun-Le Guo, and Ming-Ming Cheng. Amt: All-pairs multi-field transforms for efficient frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9801–9810, 2023.

[67] LAION-AI. aesthetic-predictor. `https://github.com/LAION-AI/aesthetic-predictor`, 2022.

[68] Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. Musiq: Multi-scale image quality transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5148–5157, 2021.

[69] Hyeonho Jeong, Suhyeon Lee, and Jong Chul Ye. Reangle-a-video: 4d video generation as video-to-video translation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11164–11175, 2025.

[70] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in Neural Information Processing Systems*, 37:21875–21911, 2024.

[71] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3883–3891, 2017.

[72] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. In *2017 IEEE Conference on computer vision and pattern recognition (CVPR)*, pages 237–246. IEEE, 2017.

A Discussion of the Error Bound and Proof

Let $t_0 > t_1 > \cdots > t_K = 0$ be the reverse sampling schedule of K steps for the n-th chunk, where $\mathbf{z}^{n,\text{tgt}}$ denotes the target chunk and $\mathbf{z}^n_{\text{init}} = \mathcal{E}(\mathbf{x}^n_{\text{init}})$ denotes the initial chunk encoded from the measurement-consistent estimate. We define the initial error as:

$$\epsilon^n_0 := \|\mathbf{z}^n_{\text{init}} - \mathbf{z}^{n,\text{tgt}}\|. \tag{15}$$

Consider the reverse sampling trajectories of both the current chunk $\mathbf{z}^n_t$ and the target chunk $\mathbf{z}^{n,\text{tgt}}_t$, coupled via the shared Gaussian noise $\boldsymbol{\xi}_0$. At the start time t_0, the noisy states are given by:

$$\mathbf{z}^n_{t_0} = (1-t_0)\mathbf{z}^n_{\text{init}} + t_0\boldsymbol{\xi}_0, \quad \mathbf{z}^{n,\text{tgt}}_{t_0} = (1-t_0)\mathbf{z}^{n,\text{tgt}} + t_0\boldsymbol{\xi}_0, \quad \boldsymbol{\xi}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{16}$$

Therefore, the initial error at the start time t_0 is explicitly scaled down by the noise level:

$$\epsilon^n_{t_0} := \|\mathbf{z}^n_{t_0} - \mathbf{z}^{n,\text{tgt}}_{t_0}\| = (1-t_0)\epsilon^n_0. \tag{17}$$

For each reverse step $t_k \to t_{k+1}$ coupled via the shared Gaussian noise $\boldsymbol{\xi}_{k+1}$, the guidance-free updates are given by:

$$\hat{\mathbf{z}}^n_{0|t_k} = \mathbf{z}^n_{t_k} - t_k\mathbf{v}_\theta(\mathbf{z}^n_{t_k}, t_k; \text{KV}^{<n}), \tag{18}$$

$$\hat{\mathbf{z}}^{n,\text{tgt}}_{0|t_k} = \mathbf{z}^{n,\text{tgt}}_{t_k} - t_k\mathbf{v}_\theta(\mathbf{z}^{n,\text{tgt}}_{t_k}, t_k; \text{KV}^{<n,\text{tgt}}), \tag{19}$$

$$\mathbf{z}^n_{t_{k+1}} = (1-t_{k+1})\hat{\mathbf{z}}^n_{0|t_k} + t_{k+1}\boldsymbol{\xi}_{k+1}, \tag{20}$$

$$\mathbf{z}^{n,\text{tgt}}_{t_{k+1}} = (1-t_{k+1})\hat{\mathbf{z}}^{n,\text{tgt}}_{0|t_k} + t_{k+1}\boldsymbol{\xi}_{k+1}, \quad \boldsymbol{\xi}_{k+1} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{21}$$

To capture the sensitivity of the vector field to previously restored chunks, we introduce the context error δ_n, defined as the average mismatch over the preceding chunks ($m < n$):

$$\delta_n := \frac{1}{n-1}\sum_{m=1}^{n-1}\|\mathbf{z}^m_0 - \mathbf{z}^{m,\text{tgt}}_0\|. \tag{22}$$

This quantity represents the accumulated prefix mismatch presented to the autoregressive model.

We assume that, at each timestep t_k, the vector field $\mathbf{v}_\theta$ is Lipschitz continuous with respect to both the current chunk and the context condition:

$$\|\mathbf{v}_\theta(\mathbf{z}, t_k; \mathbf{c}) - \mathbf{v}_\theta(\mathbf{z}', t_k; \mathbf{c})\| \le L_{z,k}\|\mathbf{z} - \mathbf{z}'\|, \tag{23}$$

$$\|\mathbf{v}_\theta(\mathbf{z}, t_k; \mathbf{c}) - \mathbf{v}_\theta(\mathbf{z}, t_k; \mathbf{c}')\| \le L_{c,k}\|\mathbf{c} - \mathbf{c}'\|_{\text{ctx}}. \tag{24}$$

We are now ready to prove Proposition 1.

Proposition 1. *Let ϵ^n_0 denote the initial error of the current chunk, and let δ_n denote the context error propagated from previously restored chunks. Under local Lipschitz assumptions on the autoregressive vector field $\mathbf{v}_\theta$, the final error ϵ^n_K of the n-th chunk after K sampling steps satisfies*

$$\epsilon^n_K \le \Lambda_K\epsilon^n_0 + B_K\delta_n, \tag{14}$$

where Λ_K and B_K depend on the sampling schedule and local Lipschitz constants.

