L∀TINO: LAtent Video consisTency INverse sOlver for High Definition Video Restoration

Anonymous authors
Paper under double-blind review

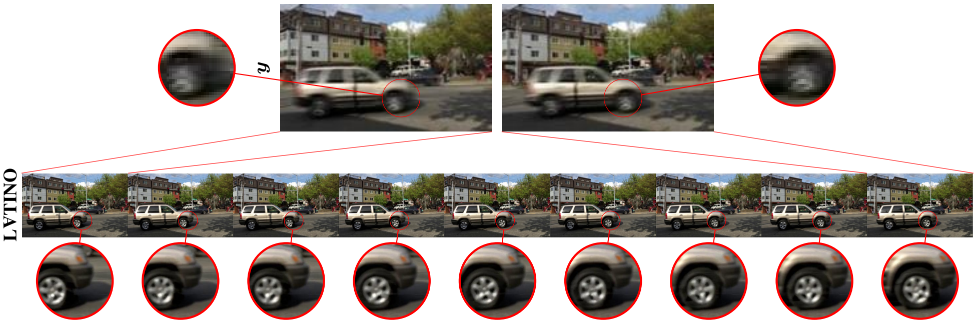


Figure 1: Results on joint spatial-temporal super-resolution by factor ×8.

Abstract

Computational imaging methods increasingly rely on powerful generative diffusion models to tackle challenging image restoration tasks. In particular, state-of-the-art zero-shot image inverse solvers leverage distilled text-to-image latent diffusion models (LDMs) to achieve unprecedented accuracy and perceptual quality with high computational efficiency. However, extending these advances to high-definition video restoration remains a significant challenge, due to the need to recover fine spatial detail while capturing subtle temporal dependencies. Consequently, methods that naively apply image-based LDM priors on a frame-by-frame basis often result in temporally inconsistent reconstructions. We address this challenge by leveraging recent advances in Video Consistency Models (VCMs), which distill video latent diffusion models into fast generators that explicitly capture temporal causality. Building on this foundation, we propose L∀TINO[1], the first zero-shot or plug-and-play inverse solver for high definition video restoration with priors encoded by VCMs. Our conditioning mechanism bypasses the need for automatic differentiation and achieves state-of-the-art video reconstruction quality with only a few neural function evaluations, while ensuring strong measurement consistency and smooth temporal transitions across frames. Extensive experiments on a diverse set of video inverse problems show significant perceptual improvements over current state-of-the-art methods that apply image LDMs frame by frame, establishing a new benchmark in both reconstruction fidelity and computational efficiency.

1 Introduction

We seek to recover an unknown video of interest $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T)$ from a noisy measurement

$$\boldsymbol{y} = \mathcal{A}\boldsymbol{x} + \boldsymbol{n}\,,$$

where $\mathcal{A}$ is a linear degradation operator acting on the full video sequence, $\boldsymbol{n}$ is additive Gaussian noise with covariance $\sigma_n^2\mathrm{Id}$, and $\boldsymbol{x}_\tau \in \mathbb{R}^n$ denotes the τth video frame.

We focus on video restoration problems that are severely ill-conditioned or ill-posed, leading to significant uncertainty about the solution. We address this difficulty by leveraging prior information

[1] L∀TINO is short for LAtent Video consisTency INverse sOlver.

about $\boldsymbol{x}$ to regularize the estimation problem and deliver meaningful solutions that are well-posed. More precisely, we adopt a Bayesian statistical approach and introduce prior information by specifying the marginal $p(\boldsymbol{x})$, so-called prior distribution, which we then combine with the likelihood function $p(\boldsymbol{y}|\boldsymbol{x}) \propto \exp\{-\|\boldsymbol{y} - \mathcal{A}\boldsymbol{x}\|_2^2/2\sigma_n^2\}$ by using Bayes' theorem to obtain the posterior

$$p(\boldsymbol{x}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})}{\int p(\boldsymbol{y}|\tilde{\boldsymbol{x}})p(\tilde{\boldsymbol{x}})\mathrm{d}\tilde{\boldsymbol{x}}}.$$

