Accelerating Video Inverse Problem Solvers with Autoregressive Diffusion Models

Taesung Kwon [*1] **Jonghyun Park** [*1] **Jong Chul Ye** [1]

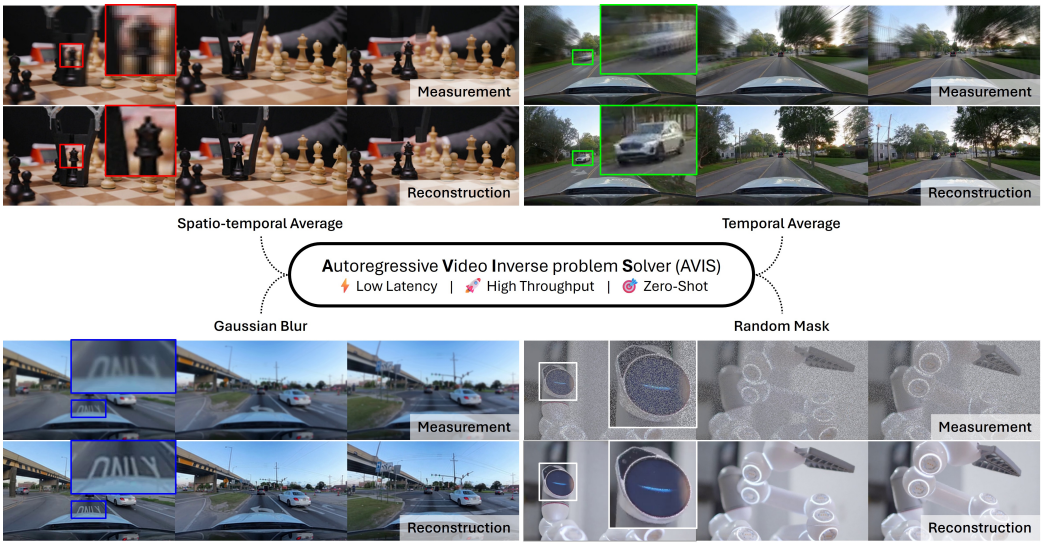

Figure 1. The AVIS framework leverages autoregressive video diffusion models to restore videos in a streaming manner, naturally eliminating latency bottlenecks. Examples show diverse video inverse problems that AVIS can solve in a zero-shot manner.

Abstract

Autoregressive video diffusion models enable low-latency generation by producing videos chunk by chunk, yet their potential for measurement-conditioned long-horizon video restoration remains underexplored. Existing diffusion-based video inverse solvers typically restore all frames holistically, delaying the first output frame until the entire video is reconstructed, and require repeated VAE passes for measurement consistency. We propose **AVIS**, a measurement-conditioned autoregressive video diffusion framework for streaming restoration. AVIS starts reverse diffusion from a measurement-consistent initialization and restores each chunk autoregressively with KV-cache conditioning. We further introduce **AVIS *Flash***, which applies measurement guidance only to the first chunk and restores later chunks through guidance-free autoregressive propagation from the corrected prefix. On five video restoration tasks, AVIS reduces initial latency from 114s to 4s and improves throughput from 0.71 to 1.18 FPS compared to a leading non-autoregressive solver, while achieving stronger restoration quality. AVIS *Flash* further increases throughput to 5.91 FPS on a single RTX 4090 GPU, offering a practical efficiency–quality trade-off for real-time long-horizon video restoration.

1. Introduction

Long-horizon video generation and restoration require models that can maintain temporal consistency while producing usable outputs with low latency. This is especially important for video inverse problems, where the goal is to reconstruct a clean video $\mathbf{x}$ from a degraded measurement $\mathbf{y}$:

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

where $\mathcal{A}$ is a degradation operator such as downsampling, masking, or blurring. Video diffusion models provide powerful priors for such ill-posed restoration tasks, but current diffusion-based video inverse solvers remain poorly matched to interactive or real-time long-video settings.

[1]Graduate School of AI, KAIST, South Korea. Correspondence to: Jong Chul Ye <jong.ye@kaist.ac.kr>.

Proceedings of the 43^{rd} International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026.

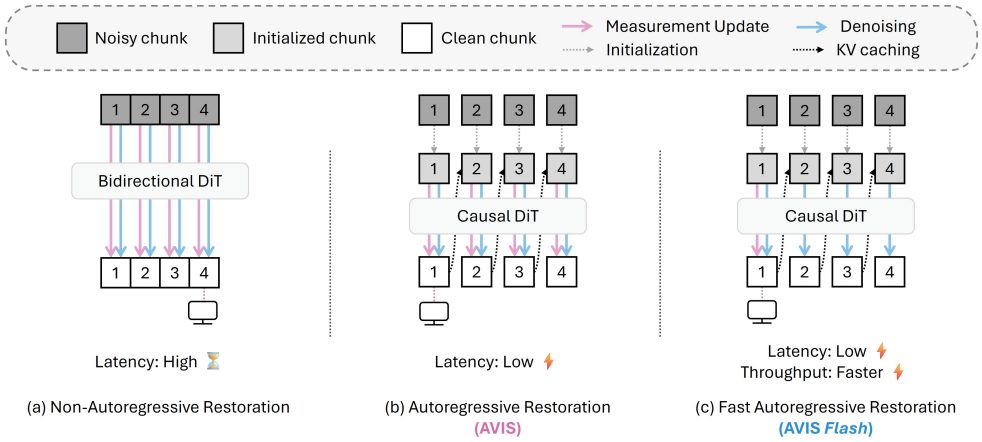

Figure 2. **Overview of our proposed AVIS and AVIS *Flash* framework.** (a) Non-autoregressive restoration processes the entire video holistically, suffering from high initial latency. (b) AVIS restores videos in a streaming manner and reduces sampling steps via measurement-consistent initialization 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.

Two bottlenecks are central. First, most existing solvers restore the entire video holistically, so the first frame cannot be displayed until all frames have been sampled. This creates an initial-latency bottleneck that becomes increasingly problematic as the video length grows. Second, modern video diffusion backbones operate in VAE latent space, while measurement consistency is often enforced in pixel space, requiring repeated VAE passes that restrict throughput.

We revisit video restoration as *streaming measurement-conditioned autoregressive video generation*. Instead of restoring all frames jointly, we use an autoregressive (AR) video diffusion backbone that generates short video chunks sequentially, conditioning each chunk on the restored prefix through KV caching. This naturally reduces latency: once the first chunk is restored, it can be displayed immediately while later chunks continue to be generated.