Proof. Since the trajectories for the current chunk $\mathbf{z}^n_{t_{k+1}}$ and the target chunk $\mathbf{z}^{n,\text{tgt}}_{t_{k+1}}$ are coupled via the shared Gaussian noise at step $k+1$, the explicit noise terms cancel in the difference:

$$\begin{aligned}&\|\mathbf{z}^n_{t_{k+1}} - \mathbf{z}^{n,\text{tgt}}_{t_{k+1}}\| \\ &= (1-t_{k+1})\Big\|\mathbf{z}^n_{t_k} - \mathbf{z}^{n,\text{tgt}}_{t_k} - t_k\Big(\mathbf{v}_\theta(\mathbf{z}^n_{t_k}, t_k; \text{KV}^{<n}) - \mathbf{v}_\theta(\mathbf{z}^{n,\text{tgt}}_{t_k}, t_k; \text{KV}^{<n,\text{tgt}})\Big)\Big\|.\end{aligned}$$

By the triangle inequality and the Lipschitz assumptions,

$$\begin{aligned}\|\mathbf{z}^n_{t_{k+1}} - \mathbf{z}^{n,\text{tgt}}_{t_{k+1}}\| &\le (1-t_{k+1})\Big(\|\mathbf{z}^n_{t_k} - \mathbf{z}^{n,\text{tgt}}_{t_k}\| + t_kL_{z,k}\|\mathbf{z}^n_{t_k} - \mathbf{z}^{n,\text{tgt}}_{t_k}\| + t_kL_{c,k}\delta_n\Big) \\ &= (1-t_{k+1})(1+t_kL_{z,k})\|\mathbf{z}^n_{t_k} - \mathbf{z}^{n,\text{tgt}}_{t_k}\| + (1-t_{k+1})t_kL_{c,k}\delta_n.\end{aligned}$$

Let $\lambda_k := (1 - t_{k+1})(1 + t_k L_{z,k})$ and $\beta_k := (1 - t_{k+1}) t_k L_{c,k}$. This yields the recursive relation:

$$\epsilon^n_{t_{k+1}} \leq \lambda_k \epsilon^n_{t_k} + \beta_k \delta_n. \tag{25}$$

Applying this relation recursively over $k = 0, \cdots, K-1$ yields the final error after K sampling steps:

$$\epsilon^n_K = \epsilon^n_{t_K} \leq \left(\prod_{r=0}^{K-1} \lambda_r \right) \epsilon^n_{t_0} + \sum_{r=0}^{K-1} \left(\prod_{\ell=r+1}^{K-1} \lambda_\ell \right) \beta_r \, \delta_n \tag{26}$$

$$\leq \underbrace{\left((1 - t_0) \prod_{r=0}^{K-1} \lambda_r \right)}_{:=\Lambda_K} \epsilon^n_0 + \underbrace{\sum_{r=0}^{K-1} \left(\prod_{\ell=r+1}^{K-1} \lambda_\ell \right) \beta_r \, \delta_n}_{:=B_K}. \tag{27}$$

This concludes the proof of the Proposition. □

Discussion of the Start Time t_0. For a fixed number of sampling steps K, the start time t_0 affects the two terms in the bound differently. Under the linear schedule $t_k = (1 - k/K) t_0$, the context-dependent coefficient B_K decreases as t_0 becomes smaller, indicating that a smaller t_0 suppresses the propagation of prefix mismatch across chunks. Meanwhile, the coefficient Λ_K does not possess an interior minimizer with respect to t_0. Its lowest values are approached only near the boundaries ($t_0 \to 0$ or $t_0 \to 1$), depending on the local Lipschitz constants of the vector field. Consequently, the bound is better viewed as an interpretive insight rather than a prescription for the optimal choice of t_0. In particular, as $t_0 \to 1$, the starting latent becomes nearly pure noise, so the reverse process becomes progressively less dependent on the pre-restored initial estimate and more dominated by the learned video prior and autoregressive context. This helps explain why a larger t_0 empirically leads to more generation-like behavior, whereas a smaller t_0 better preserves restoration fidelity in our evaluations.

B Experimental Details

B.1 Implementation Details

Our implementation is built upon the Self Forcing codebase [26] with Wan2.1-T2V-1.3B [18]. To ensure straightforward reproducibility, we retain the original configurations, except for specific parameter settings adapted for video restoration tasks (initial timestep t_0 and sampling steps K). For the video encoding, we use Wan-VAE [18] with spatial and temporal downsampling factors of 8 and 4, respectively. Consequently, an input RGB video with $1 + T$ frames at a resolution of $H \times W$ is encoded into a latent tensor of $1 + T/4$ latent frames with spatial dimensions of $H/8 \times W/8$. Following recent advancements in autoregressive (AR) modeling [25, 26], we generate a chunk of $L = 3$ latent frames at a time, consistent with the original Self Forcing setup. Although the original framework employs a 4-step diffusion, we found that adapting it to a 2-step process ($K = 2$) with a start time of $t_0 = 0.1$ provides a favorable balance between computational efficiency and visual quality, specifically for video restoration.

Initial Estimation. We detail the procedure for obtaining the initial estimate through multi-step optimization. To derive the initial estimate for each degradation, we employ the conjugate gradient (CG) method to optimize Eq. (12). For the main experiments without measurement noise, we set $\lambda = 0$. By default, the optimization is initialized directly with the measurement $\mathbf{y}$. However, to address dimensional mismatches in specific tasks, we apply tailored preprocessing to $\mathbf{y}$. For super-resolution and spatio-temporal averaging, we align the spatial dimensions using bilinear interpolation. For inpainting, inspired by DDVM [57], we apply nearest-neighbor infilling to the measurement to establish a robust starting point. Following this initialization, we perform a task-specific number of CG updates: 5 steps for Gaussian deblurring and super-resolution, 50 steps for temporal averaging, and 100 steps for spatio-temporal averaging. Notably, for inpainting, since the initialized measurement already satisfies the measurement consistency term, no further CG updates are required.

Inference and Optimization Setup. Once the initial estimate is obtained, it serves as the starting point for the reverse diffusion process. For the sampling configuration, we employ a start time of $t_0 = 0.1$ and $K = 2$ sampling steps for the main experiments. The justification for these optimal values is provided in our ablation studies (Section 5.3). For the DDS update, we avoid complex parameter tuning by fixing $\gamma = 1$ for the proximal optimization in Eq. (9). Additionally, we consistently conduct 5 CG updates across all degradation types.

Computing Resources. Leveraging the efficiency of our framework, we process videos at a 480×832 resolution with a 4-second initial latency and a 5.9 FPS throughput, requiring only 18.4 GB of VRAM on a single consumer-grade GPU (NVIDIA RTX 4090). Furthermore, we confirm that deploying our framework on a single NVIDIA H100 GPU accelerates processing even more, achieving an initial latency of 1.85 seconds and a throughput of 10.18 FPS with a VRAM consumption of 26.79 GB.

B.2 Baseline Models

The following summarizes the key ideas of the diffusion model-based video inverse problem solvers (DVIS) used for the evaluation.