We aim to leverage a state-of-the-art generative video model as $p(\boldsymbol{x})$. In recent years, the use of deep generative models as priors in Bayesian frameworks has garnered significant attention, particularly in computational imaging, where denoising diffusion models (DMs) have emerged as powerful generative priors for solving challenging inverse problems (Song & Ermon, 2019; Song et al., 2020; Chung et al., 2022; Kawar et al., 2022; Zhu et al., 2023; Song et al., 2023a; Moufad et al., 2025).

For computational efficiency, modern DMs are often trained in the latent space of a variational autoencoder (VAE), yielding Latent Diffusion Models (LDMs), which are now the backbone of widely used large-scale priors such as Stable Diffusion (Rombach et al., 2021; Podell et al.). More recently, distilled diffusion models, and notably consistency models (CMs) (Song et al., 2023b; Luo et al., 2023a), have emerged as powerful alternatives, producing high-quality samples with only a few neural function evaluations (NFEs), in contrast to the hundreds or thousands often required by iterative DM-based methods. Several recent works have explored leveraging these models in a zero-shot, or so-called Plug & Play (PnP), manner for Bayesian computational imaging (Spagnoletti et al., 2025; Garber & Tirer, 2025; Xu et al., 2024; Li et al., 2025).

Several powerful video DMs (Ho et al., 2022; Blattmann et al., 2023b;a; Chen et al., 2023; Hong et al., 2022) and fast CMs (Wang et al., 2023; Lv et al., 2025; Zhai et al., 2024; Yin et al., 2024b) have recently been proposed, offering great potential for Bayesian video restoration. However, leveraging them remains challenging, so most current methods apply image DMs frame-by-frame and enforce temporal consistency through external constraints (Kwon & Ye, 2025a;b). In challenging settings, this strategy leads to temporal flickering and incoherent dynamics, as it fails to fully capture inter-frame dependencies. This issue could be in principle mitigated by operating directly with video DMs, but applying standard DM-guidance techniques such as DPS to video DMs requires computing gradients by backpropagation through the DM, which incurs a high memory cost (Kwon et al., 2025).

We herein present L∀TINO, the first zero-shot or PnP inverse solver for Bayesian restoration of high definition videos, leveraging priors encoded by video CMs that capture fine spatial-temporal detail and causal dependencies. Moreover, by building on the recent image restoration framework of Spagnoletti et al. (2025), L∀TINO provides a gradient-free inference engine that ensures strong measurement consistency and perceptual quality, while requiring few NFEs and no automatic differentiation.

2 Background

We begin by revisiting the core concepts underlying DMs and LDMs, and briefly discuss their recent extension to generative modeling for video data, which we will use as priors in L∀TINO.

Diffusion Models. (DMs) are generative models that draw samples from a distribution of interest $\pi_0(\boldsymbol{x})$ by iteratively reversing a "noising" process, which is designed to transport $\pi_0(\boldsymbol{x})$ to a standard normal distribution (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2020; Song & Ermon, 2020). In the framework of Ho et al. (2020), the noising and reverse processes are given by the SDEs:

$$d\boldsymbol{x}_t = -\frac{\beta_t}{2}\boldsymbol{x}_t dt + \sqrt{\beta_t} d\boldsymbol{w}_t, \tag{1}$$

$$d\boldsymbol{x}_t = \left[-\frac{\beta_t}{2}\boldsymbol{x}_t - \beta_t \nabla_{\boldsymbol{x}_t} \log \pi_t(\boldsymbol{x}_t)\right] dt + \sqrt{\beta_t} d\overline{\boldsymbol{w}_t}\,, \tag{2}$$

where β_t is the noise schedule, and the score function $\nabla_{\boldsymbol{x}_t} \log \pi_t(\boldsymbol{x}_t)$, which encodes the target π_0, is represented by a network trained by denoising score matching on samples from π_0 (Vincent, 2011). For computational efficiency, modern DMs rely heavily on a (deterministic) probability flow representation of the backward process (2), given by the following ODE (Song et al., 2020):

$$d\boldsymbol{x}_t = \left[-\frac{\beta_t}{2}\boldsymbol{x}_t - \frac{\beta_t}{2} \nabla_{\boldsymbol{x}_t} \log \pi_t(\boldsymbol{x}_t)\right] dt\,. \tag{3}$$