We propose **AVIS**, an **A**utoregressive **V**ideo **I**nverse problem **S**olver. AVIS first computes a coarse measurement-consistent estimate and uses it to initialize reverse diffusion at an intermediate timestep t_0, avoiding expensive generation from pure noise. It then restores the video chunk by chunk, combining autoregressive prefix conditioning with measurement guidance. To further improve throughput, we introduce **AVIS *Flash***, which applies measurement guidance only to the first chunk and restores later chunks through guidance-free autoregressive propagation from the corrected prefix. This design exploits the memory mechanism of the AR model while retaining a measurement-grounded initialization for stable long-horizon restoration.

Our contributions are:

- **AVIS.** We investigate autoregressive video diffusion models for streaming, measurement-conditioned video restoration. AVIS starts from a measurement-consistent initialization and enforces measurement consistency for every chunk, reducing sampling cost while enabling low-latency output.
- **AVIS *Flash*.** We introduce a faster variant that retains the same initialization but applies measurement guidance only to the first chunk. Later chunks are restored through autoregressive propagation from the corrected prefix, substantially improving throughput for long-horizon restoration.

2. Method

We present AVIS as a streaming, measurement-conditioned AR video diffusion framework. We instantiate AVIS with Self-Forcing as the AR video diffusion backbone. The method consists of three components: the AR backbone, a measurement-consistency update, and a measurement-consistent initialization. We then introduce AVIS *Flash*, which removes most repeated measurement updates by relying on first-chunk guidance and AR propagation.

2.1. Autoregressive Video Diffusion Backbone

Let $\mathbf{z} = [\mathbf{z}^1, \ldots, \mathbf{z}^N]$ denote a sequence of latent video chunks. An AR video diffusion model factorizes the video distribution as

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

In practice, the past context $\mathbf{z}^{<n}$ is represented by a KV cache $\mathrm{KV}^{<n}$, enabling efficient sequential generation without re-processing all previous chunks.

The Self-Forcing backbone uses a flow-matching sampling parameterization. For the n-th chunk, let $\mathbf{z}_0^n$ be the clean

latent and $\mathbf{z}_1^n \sim \mathcal{N}(0, I)$ be Gaussian noise. The noisy state at timestep t is

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

Given a learned vector field $\mathbf{v}_\theta$, the clean estimate is

$$\hat{\mathbf{z}}_{0|t}^n = \mathbf{z}_t^n - t\mathbf{v}_\theta(\mathbf{z}_t^n, t; \mathrm{KV}^{<n}). \tag{4}$$

The next step is obtained by re-noising the clean estimate,

$$\mathbf{z}_{t-\Delta t}^n = (1-(t-\Delta t))\hat{\mathbf{z}}_{0|t}^n + (t-\Delta t)\mathbf{z}_1^n. \tag{5}$$

2.2. Guidance for Measurement Consistency

During restoration, each clean estimate should be both video-realistic and consistent with the measurement $\mathbf{y}$. Directly differentiating $\|\mathbf{y} - \mathcal{A}(\mathcal{D}(\mathbf{z}))\|^2$ through the VAE decoder $\mathcal{D}$ and diffusion backbone is costly for high-resolution video. We therefore enforce measurement consistency through a pixel-space proximal correction. At each timestep, we decode the clean latent estimate,

$$\hat{\mathbf{x}}_{0|t}^n = \mathcal{D}(\hat{\mathbf{z}}_{0|t}^n), \tag{6}$$

and solve a MAP-style local measurement correction

$$\tilde{\mathbf{x}}_{0|t}^n := \arg\min_{\mathbf{x}} \frac{\gamma}{2}\|\mathbf{y} - \mathcal{A}(\mathbf{x})\|^2 + \frac{1}{2}\|\mathbf{x} - \hat{\mathbf{x}}_{0|t}^n\|^2. \tag{7}$$

Here, γ controls the strength of measurement consistency. We use conjugate-gradient (CG) updates and fix $\gamma = 1$ to avoid complex parameter tuning. The corrected pixel-space estimate is re-encoded as

$$\tilde{\mathbf{z}}_{0|t}^n = \mathcal{E}(\tilde{\mathbf{x}}_{0|t}^n), \tag{8}$$

and substituted into Eq. (5). This decode–correct–encode update avoids neural Jacobian computation while enforcing measurement consistency in pixel space.

2.3. AVIS: Efficient Restoration from a Measurement-Consistent Starting Point

Starting reverse diffusion from pure noise requires many steps and can amplify temporal drift in an autoregressive setting. AVIS instead starts from a measurement-consistent initialization. We first compute a coarse estimate

$$\mathbf{x}_{\text{init}} = \arg\min_{\mathbf{x}} \|\mathbf{y} - \mathcal{A}(\mathbf{x})\|^2, \tag{9}$$

encode it as $\mathbf{z}_{\text{init}} = \mathcal{E}(\mathbf{x}_{\text{init}})$, and diffuse it to an intermediate timestep t_0:

$$\mathbf{z}_{t_0} = (1-t_0)\mathbf{z}_{\text{init}} + t_0\mathbf{z}_1, \quad \mathbf{z}_1 \sim \mathcal{N}(0, I). \tag{10}$$

Although $\mathbf{x}_{\text{init}}$ is measurement-consistent, it may lack perceptual realism. The AR reverse diffusion process refines this estimate toward the video prior, while the guidance in Eq. (7) keeps the trajectory faithful to the measurement. AVIS applies this guidance to every chunk, yielding a high-quality streaming solver: after each chunk is restored, it is displayed and stored in the prefix cache for subsequent chunks.

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 $\in$ {AVIS, AVIS *Flash*}.
$\mathbf{x}_{\text{init}} = \arg\min_{\mathbf{x}} \|\mathbf{y} - \mathcal{A}(\mathbf{x})\|^2$
$\mathbf{z}_{\text{init}} \leftarrow \mathcal{E}(\mathbf{x}_{\text{init}})$ ▷ Pre-restoration
$\mathrm{KV}^0 \leftarrow \emptyset$
for $n = 1 : N$ **do**
  $\mathbf{z}_{t_0}^n \leftarrow (1-t_0)\mathbf{z}_{\text{init}}^n + t_0\mathbf{z}_1,\ \mathbf{z}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ▷ Initialization
  for $t : t_0 \to 0$ **do**
    $\hat{\mathbf{z}}_{0|t}^n \leftarrow \mathbf{z}_t^n - t\mathbf{v}_\theta(\mathbf{z}_t^n, t; \mathrm{KV}^{<n})$
    if mode **is** AVIS **or** (mode **is** AVIS *Flash* and $n = 1$) **then**
      $\hat{\mathbf{z}}_{0|t}^n \leftarrow$ Solve Eq. (7) via CG
    end if
    $\mathbf{z}_{t-\Delta t}^n \leftarrow (1-(t-\Delta t))\hat{\mathbf{z}}_{0|t}^n + (t-\Delta t)\mathbf{z}_1,\ \mathbf{z}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  end for
  Display $\mathcal{D}(\mathbf{z}_0^n)$ **and** $\mathrm{KV}^{\le n} \leftarrow$ Update $\mathrm{KV}(\mathbf{z}_0^n, \mathrm{KV}^{<n})$
end for
return $[\mathbf{z}_0^1, \ldots, \mathbf{z}_0^N]$