- **DiffIR2VR-Zero** [50] adapts pre-trained image restoration diffusion models [58] for video restoration without additional training, maintaining temporal consistency through hierarchical latent warping and token merging techniques.
- **VISION-XL** [13] is a zero-shot inverse problem solver for high-definition video restoration using latent image diffusion models [6], efficiently achieving temporal consistency on a single GPU through pseudo-batch sampling and inversion strategies.
- **LVTINO** [15] is a zero-shot inverse problem solver for high-definition video restoration that integrates Video Consistency Models (VCMs) [25] and Image Consistency Models (ICMs) [56], thereby reducing sampling steps and achieving superior restoration quality.

For all baselines, we used their official implementations in the same environment as AVIS. We configured them to process the same resolution and number of frames, while applying the same

degradation operators. Additionally, since LVTINO exceeded the 24GB VRAM limit, we precomputed the text embeddings and removed the text encoders from both the image and video diffusion models during inference to fit within memory constraints.

Other Baselines. Although there exist other potential candidates for DVIS, direct comparisons with methods such as SVI [12] and VDPS [14] were infeasible due to limitations in their supported resolutions and high memory consumption caused by heavy operations (*e.g.*, batch processing and Jacobian computations). Furthermore, several recent methods [10, 11, 59] were excluded from our evaluation due to the lack of publicly available implementations. Nevertheless, we believe our experimental setup provides a sufficiently representative baseline for evaluating zero-shot video restoration. As VISION-XL [13] and LVTINO [15] are currently recognized as the leading zero-shot solvers for high-definition video restoration, our experiments successfully demonstrate the effectiveness of our proposed method within the scope of zero-shot video inverse problem methodologies.

B.3 Evaluation Metrics

To ensure straightforward reproducibility, our evaluations rely on official implementations and widely adopted PyTorch [60] packages. For frame-wise evaluation, we extract individual frames from each video sequence to compute the respective metrics. Specifically, we utilize `pytorch-msssim` for SSIM [51], `lpips` for LPIPS [52], and `pytorch-fid` for FID [53]. For video quality assessment, we employ FVD [54] and VBench [55], adopting a publicly available repository [61] for FVD computation and the official repository for VBench. All primary evaluations are conducted on a single NVIDIA RTX 4090 GPU. To further demonstrate the computational efficiency of our method, we additionally report inference times measured on a single NVIDIA H100 GPU.

For completeness, we provide a concise description of the evaluation metrics used in our experiments:

- **PSNR** measures the pixel-level accuracy between the restored and ground-truth frames.
- **SSIM** evaluates the structural similarity between the restored and ground-truth frames.
- **LPIPS** calculates the perceptual similarity based on feature embeddings extracted from a pre-trained backbone [52].
- **FID** measures the distance between the feature distributions of generated and real images using the Inception-v3 network [62]. It is computed under the assumption that both feature distributions follow multivariate Gaussian distributions.
- **FVD** extends the principles of FID to measure the quality and diversity of generated videos, utilizing features extracted from an Inflated 3D ConvNet (I3D) [63].
- **Subject Consistency (VBench)** evaluates whether the appearance of a subject remains consistent by calculating the DINO [64] feature similarity across frames.
- **Background Consistency (VBench)** assesses the temporal stability of the background by computing the CLIP [65] feature similarity across frames.
- **Motion Smoothness (VBench)** measures the temporal continuity of the generated motions using motion priors derived from a video frame interpolation model [66].
- **Aesthetic Quality (VBench)** evaluates the overall artistic appeal and visual naturalness of the video frames using the LAION aesthetic predictor [67].
- **Imaging Quality (VBench)** evaluates the visual distortion (*e.g.*, over-exposure, noise, blur) appearing in the video frames using the MUSIQ [68] image quality predictor.

C Further Experimental Results

C.1 Long Video Restoration

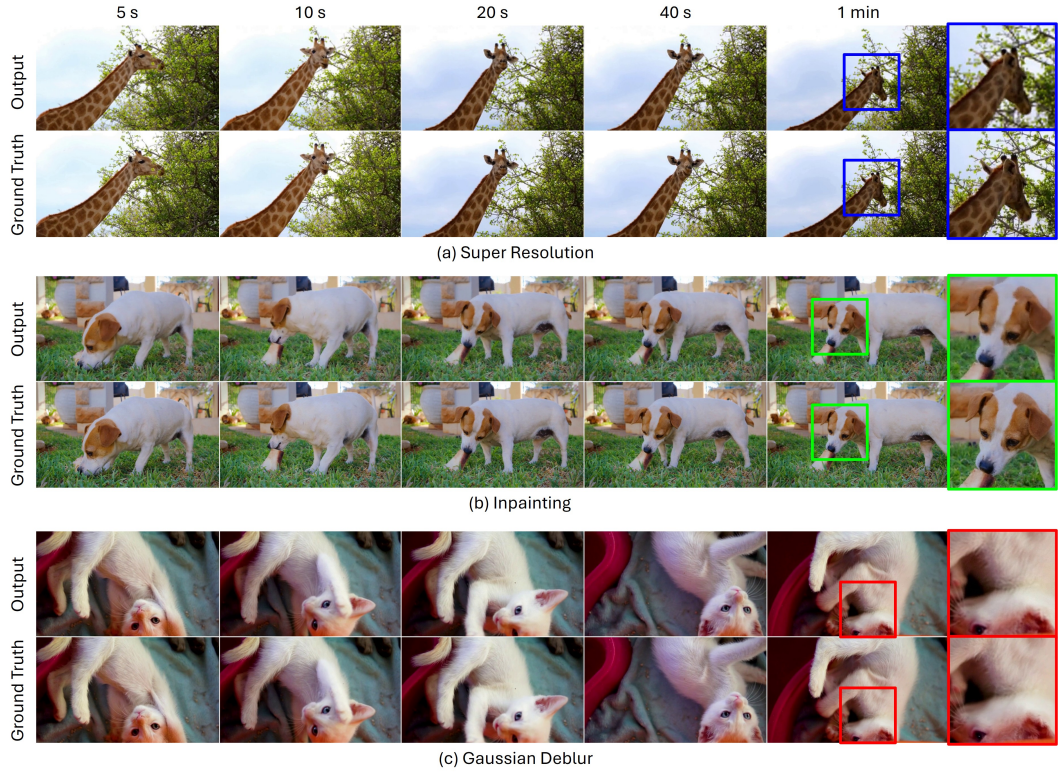

Figure 4: **Qualitative results of long video restoration.** By periodically re-injecting measurement consistency (every 7 chunks), AVIS *Flash* effectively prevents temporal drift and remains consistent with the ground truth, even in the final frames. The timestamps at the top indicate the elapsed time within the long video.