Latent Diffusion Models. LDMs dramatically increase the computational efficiency of DMs by operating in the low-dimensional latent space of an autoencoder $(\mathcal{E}, \mathcal{D})$, rather than directly in pixel space (Rombach et al., 2021). This substantially reduces compute and memory costs, enabling models like Stable Diffusion (SD) to generate large images and video (Podell et al.; Wang et al., 2025).

Video Diffusion Models. Extending DMs to video is an active area of research, requiring models to capture temporal coherence and causality. Below, we highlight some key contributions to this field:

Ho et al. (2022) introduce a spatiotemporal U-Net-based DM tailored for video generation. Their architecture applies 3D convolutions to jointly process space and time, integrates spatial attention blocks for fine-grained detail, as well as temporal attention layers to capture inter-frame dependencies.

Blattmann et al. (2023b;a) propose to repurpose pre-trained LDMs to video through the incorporation of trainable temporal layers l_i^{ϕ} into a frozen U-Net backbone. The temporal layers reshape input batches into a temporally coherent sequence of frames by using a temporal self-attention mechanism.

Wang et al. (2025) introduce a state-of-the-art video foundation model built on three components: (i) *Wan-VAE*, a lightweight 3D causal variational autoencoder, inspired by Wu et al. (2024), that compresses a video $\boldsymbol{x} \in \mathbb{R}^{(1+T)\times H\times W\times 3}$ into a latent tensor $\boldsymbol{z} \in \mathbb{R}^{(1+T/4)\times H/8\times W/8\times C}$ while ensuring temporal causality; (ii) a *Diffusion Transformer (DiT)* Peebles & Xie (2022) that applies patchification, self-attention, and cross-attention to model spatio-temporal context and text conditioning; and (iii) a *text encoder* (umT5) Chung et al. (2023) for semantic conditioning. This architecture enables efficient training and scalable generation of high-resolution, temporally coherent videos.

Consistency Models. Consistency Models (CMs) are single-step DM samplers derived from the probability-flow ODE (3). They rely on a so-called *consistency function* $f : (\boldsymbol{x}_t, t) \mapsto \boldsymbol{x}_\eta$ that maps any state $\boldsymbol{x}_t$ on a trajectory $\{\boldsymbol{x}_t\}_{t\in[\eta,K]}$ of (3) backwards to $\boldsymbol{x}_\eta$, for some small $\eta > 0$, ensuring $f(\boldsymbol{x}_t, t) = f(\boldsymbol{x}_{t'}, t')$ for all $t, t' \in [\eta, K]$. Two-step CMs achieve superior quality by re-noising $\boldsymbol{x}_\eta = f(\boldsymbol{x}_t, t)$ following (1) for some intermediate time $s \in (\eta, K)$, followed by $f(\boldsymbol{x}_s, s)$ to bring back $\boldsymbol{x}_s$ close to the target π_0. Multi-step CMs apply this strategy recursively in 4 to 8 steps, combining top performance with computational efficiency (Song et al., 2023b; Kim et al., 2024).

Latent Consistency Models. CMs can also be trained in latent space by distilling a pre-trained LDM into a latent CM (LCM) (Luo et al., 2023a;b). A particularly effective distillation strategy is *Distribution Matching Distillation* (DMD) (Yin et al., 2023), which trains a generator G_θ to match the diffused data distribution by minimizing a KL divergence over timesteps, using a frozen teacher DM as reference. Its improved version, DMD2 (Yin et al., 2024a), adds a GAN-based loss to further enhance fidelity, and enables few-step samplers (e.g., 4 steps) by conditioning G_θ on discrete timesteps t_i. In practice, G_θ is often initialized from a pre-trained SDXL model (Podell et al.). We use DMD2 (Yin et al., 2024a) within our video prior, as prior distribution on individual video frames.