2.4. AVIS *Flash*: Accelerating AVIS via AR Propagation

AVIS *Flash* further improves throughput by eliminating repeated proximal measurement updates after the first chunk. It shares the same measurement-consistent initialization at t_0 with AVIS for every chunk, but applies Eq. (7) only when $n = 1$. Thus, later chunks are not generated from pure noise; they start from the noised latent of $\mathbf{x}_{\text{init}}$ and are refined by the AR prior. Once the first chunk is grounded to the measurement, its latent features are stored in the prefix cache KV^1. For all later chunks ($n \ge 2$), AVIS *Flash* performs proximal-update-free AR sampling using Eq. (4) and Eq. (5), conditioned on the cache $\mathrm{KV}^{<n}$. The measurement-consistent initialization anchors each chunk to the measurement, while the corrected prefix provides a temporal anchor that propagates structure across chunks.

Chunk-wise error decomposition. To clarify the intuition, we use the following local decomposition. 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 AR vector field $\mathbf{v}_\theta$, the final error ϵ_K^n of the n-th chunk after K sampling steps satisfies

$$\epsilon_K^n \le \Lambda_K \epsilon_0^n + B_K \delta_n, \tag{11}$$

where Λ_K and B_K depend on the sampling schedule and local Lipschitz constants. This decomposition is interpretive rather than a finite-step guarantee: the initialization reduces ϵ_0^n, while the restored prefix affects later chunks through the additive context term $B_K\delta_n$. This suggests a quality-throughput continuum: AVIS uses per-chunk guidance, AVIS *Flash* uses first-chunk guidance, and periodic re-injection offers an middle ground for long sequences.

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 1. **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.

3. Experiments

Setup. We evaluate on 100 high-resolution videos from Pexels, resized to 480×854 and cropped to 81 frames. We consider five restoration tasks: $4\times$ super-resolution, random inpainting with 50% masking, Gaussian deblurring, temporal averaging, and spatio-temporal averaging. We compare against DiffIR2VR-Zero, VISION-XL, and LVTINO. We report distortion and perceptual metrics, VBench video-quality metrics, and efficiency metrics including initial latency, total time, and throughput. All experiments are run on a single NVIDIA RTX 4090 with $t_0 = 0.1$ and $K = 2$.

Efficiency. Table 2 shows the main efficiency comparison on $4\times$ video super-resolution. Non-autoregressive baselines must restore the full video before displaying the first frame, resulting in 114–167s initial latency for recent high-quality solvers. In contrast, AVIS and AVIS *Flash* restore videos chunk by chunk and reduce initial latency to 4s. AVIS improves throughput from 0.71 FPS to 1.18 FPS relative to LVTINO, while AVIS *Flash* reaches 5.91 FPS on a single RTX 4090 by removing guidance from later chunks. On a single NVIDIA H100, AVIS *Flash* is further accelerated to 10.2 FPS.

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

Table 2. **Efficiency comparison for $4\times$ video super-resolution.** AVIS enables streaming restoration, and AVIS *Flash* substantially improves throughput by avoiding repeated guidance for later chunks.

Long-horizon and controllable restoration. To better match long-horizon video settings, we additionally evaluate AVIS *Flash* on longer video sequences in the supplementary material. These results show that periodic measurement re-injection can suppress drift and maintain stable autoregressive restoration beyond the default 81-frame setting. We also include an exploratory proof-of-concept for inference-time novel view synthesis, where depth-based warping under target camera trajectories produces disoccluded regions and AVIS *Flash* acts as a trajectory-conditioned inpainting module. We treat this as a preliminary demonstration of controllable video editing rather than a primary benchmark.

3.1. Ablation Study

To isolate the contributions of our core components, we report the ablation study results on AVIS *Flash* for video inpainting in Table 3.

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 3(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,

Setting	PSNR↑	SSIM↑	LPIPS↓	FVD↓	FID↓	Sub. C.↑	Bg. C.↑	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 3. **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.

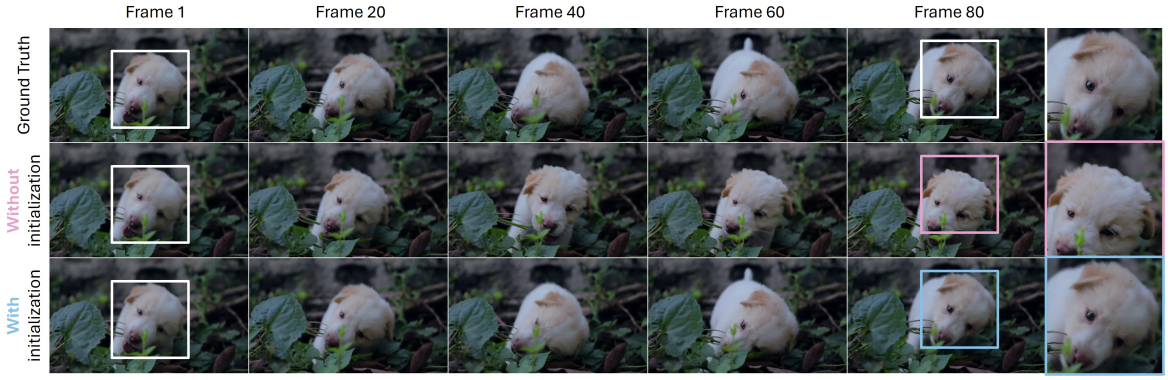

Figure 3. **Effect of AR propagation and initialization.** (Top) Ground-truth video. (Middle) While AR propagation preserves the preceding context, it exhibits gradual error accumulation over time. (Bottom) Our initialization for the reverse diffusion effectively mitigates this temporal drift.

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. For longer videos, optional periodic re-injection of measurement consistency can further trade throughput for long-range fidelity, as described in Section 2.4.

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 3(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 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 3(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.

4. Conclusion

We presented AVIS and AVIS *Flash*, measurement-conditioned AR video diffusion frameworks for streaming video restoration. By combining measurement-consistent initialization, chunk-wise generation, and first-chunk guidance, our approach reduces latency while maintaining strong restoration quality, suggesting AR diffusion as a promising foundation for efficient long-horizon restoration.