To further support the claims made in Section 4.3, we conduct additional experiments evaluating the scalability of our approach on longer video sequences. Specifically, we aim to restore a one-minute video consisting of 960 frames at 16 FPS. As described in Section 4.3, AVIS *Flash* can be further stabilized by periodically re-injecting measurement consistency. In this long video experiment, we apply guidance every 7 chunks to prevent error accumulation over time.

As demonstrated in Figure 4, even for a one-minute sequence, AVIS *Flash* successfully retains the ground truth context and prevents noticeable temporal drift in later frames. This periodic resetting effectively mitigates accumulated context mismatch, ensuring robust restoration regardless of the overall sequence length.

C.2 Robustness of AVIS *Flash* on Long Videos

We analyze the robustness of AVIS *Flash* on long videos from the Pexels dataset with respect to video length and guidance interval.

Video length. We evaluate AVIS *Flash* on 30-second (480-frame) and 60-second (960-frame) videos across all degradations (Table 6). Restoration quality is preserved as the sequence length doubles, with no sign of degradation as the autoregressive rollout proceeds.

Guidance interval. We evaluate the effect of the guidance interval N, where guidance is applied every N chunks, on 60-second videos (Table 7). Dense guidance yields better quality, while sparse guidance yields larger speedup. Since most of the speedup is obtained by increasing N from 1 to 7, we adopt $N{=}7$ in AVIS *Flash*.

Duration	PSNR↑	SSIM↑	LPIPS↓	FID↓	FVD↓
Super Resolution					
30 s	**30.49**	0.848	**0.235**	20.19	141.5
60 s	30.45	**0.850**	**0.235**	**16.80**	**124.1**
Inpainting					
30 s	**30.51**	**0.853**	**0.273**	15.06	158.4
60 s	30.44	**0.853**	0.274	**11.88**	**150.1**
Gaussian Deblur					
30 s	**30.82**	0.850	**0.273**	17.62	115.5
60 s	30.79	**0.851**	**0.273**	**14.95**	**82.50**
Temporal Average					
30 s	**32.14**	**0.894**	**0.127**	6.480	103.2
60 s	32.02	0.893	0.128	**4.370**	**96.53**
Spatio-Temporal Average					
30 s	**30.29**	0.843	**0.248**	21.14	133.1
60 s	30.25	**0.845**	**0.248**	**17.51**	**127.7**

Table 6: **AVIS *Flash* performance across video durations.** Restoration quality is preserved even when the video length doubles.

Interval N	PSNR↑	SSIM↑	LPIPS↓	FID↓	FVD↓	Speedup
Super Resolution						
1	**30.92**	**0.856**	**0.223**	**15.98**	**77.60**	1.00×
7	30.45	0.850	0.235	16.80	124.1	4.94×
14	30.40	0.849	0.237	17.14	120.6	5.35×
Inpainting						
1	**31.43**	**0.877**	**0.227**	**9.140**	**140.3**	1.00×
7	30.44	0.853	0.274	11.88	150.1	3.54×
14	30.31	0.849	0.281	12.49	160.9	3.81×
Gaussian Deblur						
1	**31.17**	**0.854**	**0.265**	**14.47**	**73.93**	1.00×
7	30.79	0.851	0.273	14.95	82.50	3.83×
14	30.74	0.850	0.275	15.28	93.31	5.80×
Temporal Average						
1	**32.80**	**0.905**	**0.116**	**4.150**	**77.40**	1.00×
7	32.02	0.893	0.128	4.370	96.53	4.93×
14	31.89	0.890	0.130	4.440	92.36	5.34×
Spatio-Temporal Average						
1	**30.52**	**0.848**	**0.241**	**16.82**	**95.20**	1.00×
7	30.25	0.845	0.248	17.51	127.7	4.89×
14	30.22	0.845	0.249	17.71	117.7	5.28×

Table 7: **Effect of the guidance interval on 60-second videos.** Guidance is applied every N chunks. Denser guidance improves quality, while sparser guidance yields larger speedup.

C.3 Novel View Synthesis

In this section, we explore the feasibility of applying AVIS *Flash* for inference-time novel view synthesis. We adopt the depth-based geometric warping pipeline from [69] for initial novel view rendering. Specifically, we first estimate the depth map of each video frame using the monocular depth estimator [70]. Using the input video frames and their corresponding depth maps, we then lift the pixels to construct dynamic 3D point clouds. These point clouds are then reprojected onto the 2D plane according to a target camera trajectory.

During this reprojection, disocclusions naturally occur, exposing previously occluded regions and resulting in "holes" in the warped frames. We demonstrate that AVIS *Flash* can act as an inpainting operator to fill in these missing regions. As shown in Figure 5, AVIS *Flash* can plausibly synthesize the disoccluded content, demonstrating its potential applicability for inference-time novel view synthesis.