Video Consistency Models. Recent advancements have extended CMs to video generation. Wang et al. (2023) propose VideoLCM, the first LCM framework for videos, derived by distilling a pre-trained text-to-video DM; it can generate temporally coherent videos in as few as four steps. Yin et al. (2024b) present a theoretical and practical framework to convert slow bidirectional DMs into fast auto-regressive video generators. This conversion enables frame-by-frame causal sampling, allowing generation of very long, temporally consistent videos. Our proposed L∀TINO method incorporates the CM variant of Wan (Wang et al., 2025), distilled via DMD (Yin et al., 2023), into our video prior to effectively capture subtle spatial-temporal dependencies and long-range temporal causality.

Zero-shot (plug & play) posteror sampling. Zero-shot methods leverage a prior model $p(\boldsymbol{x})$ (implicit in a pretrained denoiser or generative model) and the known degradation $p(\boldsymbol{y}|\boldsymbol{x})$ to obtain an estimate of the posterior distribution $p(\boldsymbol{x}|\boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})$. Whereas early zero-shot literature concentrates in maximum a posteriori point estimators (Venkatakrishnan et al., 2013; Monod et al., 2022), we concentrate here on producing samples from the posterior $p(\boldsymbol{x}|\boldsymbol{y})$. This has been addressed by combining prior and likelihood information in various ways, like the split Gibbs sampler (Vono et al., 2019), a discretization of the Langevin SDE (Laumont et al., 2022), a guided diffusion model (Chung et al., 2022; Zhu et al., 2023; Song et al., 2023a; Kwon & Ye, 2025a;b; Kwon et al., 2025) or a guided consistency model (Spagnoletti et al., 2025; Garber & Tirer, 2025; Xu et al., 2024; Li et al., 2025), which is the approach we pursue in this work.

LATINO (Spagnoletti et al., 2025) constructs a Markov chain approximating a Langevin diffusion $\boldsymbol{x}_s$ targeting $p(\boldsymbol{x}|\boldsymbol{y})$ by using the following splitting scheme:

$$\boldsymbol{u} = \boldsymbol{x}_k + \int_0^{\delta_k} \nabla \log p(\tilde{\boldsymbol{x}}_s)\,\mathrm{d}s + \sqrt{2}\,\mathrm{d}\boldsymbol{w}_s, \quad \tilde{\boldsymbol{x}}_0 = \boldsymbol{x}_k\,, \tag{4}$$

$$\boldsymbol{x}_{k+1} = \boldsymbol{u} + \delta_k \nabla \log p(\boldsymbol{y}|\boldsymbol{x}_{k+1})\,, \tag{5}$$

with step-size δ_k. Note that the first step corresponds to an overdamped Langevin diffusion targeting the prior $p(\boldsymbol{x})$, while the second step incorporates the likelihood via an implicit Euler step.

In order to embed an LCM $(\mathcal{E}, \mathcal{D}, f_\theta)$ as prior $p(\boldsymbol{x})$, LATINO replaces (4), which is intractable, with a stochastic auto-encoder (SAE) step that applies the forward and reverse transports (1)-(3) as follows

$$\begin{aligned} \boldsymbol{z} &= \sqrt{\alpha_{t_k}}\mathcal{E}\big(\boldsymbol{x}_k\big) + \sqrt{1-\alpha_{t_k}}\boldsymbol{\epsilon}\,, \\ \boldsymbol{u} &= \mathcal{D}\big(f_\theta(\boldsymbol{z}, t_k)\big)\,, \\ \boldsymbol{x}_{k+1} &= \boldsymbol{u} + \delta_k \nabla \log p(\boldsymbol{y}|\boldsymbol{x}_{k+1})\,, \end{aligned}$$