References

Chang, P., Tang, J., Gross, M., and Azevedo, V. C. 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.

Chung, H., Kim, J., Mccann, M. T., Klasky, M. L., and Ye, J. C. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=OnD9zGAGT0k.

Chung, H., Lee, S., and Ye, J. C. 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.

Daras, G., Nie, W., Kreis, K., Dimakis, A. G., Mardani, M., Kovachki, N. B., and Vahdat, A. Warped diffusion: Solving video inverse problems with image diffusion models. *Advances in Neural Information Processing Systems*, 37:101116–101143, 2024.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Ge, S., Mahapatra, A., Parmar, G., Zhu, J.-Y., and Huang, J.-B. On the content bias in fréchet video distance. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7277–7288, 2024.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Huang, X., Li, Z., He, G., Zhou, M., and Shechtman, E. 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.

Huang, Z., He, Y., Yu, J., Zhang, F., Si, C., Jiang, Y., Zhang, Y., Wu, T., Jin, Q., Chanpaisit, N., et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21807–21818, 2024.

Kim, J., Kim, B. S., and Ye, J. C. Flowdps: Flow-driven posterior sampling for inverse problems. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12328–12337, 2025.

Kwon, T. and Ye, J. C. Solving video inverse problems using image diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=TRWxFUzK9K.

Kwon, T. and Ye, J. C. Vision-xl: High definition video inverse problem solver using latent image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10465–10474, 2025b.

Kwon, T., Song, G., Kim, Y., Kim, J., Ye, J. C., and Jang, M. Video diffusion posterior sampling for seeing beyond dynamic scattering layers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Lin, X., He, J., Chen, Z., Lyu, Z., Dai, B., Yu, F., Qiao, Y., Ouyang, W., and Dong, C. Diffbir: Toward blind image restoration with generative diffusion prior. In *European conference on computer vision*, pp. 430–448. Springer, 2024.

Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.02747.

Liu, X., Gong, C., 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.

Mardani, M., Song, J., Kautz, J., and Vahdat, A. 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.

Park, J. and Ye, J. C. Flowlps: Langevin-proximal sampling for flow-based inverse problem solvers. *arXiv preprint arXiv:2512.07150*, 2025.

Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., and Rombach, R. 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.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Rout, L., Raoof, N., Daras, G., Caramanis, C., Dimakis, A., and Shakkottai, S. Solving linear inverse problems provably via posterior sampling with latent diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

Song, B., Kwon, S. M., Zhang, Z., Hu, X., Qu, Q., and Shen, L. 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.

Song, J., Vahdat, A., Mardani, M., and Kautz, J. Pseudoinverse-guided diffusion models for inverse problems. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=9_gsMA8MRKQ.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2020.

Spagnoletti, A., Almansa, A., and Pereyra, M. 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.

Teed, Z. and Deng, J. 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*, pp. 402–419. Springer, 2020.

Unterthiner, T., van Steenkiste, S., Kurach, K., Marinier, R., Michalski, M., and Gelly, S. FVD: A new metric for video generation, 2019. URL https://openreview.net/forum?id=rylgEULtdN.

Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.-W., Chen, D., Yu, F., Zhao, H., Yang, J., et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Wang, Y., Yu, J., and Zhang, J. 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.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Yeh, C.-H., Lin, C.-Y., Wang, Z., Hsiao, C.-W., Chen, T.-H., and Liu, Y.-L. Diffir2vr-zero: Zero-shot video restoration with diffusion-based image restoration models. *arXiv preprint arXiv:2407.01519*, 2024.

Yin, T., Gharbi, M., Park, T., Zhang, R., Shechtman, E., Durand, F., and Freeman, W. T. Improved distribution matching distillation for fast image synthesis. *Advances in neural information processing systems*, 37:47455–47487, 2024.

Yin, T., Zhang, Q., Zhang, R., Freeman, W. T., Durand, F., Shechtman, E., and Huang, X. From slow bidirectional to fast autoregressive video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22963–22974, 2025.

Zhang, B., Chu, W., Berner, J., Meng, C., Anandkumar, A., and Song, Y. Improving diffusion inverse problem solving with decoupled noise annealing. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 20895–20905, 2025.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

A. Additional Experimental Results

A.1. Long Video Restoration

We further evaluate AVIS *Flash* on a one-minute video consisting of 960 frames at 16 FPS. For long sequences, context errors may gradually accumulate as restoration proceeds autoregressively. To study an intermediate operating point between AVIS and AVIS *Flash*, we periodically re-inject measurement guidance every 7 chunks. As shown in Figure 4, this optional re-injection improves long-range stability while preserving much of the throughput advantage of AVIS *Flash*.