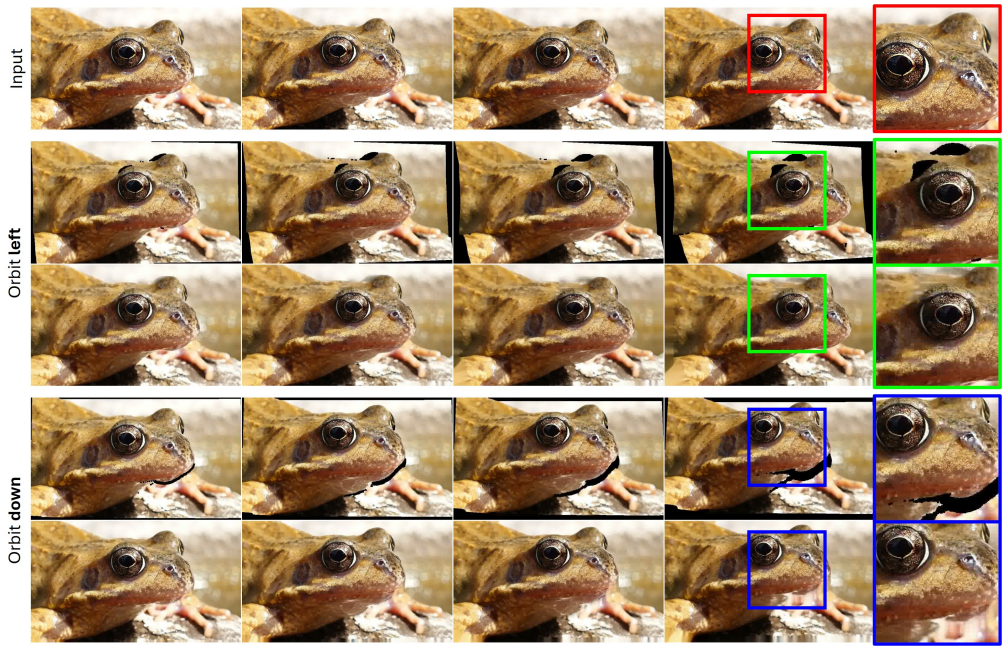

Figure 5: **Qualitative results of novel view video synthesis.** We demonstrate the capability of AVIS *Flash* to generate novel views under target camera trajectories, including orbit left and orbit down. For each camera trajectory, the upper row shows the initial warped frames, and the lower row shows the inpainted outputs of AVIS *Flash*. The boxes highlight corresponding regions in the final frames, illustrating how the method can plausibly synthesize the intended camera movements.

C.4 Spatio-Temporal Super-Resolution

We evaluate AVIS and AVIS *Flash* on noisy spatio-temporal SR with scale factors of 4× and 8× on GoPro240 [71] and Adobe240 [72]. These settings follow the official protocols of LVTINO [15]. As shown in Table 8, AVIS achieves the best overall restoration quality on both datasets and at both scale factors, with especially large reductions in FVD. Notably, AVIS *Flash* remains close to AVIS even at the 8× scale factor while bypassing measurement updates for subsequent chunks. These results indicate that both variants remain effective under a severely ill-conditioned forward operator.

	Scale factor 4×					Scale factor 8×				
Method	PSNR↑	SSIM↑	LPIPS↓	FID↓	FVD↓	PSNR↑	SSIM↑	LPIPS↓	FID↓	FVD↓
GoPro240										
LVTINO	26.45	0.761	0.175	65.51	474.3	22.69	0.620	0.354	**138.6**	923.2
AVIS	**27.29**	**0.814**	**0.135**	**56.94**	**124.2**	**23.21**	**0.646**	**0.317**	141.8	**542.2**
AVIS *Flash*	27.02	0.805	0.145	59.77	141.2	23.09	0.642	0.323	139.4	559.5
Adobe240										
LVTINO	25.38	0.745	0.183	55.84	593.1	21.93	0.625	0.362	125.0	833.3
AVIS	**26.32**	**0.777**	**0.166**	**50.39**	**167.1**	**22.59**	**0.629**	**0.343**	113.4	**570.7**
AVIS *Flash*	26.04	0.767	0.174	51.54	179.9	22.39	0.624	0.348	**111.8**	574.8

Table 8: **Spatio-temporal super-resolution on GoPro240 and Adobe240.** AVIS and AVIS *Flash* remain effective even at a scale factor of 8×. **Bold** and underline indicate the best and second-best results, respectively.

C.5 Chunk-wise Measurement Consistency

We report the RMSE of the measurement consistency term ($\|\mathcal{A}(\mathbf{x}_0) - \mathbf{y}\|$) per chunk for both AVIS and AVIS *Flash* across all degradations in Table 9. For AVIS, the residual stays comparable to or below that of the first chunk throughout the rollout. For AVIS *Flash*, the residual rises above the first chunk but remains nearly flat in all subsequent chunks rather than growing. This result indicates that error accumulation in AVIS *Flash* does not diverge over the autoregressive rollout.

Method	Chunk 1	Chunk 2	Chunk 3	Chunk 4	Chunk 5	Chunk 6	Chunk 7
Super Resolution							
AVIS	0.030	0.030	0.030	0.030	0.030	0.030	0.030
AVIS *Flash*	0.030	0.036	0.036	0.035	0.036	0.036	0.036
Inpainting							
AVIS	0.048	0.049	0.050	0.049	0.050	0.050	0.050
AVIS *Flash*	0.048	0.056	0.057	0.057	0.058	0.058	0.058
Gaussian Deblur							
AVIS	0.017	0.017	0.017	0.016	0.016	0.016	0.016
AVIS *Flash*	0.017	0.021	0.019	0.019	0.019	0.020	0.020
Temporal Average							
AVIS	0.040	0.036	0.036	0.037	0.036	0.037	0.037
AVIS *Flash*	0.040	0.041	0.042	0.043	0.043	0.042	0.043
Spatio-Temporal Average							
AVIS	0.022	0.023	0.023	0.023	0.023	0.023	0.023
AVIS *Flash*	0.022	0.028	0.026	0.026	0.026	0.026	0.026

Table 9: **Chunk-wise measurement residual (RMSE).** The residual of AVIS *Flash* remains nearly flat after the first chunk, indicating no divergent error accumulation.

C.6 Backbone Fairness Check

To examine whether the gains of AVIS are primarily due to the video prior rather than the solving framework, we additionally evaluate the leading non-autoregressive solver, LVTINO [15], coupled with the same autoregressive backbone [26] as AVIS, denoted as LVTINO-AR, across all five tasks.

As shown in Table 10, AVIS and LVTINO-AR are broadly comparable in restoration quality. At comparable quality, however, the two differ substantially in cost, as shown in Table 11. On 4× super-resolution with 81-frame input on a single RTX 4090, LVTINO-AR requires 89.2 seconds per video (0.91 FPS), whereas AVIS requires 68.5 seconds (1.18 FPS) and AVIS *Flash* 13.7 seconds (5.91 FPS). In addition, LVTINO-AR requires an auxiliary image consistency model [56] alongside the video prior, incurring additional memory overhead, whereas AVIS operates with a single backbone.