where we note that the SAE step preserves three fundamental properties of (4): *(i)* contraction of random iterates $\boldsymbol{x}_k$ towards the prior $p(\boldsymbol{x})$; *(ii)* $p(\boldsymbol{x})$ is the unique invariant distribution; and *(iii)* the amount of contraction is controlled via t_k, which plays a role analogous to the integration step-size δ_k. As demonstrated in (Spagnoletti et al., 2025), LATINO exhibits high computational efficiency, requiring only a few NFEs. By leveraging a state-of-the-art SDXL LCM (Yin et al., 2024a), it achieves remarkable accuracy and perceptual quality across a range of challenging imaging tasks.

3 L∀TINO FOR HIGH DEFINITION VIDEO POSTERIOR SAMPLING

We are now ready to present our proposed LAtent Video consisTency INverse sOlver (L∀TINO), which approximately draws samples from the posterior distribution

$$p(\boldsymbol{x}|\boldsymbol{y}, c, \lambda) = \frac{p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x}|c,\lambda)}{\int_{\mathbb{R}^n} p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x}|c,\lambda)\mathrm{d}\boldsymbol{x}}\,,$$

parametrized by the data $\boldsymbol{y}$, a text prompt c, and a spatiotemporal regularization parameter $\lambda \in \mathbb{R}_+^3$. As mentioned previously, L∀TINO is a zero-shot Langevin posterior sampler specialised for video restoration, which jointly leverages prior information from both Video Consistency Models (VCMs) and Image Consistency Models (ICMs). In addition, L∀TINO is highly computationally efficient, requiring only a small number of NFEs and operating in a gradient-free manner, which significantly reduces memory usage and enables scalability to long video sequences.

A main novelty in L∀TINO is the use of the following product-of-experts prior for video restoration

$$p(\boldsymbol{x}|c,\lambda) \propto p_V^{\eta}(\boldsymbol{x}|c)p_I^{1-\eta}(\boldsymbol{x}|c)p_\phi(\boldsymbol{x}|\lambda)\,,$$

where $\eta \in (0,1)$ is a temperature parameter and $p_V(\boldsymbol{x}|c)$, $p_I(\boldsymbol{x}|c)$, and $p_\phi(\boldsymbol{x}|\lambda)$ are as follows:

- $p_V(\boldsymbol{x}|c)$ is implicitly defined via a text-to-video LCM designed to capture subtle spatial-temporal dependencies as well as long-range temporal causality. It is specified by an encoder-decoder pair $(\mathcal{E}_V, \mathcal{D}_V)$ and consistency function f_ϑ^V operating in their latent space.
- $p_I(\boldsymbol{x}|c)$ is implicitly defined via a high-resolution text-to-image LCM, acting separately on each frame, to recover fine spatial detail and enhance perceptual quality. It is specified by an encoder-decoder pair $(\mathcal{E}_I, \mathcal{D}_I)$ and consistency function f_θ^I operating in their latent space.
- $p_\phi(\boldsymbol{x}|\lambda) \propto \exp\{-\phi_\lambda(\boldsymbol{x})\}$ where ϕ_λ is a convex regularizer promoting background stability and smooth temporal transitions across frames, with $\lambda \in \mathbb{R}_+^3$ controlling the regularity enforced. Without loss of generality, in our experiments we use the total-variation norm

$$\phi_\lambda(\boldsymbol{x}) = \mathrm{TV}_3^\lambda(\boldsymbol{x}) \triangleq \sum_{\tau,c,i,j} \sqrt{\lambda_h^2\big(D_h\boldsymbol{x}_{\tau,c,i,j}\big)^2 + \lambda_v^2\big(D_v\boldsymbol{x}_{\tau,c,i,j}\big)^2 + \lambda_t^2\big(D_t\boldsymbol{x}_{\tau,c,i,j}\big)^2}\,.$$

where (D_h, D_v, D_t) is the three-dimensional discrete gradient. Note that TV_3^λ is not smooth.