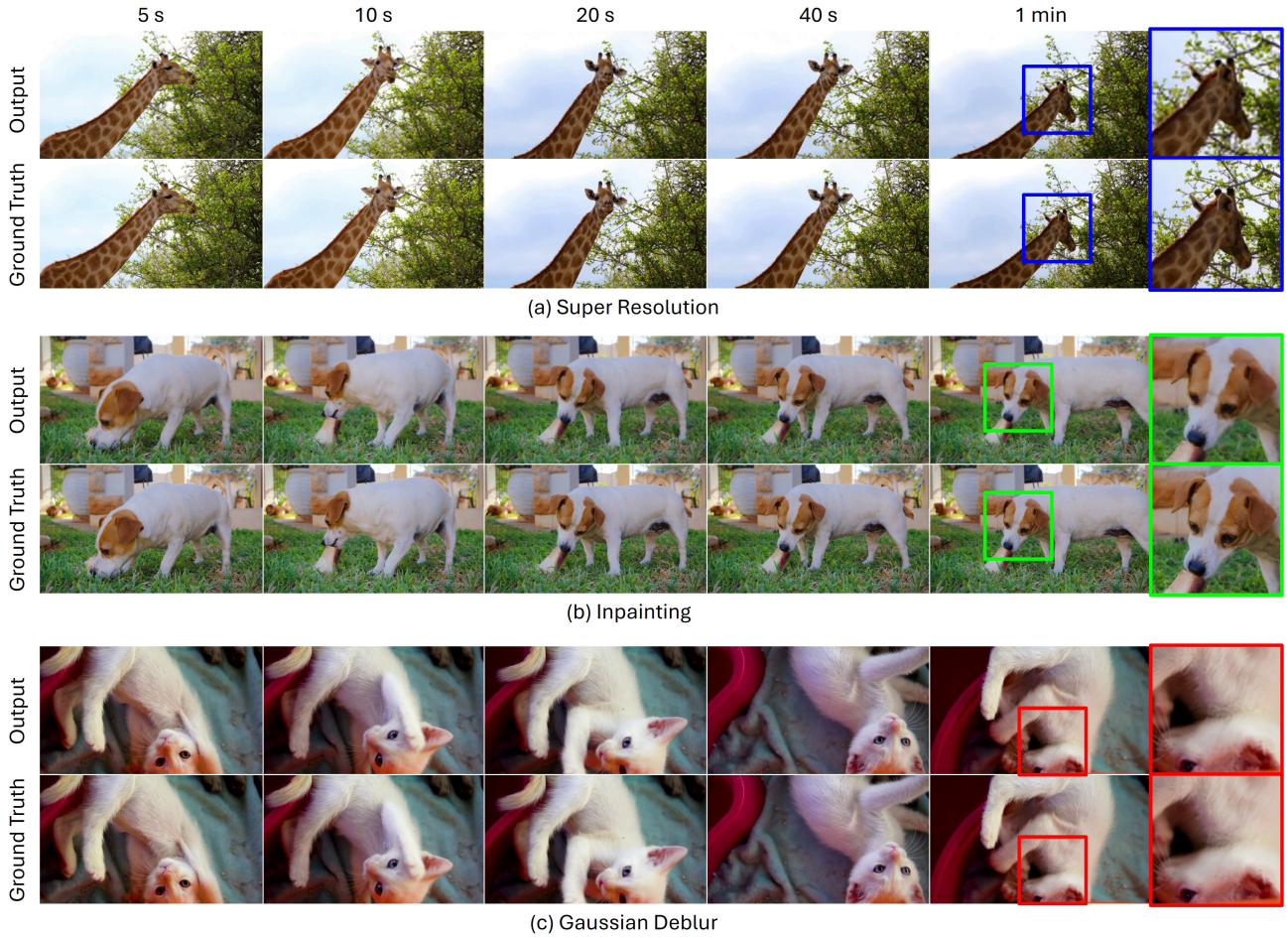

Figure 4. **Qualitative results of long video restoration.** By periodically re-injecting measurement consistency every 7 chunks, AVIS *Flash* prevents noticeable temporal drift and remains consistent with the ground truth even in later frames. The timestamps indicate the elapsed time within the long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

A.2. Exploratory Novel View Video Synthesis

We include this experiment as an exploratory proof-of-concept for inference-time controllable video editing. Following a depth-based geometric warping pipeline, we first estimate frame-wise depth maps, lift pixels into dynamic 3D point clouds, and reproject them according to a target camera trajectory. This reprojection exposes disoccluded regions, resulting in holes in the warped frames. AVIS *Flash* is then used as a trajectory-conditioned inpainting module to fill the missing regions. The results suggest potential applicability to inference-time novel-view video synthesis, although high-fidelity trajectory-controlled video generation remains an open direction.

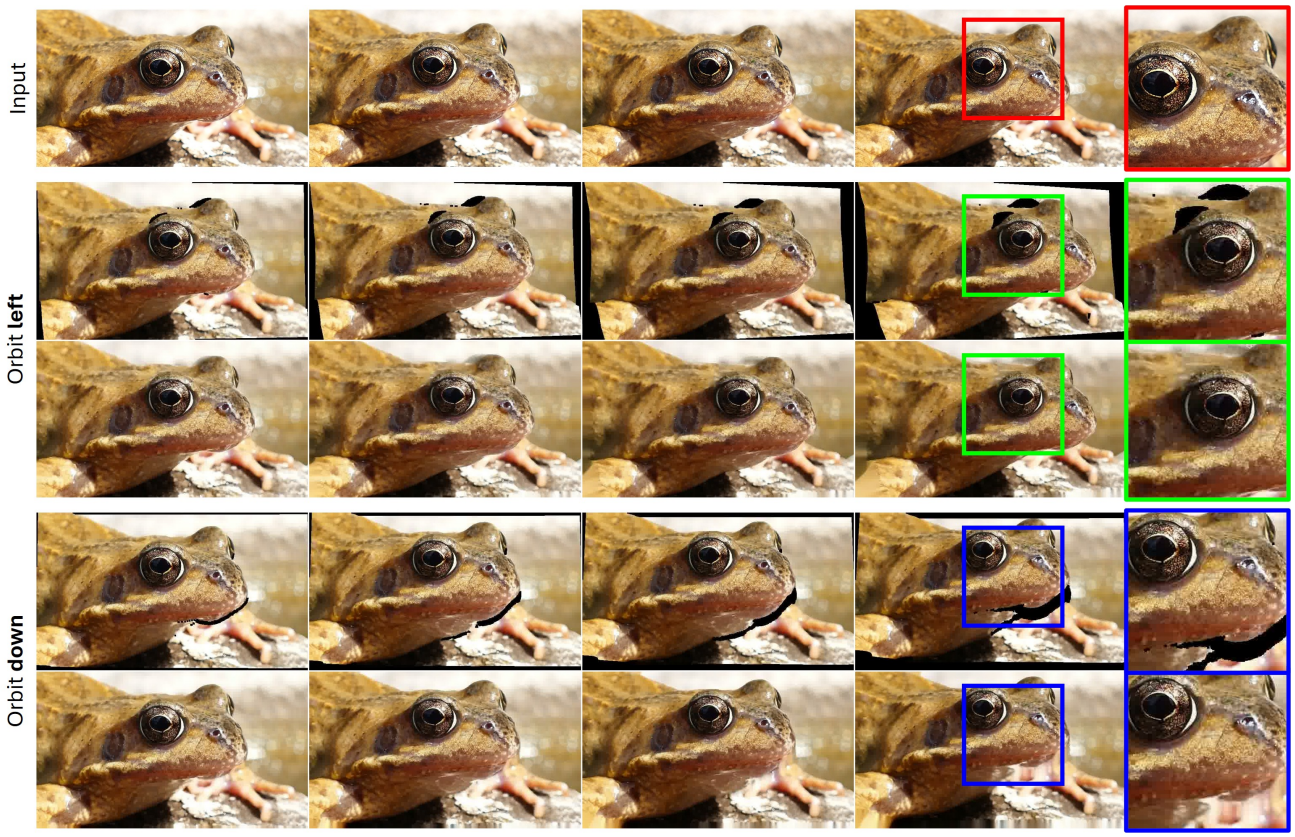

Figure 5. **Exploratory novel-view video synthesis.** We apply AVIS *Flash* as a trajectory-conditioned inpainting module for disoccluded regions produced by depth-based warping under target camera trajectories, including orbit-left and orbit-down motions. For each trajectory, the upper row shows the initial warped frames and the lower row shows the AVIS *Flash* outputs. This experiment is intended as a proof-of-concept rather than a primary benchmark.