Method	PSNR↑	SSIM↑	LPIPS↓	FVD↓	FID↓
Super Resolution					
LVTINO	30.04	<u>0.824</u>	<u>0.102</u>	59.42	**18.32**
LVTINO-AR	<u>30.28</u>	0.818	0.113	<u>48.50</u>	<u>21.99</u>
AVIS	**30.38**	**0.826**	**0.101**	**40.36**	23.94
Inpainting					
LVTINO	26.36	0.737	0.187	322.0	35.75
LVTINO-AR	<u>29.79</u>	<u>0.840</u>	<u>0.110</u>	<u>111.0</u>	<u>20.59</u>
AVIS	**31.27**	**0.870**	**0.075**	**68.90**	**10.21**
Gaussian Deblur					
LVTINO	30.64	<u>0.835</u>	<u>0.109</u>	47.45	**20.37**
LVTINO-AR	**31.53**	**0.841**	**0.100**	**27.25**	<u>21.62</u>
AVIS	<u>30.69</u>	0.831	0.110	<u>36.07</u>	24.40
Temporal Average					
LVTINO	31.68	0.878	<u>0.069</u>	77.76	8.015
LVTINO-AR	<u>32.06</u>	**0.893**	0.083	**48.31**	**5.836**
AVIS	**32.10**	<u>0.887</u>	**0.063**	<u>54.85</u>	<u>6.803</u>
Spatio-Temporal Average					
LVTINO	29.85	<u>0.818</u>	<u>0.110</u>	95.35	**19.73**
LVTINO-AR	**30.33**	**0.824**	**0.103**	<u>78.67</u>	<u>19.97</u>
AVIS	<u>29.99</u>	<u>0.818</u>	0.112	**62.11**	25.61

Table 10: **Backbone fairness check across five tasks.** LVTINO-AR denotes LVTINO coupled with the same autoregressive backbone as AVIS. **Bold** and <u>underline</u> indicate the best and second-best results, respectively.

Method	Time (s)↓	FPS↑	Aux. image model
LVTINO	114	0.71	Required
LVTINO-AR	89.2	0.91	Required
AVIS	<u>68.5</u>	<u>1.18</u>	Not required
AVIS *Flash*	**13.7**	**5.91**	Not required

Table 11: **Runtime comparison on 4× super-resolution** using a single RTX 4090.

C.7 Runtime Analysis

Table 12 reports detailed runtime, throughput, and peak memory of AVIS *Flash* for each degradation, measured on 81-frame videos using a single RTX 4090. The total runtime is decomposed into initialization, sampling, and VAE decoding. AVIS *Flash* restores the first chunk within 4 to 7 seconds and sustains 4.9 to 5.9 FPS thereafter, with peak memory below 20 GB. Across all degradations, the latency and throughput remain far better than those of the baselines in Table 1.

Task	Initialization (s)	Sampling (s)	VAE Decoding (s)	Latency (s)	FPS (frame/s)	Memory (GB)
Super Resolution	2.47	7.13	4.10	4.07	5.9	18.4
Inpainting	5.16	7.28	4.10	6.79	4.9	18.4
Gaussian Deblur	3.31	7.50	4.10	4.97	5.4	17.8
Temporal Average	3.57	7.26	4.10	5.19	5.4	19.9
Spatio-Temporal Average	3.87	7.11	4.10	5.47	5.4	18.4

Table 12: **Detailed runtime of AVIS *Flash* for each degradation.** Measured on 81-frame videos using a single RTX 4090.

C.8 Further Results on the Effect of the Initialization

We provide further results of the initialization ablation in Section 5.3 on the other four tasks under the strong noise setting (σ_n=0.05, λ=0.3). As shown in Table 13, AVIS achieves the best or tied-best results on every metric across all tasks, consistently improving over all three controls. These results demonstrate that the final restoration cannot be explained by the initialization or its VAE reconstruction alone.

Setting	PSNR↑	SSIM↑	LPIPS↓	FID↓	FVD↓
Super Resolution					
$\mathbf{x}_{\text{init}}^{\lambda}$	26.69	0.666	0.297	66.89	1066
$\mathcal{D}(\mathcal{E}(\mathbf{x}_{\text{init}}^{\lambda}))$	<u>27.03</u>	0.688	<u>0.289</u>	65.27	1027
AVIS w/o DDS	27.02	<u>0.690</u>	**0.271**	<u>59.74</u>	<u>831.6</u>
AVIS	**27.07**	**0.696**	**0.271**	**58.72**	**725.8**
Gaussian Deblur					
$\mathbf{x}_{\text{init}}^{\lambda}$	21.62	0.351	0.464	112.7	1585
$\mathcal{D}(\mathcal{E}(\mathbf{x}_{\text{init}}^{\lambda}))$	22.28	<u>0.386</u>	<u>0.454</u>	<u>110.8</u>	1254
AVIS w/o DDS	<u>22.31</u>	0.382	0.460	111.4	<u>1236</u>
AVIS	**23.77**	**0.423**	**0.410**	**89.71**	**591.1**
Temporal Average					
$\mathbf{x}_{\text{init}}^{\lambda}$	<u>27.60</u>	0.721	0.236	45.72	464.7
$\mathcal{D}(\mathcal{E}(\mathbf{x}_{\text{init}}^{\lambda}))$	27.52	<u>0.724</u>	<u>0.234</u>	<u>43.86</u>	<u>439.9</u>
AVIS w/o DDS	26.74	0.688	0.249	46.52	444.2
AVIS	**28.35**	**0.776**	**0.197**	**35.94**	**395.9**
Spatio-Temporal Average					
$\mathbf{x}_{\text{init}}^{\lambda}$	25.92	0.650	0.332	77.16	1393
$\mathcal{D}(\mathcal{E}(\mathbf{x}_{\text{init}}^{\lambda}))$	26.27	0.672	0.319	74.58	1328
AVIS w/o DDS	<u>26.31</u>	<u>0.673</u>	<u>0.304</u>	<u>69.28</u>	<u>1045</u>
AVIS	**26.41**	**0.686**	**0.295**	**67.57**	**933.3**

Table 13: **Further results on the effect of the initialization** (σ_n=0.05, λ=0.3). AVIS consistently improves over the initialization and its VAE reconstruction. **Bold** and <u>underline</u> indicate the best and second-best results, respectively.

C.9 Qualitative Evaluation

We provide comprehensive qualitative comparisons for each degradation type, with detailed discussions provided in the respective figure captions. All figures are best viewed zoomed in. For complete video comparisons, please refer to our project page.