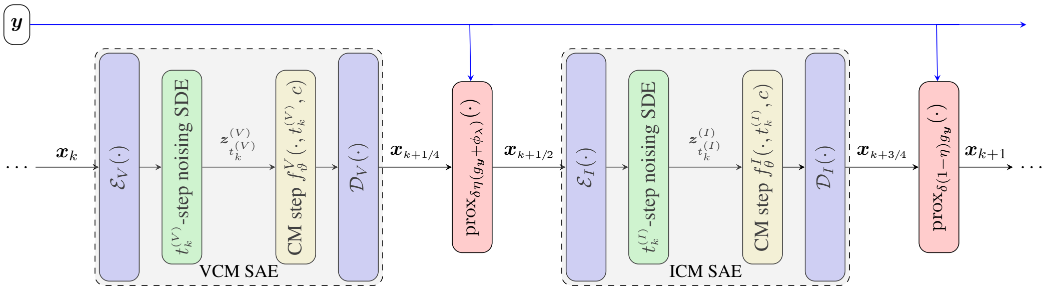

$\boldsymbol{y}$
$\cdots$
$\boldsymbol{x}_k$
$\mathcal{E}_V(\cdot)$
$t_k^{(V)}$-step noising SDE
$\boldsymbol{z}^{(V)}_{t_k^{(V)}}$
CM step $f^V_\vartheta(\cdot, t_k^{(V)}, c)$
$\mathcal{D}_V(\cdot)$
VCM SAE
$\boldsymbol{x}_{k+1/4}$
$\text{prox}_{\delta\eta(g_{\boldsymbol{y}}+\phi_\lambda)}(\cdot)$
$\boldsymbol{x}_{k+1/2}$
$\mathcal{E}_I(\cdot)$
$t_k^{(I)}$-step noising SDE
$\boldsymbol{z}^{(I)}_{t_k^{(I)}}$
CM step $f^I_\theta(\cdot, t_k^{(I)}, c)$
$\mathcal{D}_I(\cdot)$
ICM SAE
$\boldsymbol{x}_{k+3/4}$
$\text{prox}_{\delta(1-\eta)g_{\boldsymbol{y}}}(\cdot)$
$\boldsymbol{x}_{k+1}$
$\cdots$

Figure 2: One step of the L∀TINO solver, a discretization of the Langevin SDE (7) which targets the posterior $p(\boldsymbol{x}|\boldsymbol{y}, c, \lambda)$, involving two stochastic autoencoding (SAE) steps and two proximal steps.

Following a PnP philosophy, $p(\boldsymbol{x}|\boldsymbol{y}, c, \lambda)$ combines an analytical likelihood function $p(\boldsymbol{y}|\boldsymbol{x})$ with a prior distribution $p(\boldsymbol{x}|c, \lambda)$ that is represented implicitly by a pre-trained machine learning model. However, unlike conventional PnP approaches that exploit a denoising operator (e.g., PnP Langevin (Laumont et al., 2022)), L∀TINO leverages the LATINO framework of Spagnoletti et al. (2025) which is specialised for embedding generative models as priors, notably distilled foundation CMs.

To draw samples from $p(\boldsymbol{x}|\boldsymbol{y}, c, \lambda)$, L∀TINO considers a Moreau-Yosida regularized overdamped Langevin diffusion, given by the SDE

$$\begin{aligned} \mathrm{d}\boldsymbol{x}_s =& \nabla \log p(\boldsymbol{y}|\boldsymbol{x}_s)\mathrm{d}s + \nabla \log p_V^{\eta}(\boldsymbol{x}_s|c)\mathrm{d}s + \nabla \log p_I^{(1-\eta)}(\boldsymbol{x}_s|c)\mathrm{d}s \\ &+ \nabla \log \tilde{p}_{\gamma\phi}(\boldsymbol{x}_s|\lambda)\mathrm{d}s + \sqrt{2}\mathrm{d}\boldsymbol{w}_s\,, \end{aligned} \tag{6}$$