A.3. Qualitative Comparisons

We provide additional qualitative comparisons for each degradation type. All figures are best viewed zoomed in. For complete video comparisons, please refer to our project page.

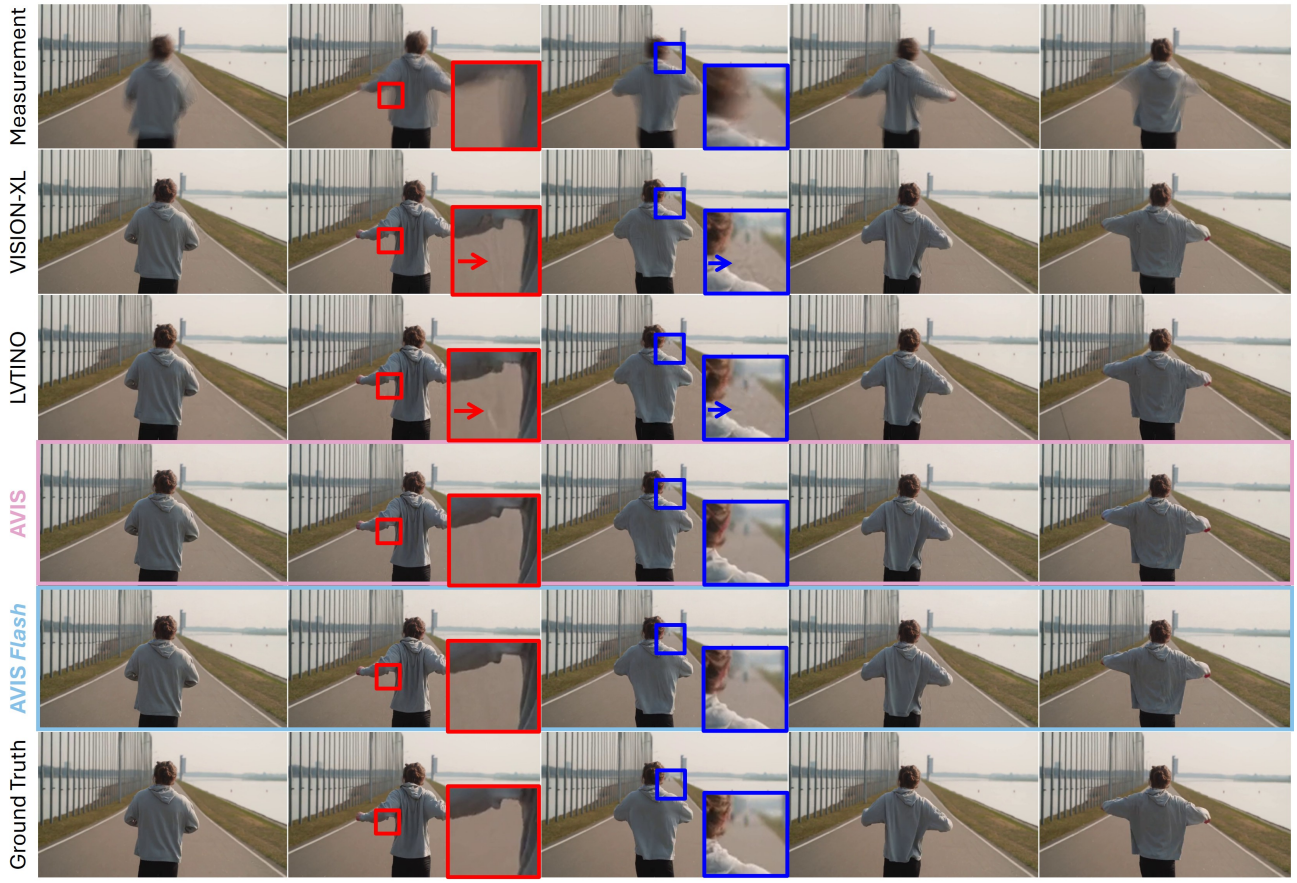

Figure 6. **Qualitative comparisons on temporal averaging.** VISION-XL and LVTINO suffer from noticeable artifacts in the highlighted regions. In contrast, AVIS recovers the most plausible details. Notably, despite its much faster inference, AVIS *Flash* maintains visual quality comparable to AVIS, preserving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