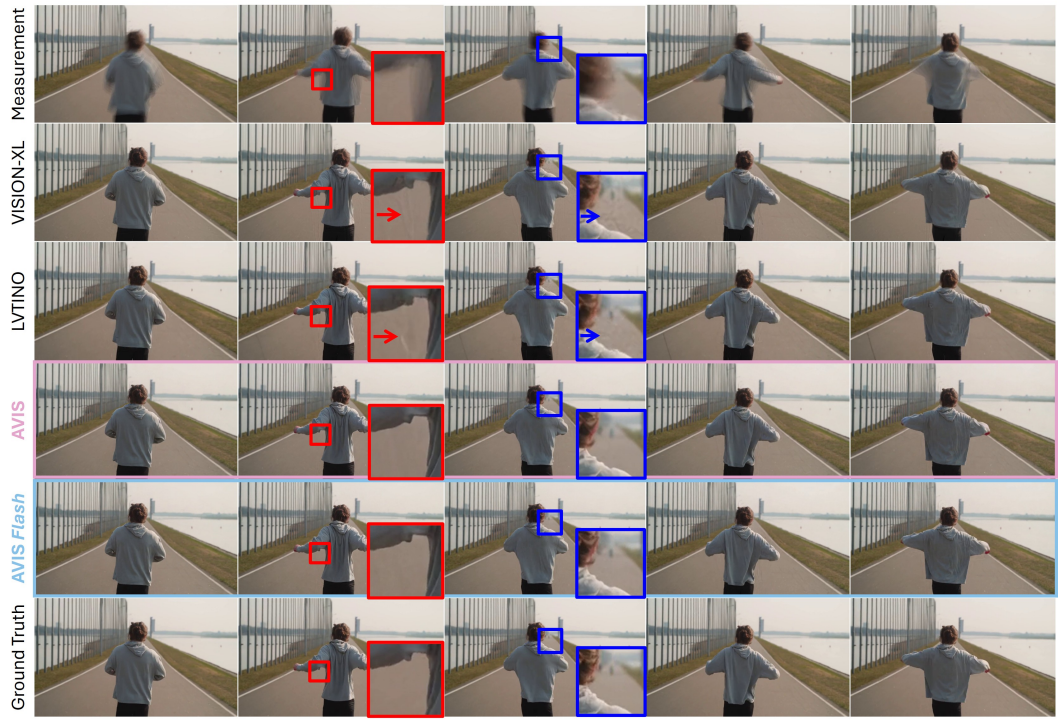

Figure 6: **Qualitative comparisons on temporal averaging.** VISION-XL and LVTINO suffer from noticeable artifacts in the highlighted regions. In contrast, AVIS recovers the most plausible details. Notably, despite its much faster inference, AVIS *Flash* maintains visual quality comparable to AVIS, preserving overall structural integrity.

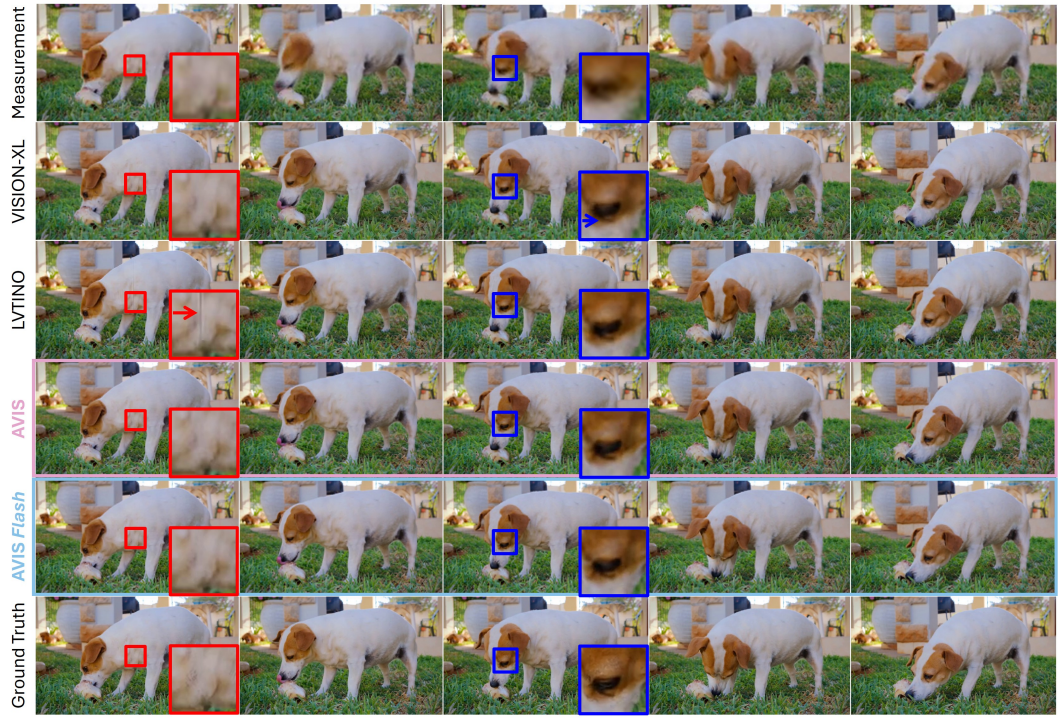

Figure 7: **Qualitative comparisons on spatio-temporal averaging.** LVTINO produces noticeable vertical artifacts (red arrow), and VISION-XL struggles to reconstruct fine details (*e.g.*, around the eye). In contrast, AVIS restores the most plausible details. Furthermore, despite its much higher throughput, AVIS *Flash* avoids the artifacts seen in the baselines, remaining highly competitive.