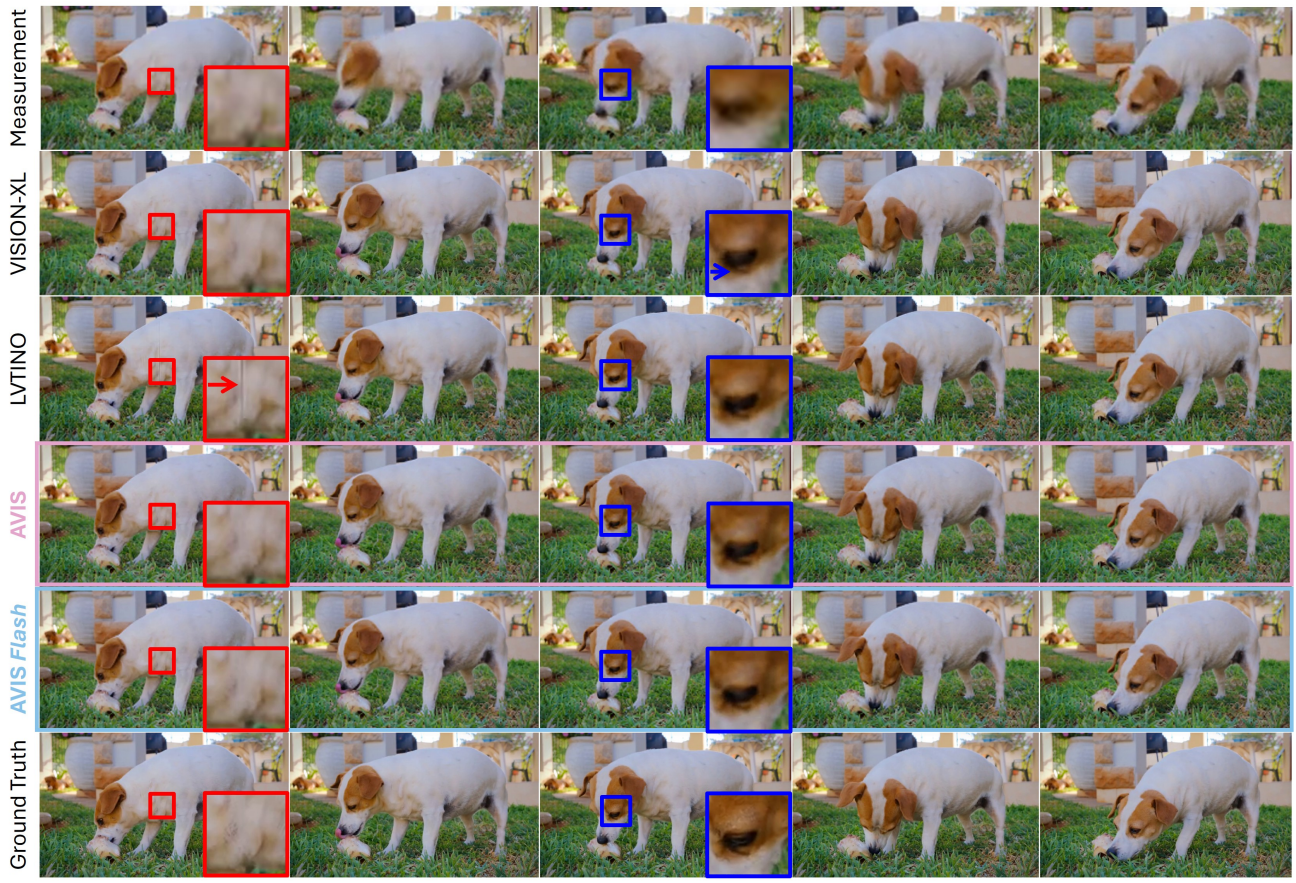

Figure 7. **Qualitative comparisons on spatio-temporal averaging.** LVTINO produces noticeable vertical artifacts (red arrow), and VISION-XL struggles to reconstruct fine details (e.g., around the eye). In contrast, AVIS restores the most plausible details. Furthermore, despite its much higher throughput,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

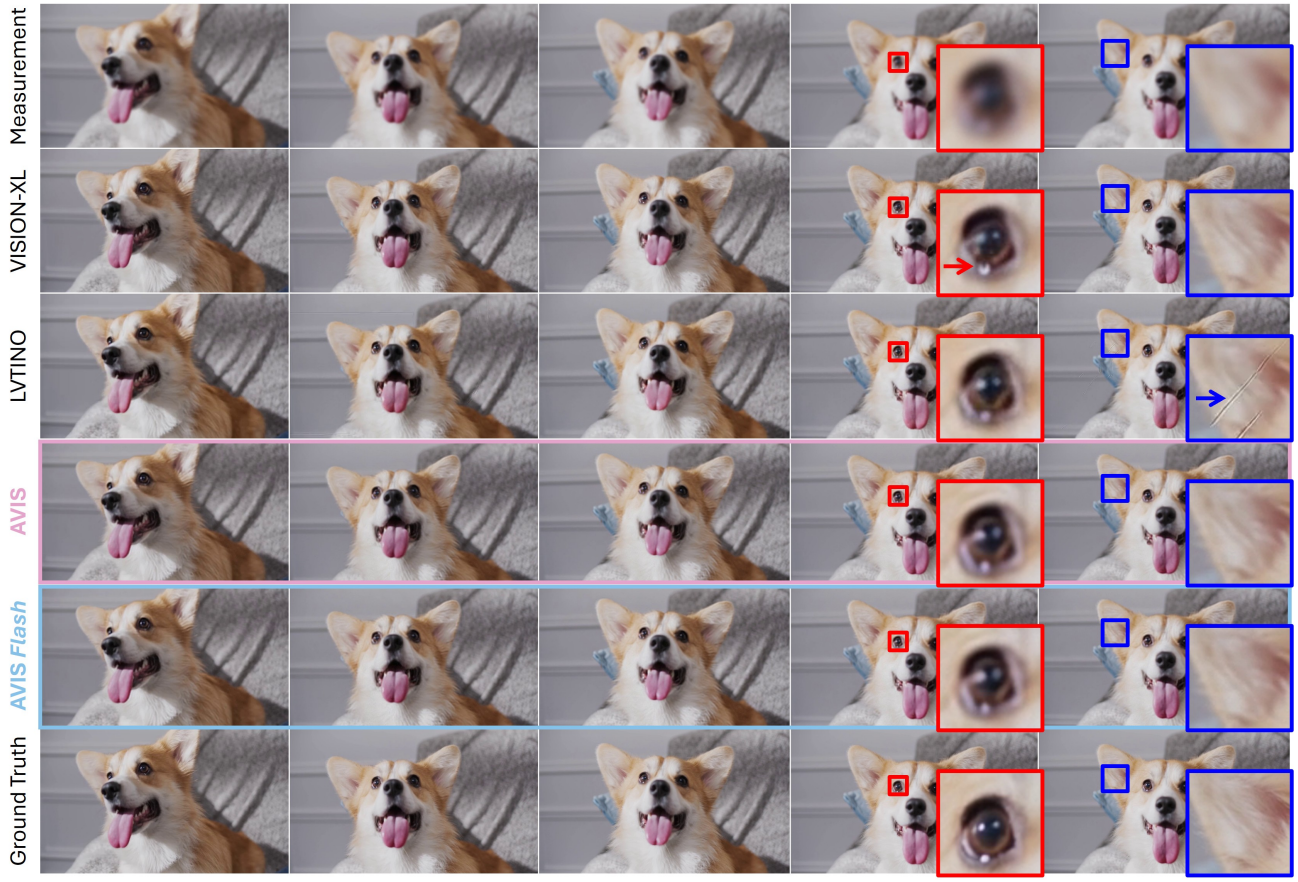

Figure 10. **Qualitative comparisons on Gaussian deblurring.** VISION-XL introduces minor distortions (red box), and LVTINO creates unnatural artifacts (blue box). In contrast, our proposed AVIS and AVIS *Flash* faithfully preserve fine details and structural integrity.

A.4. 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 LVTINO (Spagnoletti et al., 2026) with the same autoregressive backbone (Huang et al., 2025) used by AVIS on the $4\times$ super-resolution task. As shown in Table 4, replacing the original bidirectional prior of LVTINO with the AR backbone gives mixed results: PSNR and FVD improve, while SSIM, LPIPS, FID, and most VBench metrics degrade. Under this matched-backbone setting, AVIS still achieves stronger overall performance than LVTINO (AR), suggesting that its gains are not explained solely by the choice of video prior.

Method	PSNR↑	SSIM↑	LPIPS↓	FVD↓	FID↓	Sub. Con.↑	Bg. Con.↑	M. Smooth.↑	Aesth.↑	Imag.↑
LVTINO (AR)	<u>30.28</u>	0.818	0.113	<u>48.50</u>	<u>21.99</u>	96.13	95.75	<u>99.11</u>	51.27	<u>55.84</u>
LVTINO	30.04	<u>0.824</u>	<u>0.102</u>	59.42	**18.32**	**96.36**	<u>96.09</u>	<u>99.11</u>	**52.63**	**59.05**
AVIS	**30.38**	**0.826**	**0.101**	**40.36**	23.94	<u>96.30</u>	**96.21**	**99.24**	<u>52.19</u>	55.13

Table 4. **Backbone fairness check on $4\times$ super-resolution.** We evaluate LVTINO with the same AR backbone as AVIS. Although switching LVTINO to the AR backbone improves some metrics, AVIS still achieves stronger overall performance, indicating that its gains are not due only to the backbone choice. **Bold** and <u>underline</u> denote the best and second-best results.

B. Implementation Details

B.1. AVIS Implementation

Our implementation is built on the Self-Forcing codebase (Huang et al., 2025) with Wan2.1-T2V-1.3B (Wan et al., 2025). We retain the original Self-Forcing configuration except for restoration-specific settings such as the initial timestep t_0 and the number of sampling steps K. For video encoding, we use Wan-VAE (Wan et al., 2025), which has spatial and temporal downsampling factors of 8 and 4, respectively. Thus, an RGB video with $1 + T$ frames at resolution $H \times W$ is encoded into a latent tensor with $1 + T/4$ latent frames and spatial size $H/8 \times W/8$. Following the Self-Forcing setup, we generate $L = 3$ latent frames per chunk. Although the original framework uses 4 diffusion steps, we use $K = 2$ steps with $t_0 = 0.1$, which provides a favorable efficiency–quality trade-off for restoration.

Initial estimation. We obtain the initial estimate by optimizing Eq. (9) with conjugate gradient (CG) updates. The optimization is typically initialized from the measurement $\mathbf{y}$. For tasks with dimensional mismatch, we first align dimensions using simple preprocessing: bilinear interpolation for super-resolution and spatio-temporal averaging, and nearest-neighbor infilling for inpainting. We use 5 CG steps for Gaussian deblurring and super-resolution, 50 steps for temporal averaging, and 100 steps for spatio-temporal averaging. For inpainting, the initialized measurement already satisfies the measurement-consistency term, so no additional CG updates are used.

Inference and optimization. For the main experiments, we use $t_0 = 0.1$ and $K = 2$ sampling steps. For the measurement guidance in Eq. (7), we fix $\gamma = 1$ and use 5 CG updates across all degradation types.

Computing resources. On a single NVIDIA RTX 4090, AVIS *Flash* processes videos at 480×832 resolution with 4-second initial latency, 5.9 FPS throughput, and 18.4 GB VRAM usage. On a single NVIDIA H100, throughput increases to 10.18 FPS with 1.85-second initial latency and 26.79 GB VRAM usage.

B.2. Baseline Models

We compare against representative diffusion model-based video inverse problem solvers (DVIS):

- **DiffIR2VR-Zero** (Yeh et al., 2024) adapts pretrained image restoration diffusion models (Lin et al., 2024) to video restoration and maintains temporal consistency using hierarchical latent warping and token merging.
- **VISION-XL** (Kwon & Ye, 2025b) is a zero-shot high-definition video inverse solver based on latent image diffusion models (Podell et al., 2024), using pseudo-batch sampling and inversion strategies for temporal consistency.
- **LVTINO** (Spagnoletti et al., 2026) combines Video Consistency Models (Yin et al., 2025) and Image Consistency Models (Yin et al., 2024) for high-definition zero-shot video restoration.

We use official implementations for all baselines in the same environment as AVIS, with the same resolution, number of frames, and degradation operators. Since LVTINO exceeded the 24GB VRAM limit, we pre-compute text embeddings and remove the text encoders from both the image and video diffusion models during inference. This does not change the conditioning signal or the generated outputs compared to running the text encoders online; it only reduces inference-time memory usage. Therefore, our LVTINO results correspond to the full text-conditioned model and do not incur a performance drop from removing the text encoders.

Some recent DVIS methods were not included due to unsupported resolutions, high memory requirements, or unavailable public implementations. Nevertheless, VISION-XL and LVTINO are strong recent zero-shot solvers for high-definition video restoration, making them representative baselines for our setting.

B.3. Evaluation Metrics

We evaluate frame-wise restoration quality using PSNR, SSIM (Wang et al., 2004), LPIPS (Zhang et al., 2018), and FID (Heusel et al., 2017). We evaluate video quality using FVD (Unterthiner et al., 2019) and VBench (Huang et al., 2024), including Subject Consistency, Background Consistency, Motion Smoothness, Aesthetic Quality, and Imaging Quality. We use official or widely adopted PyTorch implementations: `pytorch-msssim` for SSIM, `lpips` for LPIPS, `pytorch-fid` for FID, a public FVD implementation (Ge et al., 2024), and the official VBench repository. All primary evaluations are conducted on a single NVIDIA RTX 4090 GPU; additional timing measurements are reported on a single NVIDIA H100 GPU.

C. Related Work

C.1. Diffusion for Video Inverse Problems

Diffusion model-based Inverse problem Solvers (DIS) (Chung et al., 2023; Song et al., 2023; Wang et al., 2023; Chung et al., 2024; Mardani et al., 2024; Song et al., 2024; Rout et al., 2024; Zhang et al., 2025; Kim et al., 2025; Park & Ye, 2025) enable repurposing the diffusion or flow-based models (Ho et al., 2020; Song et al., 2020; Dhariwal & Nichol, 2021; Rombach et al., 2022; Podell et al., 2024; Liu et al., 2023; Lipman et al., 2023; Esser et al., 2024) 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) (Daras et al., 2024; Kwon & Ye, 2025a;b; Kwon et al., 2025; Spagnoletti et al., 2026) extend this paradigm to the video modality. One line of work (Daras et al., 2024; Kwon & Ye, 2025a;b) repurposes image diffusion priors (Dhariwal & Nichol, 2021; Podell et al., 2024) for video restoration by introducing explicit temporal conditioning. Warping-based approaches (Chang et al., 2024; Daras et al., 2024) explicitly control temporal consistency using optical flow (Teed & Deng, 2020). An alternative approach enforces temporal consistency without optical flow through batch-consistent sampling (Kwon & Ye, 2025a;b), which efficiently synchronizes stochastic components across frames. More recently, methods leveraging native video diffusion priors (Kwon et al., 2025; Spagnoletti et al., 2026) 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.