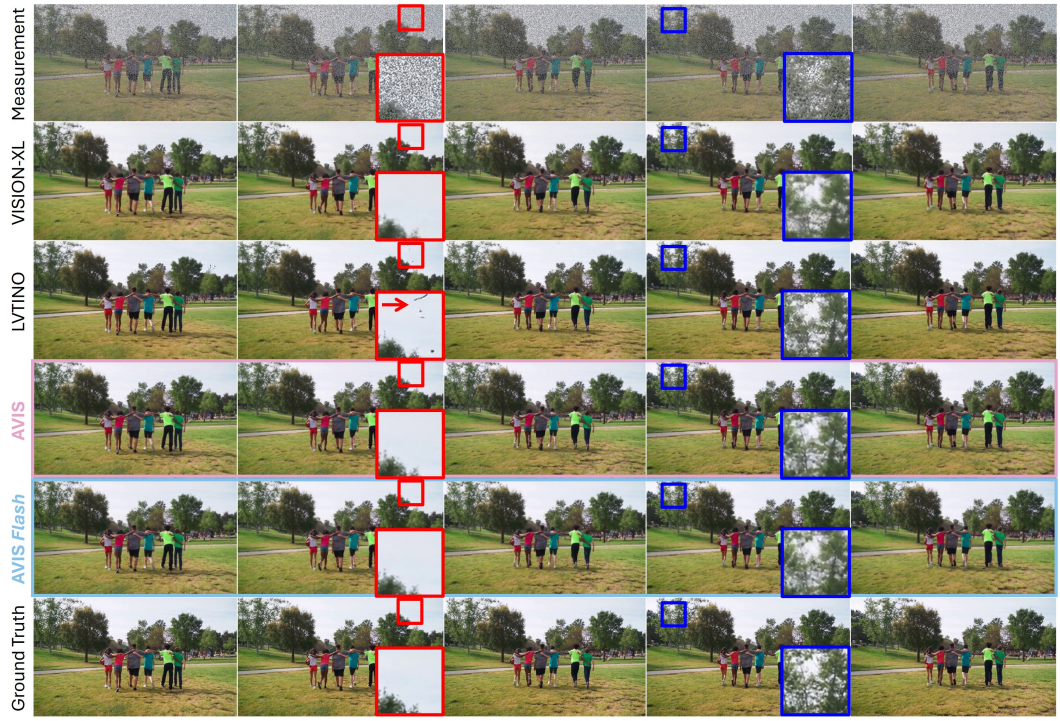

Figure 8: **Qualitative comparisons on inpainting.** LVTINO introduces unnatural floating artifacts in the restored sky region (red arrow), and VISION-XL yields overly smoothed, blurry textures when reconstructing the trees (blue box). In contrast, AVIS and AVIS *Flash* produce more plausible structures and preserve overall scene consistency.



Figure 9: **Qualitative comparisons on super-resolution.** LVTINO produces unnatural artifacts in the highlighted region (blue box). While VISION-XL and AVIS *Flash* yield slightly softer results, AVIS restores finer details. Notably, even with its highly accelerated inference, AVIS *Flash* successfully preserves structural integrity without severe artifacts.

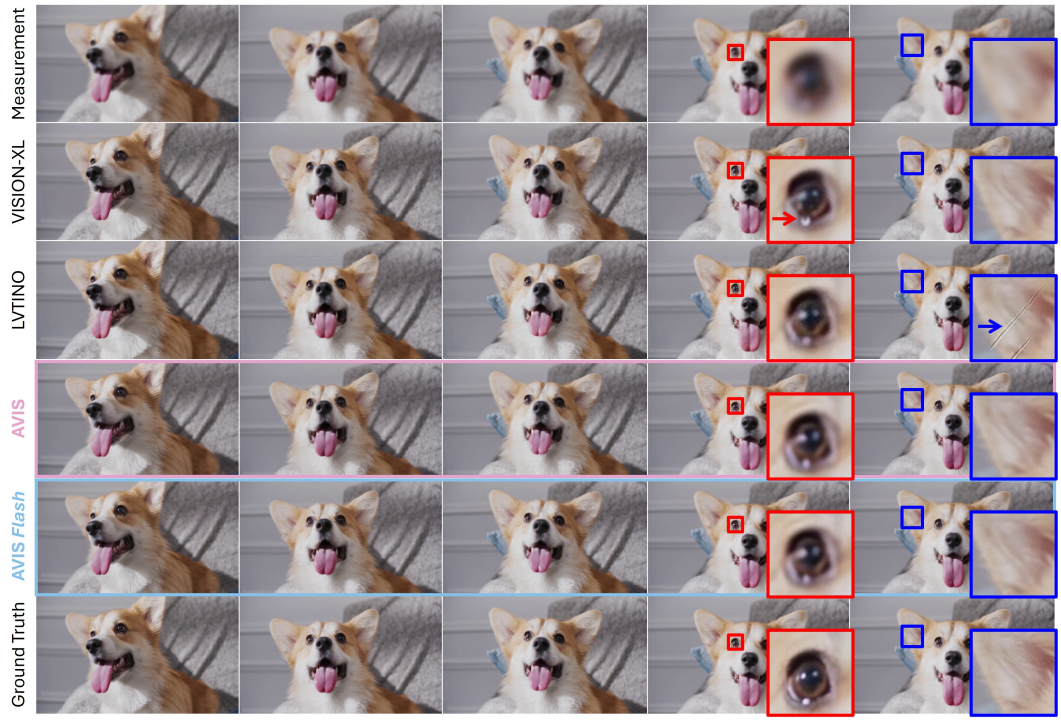

Figure 10: **Qualitative comparisons on Gaussian deblurring.** VISION-XL introduces minor distortions (red box), and LVTINO creates unnatural artifacts (blue box). In contrast, our proposed AVIS and AVIS *Flash* faithfully preserve fine details and structural integrity.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction clearly reflect the scope and contributions of our proposed method. Specifically, we introduce a novel Autoregressive Video Inverse problem Solver (AVIS) framework that leverages autoregressive video diffusion models to resolve the inefficiencies of prior works. The claims regarding the efficient deployment of AVIS are consistently reflected in the main text.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss the limitations in Section 6.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: All assumptions and their justifications are stated with each theorem, and full proofs appear in Appendix A.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: Section 5 and Appendix B provide detailed information regarding the datasets, inverse problems, baselines, evaluation metrics, implementation details, and computing resources. The source code and scripts are also provided with exact settings to reproduce the results.

Guidelines:

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

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: The code and scripts are provided with exact settings to reproduce the results. The datasets were sourced from Pexels, which is a publicly accessible platform.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: All necessary experimental and implementation details are provided in Section 5 and Appendix B.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: Error bars are not provided in this work.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We provide detailed information regarding the computing resources utilized for our experiments in Section 5 and Appendix B.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: This research adheres to all points of the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We discuss the potential societal impacts in Section 6.

Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: We do not release any pre-trained language models, image generators, or scraped datasets that have a high risk for misuse. Our work solely introduces an algorithmic framework (AVIS) that utilizes existing pre-trained models.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We provide appropriate citations for all code, datasets, and models utilized in our work.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.

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

13. **New assets**

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

    Answer: [Yes]

    Justification: The paper submission contains the implementation code which is documented.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The work does not involve any research with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The work does not involve any research with human subjects.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.