where $\boldsymbol{w}_s$ denotes a n-dimensional Brownian motion and $\tilde{p}_{\gamma\phi}(\boldsymbol{x}_s|\lambda)$ is the γ-Moreau-Yosida approximation of the non-smooth factor $p_\phi(\boldsymbol{x}_s|\lambda)$, given by (Pereyra, 2016)

$$\tilde{p}_{\gamma\phi}(\boldsymbol{x}|\lambda) \propto \sup_{\boldsymbol{u}\in\mathbb{R}^n} p_\phi(\boldsymbol{u}|\lambda) \exp\{-\frac{1}{2\gamma}\|\boldsymbol{x} - \boldsymbol{u}\|_2^2\}\,,$$

with $\gamma > 0$. As mentioned previously, $\tilde{p}_{\gamma\phi}(\boldsymbol{x}|\lambda)$ is log-concave and Lipchitz differentiable by construction because ϕ_λ is convex on $\mathbb{R}^n$ (Pereyra, 2016). The likelihood $p(\boldsymbol{y}|\boldsymbol{x}) \propto \exp\{-\|\boldsymbol{y} - \mathcal{A}\boldsymbol{x}\|_2^2/2\sigma_n^2\}$ is also log-concave and Lipchitz differentiable.

Under mild regularity assumptions on $p_V(\boldsymbol{x}|c)$ and $p_I(\boldsymbol{x}|c)$, starting from an initial condition $\boldsymbol{x}_0$, the process $\boldsymbol{x}_s$ converges to a γ-neighborhood of $p(\boldsymbol{x}|\boldsymbol{y}, c, \lambda)$ exponentially fast as $s \to \infty$ (Laumont et al., 2022). While solving (6) exactly is not possible, considering numerical approximations of $\boldsymbol{x}_s$ provides a powerful computational framework for deriving approximate samplers for $p(\boldsymbol{x}|\boldsymbol{y}, c)$.

L∀TINO stems from approximating (6) by a Markov chain derived from the following recursion: given an initialization $\boldsymbol{x}_0$ and a step-size $\delta > 0$, for all $k \geq 0$,

$$\begin{aligned} &\underbrace{\boldsymbol{x}_{k+1/4} = \boldsymbol{x}_k + \int_0^\delta \eta\nabla \log p_V(\tilde{\boldsymbol{x}}_s|c)\mathrm{d}s + \sqrt{2\eta}\,\mathrm{d}\boldsymbol{w}_s, \quad \tilde{\boldsymbol{x}}_0 = \boldsymbol{x}_k}_{\text{VCM prior step}} \\ &\underbrace{\boldsymbol{x}_{k+1/2} = \boldsymbol{x}_{k+1/4} + \eta\delta\nabla \log p\left(\boldsymbol{y}|\boldsymbol{x}_{k+1/2}\right) + \eta\delta\nabla \log \tilde{p}_{\gamma\phi}\left(\boldsymbol{x}_{k+1/2}|\lambda\right)}_{\text{implicit likelihood half-step with } \phi\text{-regularization}} \\ &\underbrace{\boldsymbol{x}_{k+3/4} = \boldsymbol{x}_{k+1/2} + \int_0^\delta (1-\eta)\nabla \log p_I(\tilde{\boldsymbol{x}}_s|c)\mathrm{d}s + \sqrt{2(1-\eta)}\,\mathrm{d}\boldsymbol{w}_s, \quad \tilde{\boldsymbol{x}}_0 = \boldsymbol{x}_{k+1/2}}_{\text{ICM prior step}} \\ &\underbrace{\boldsymbol{x}_{k+1} = \boldsymbol{x}_{k+3/4} + (1-\eta)\delta\nabla \log p(\boldsymbol{y}|\boldsymbol{x}_{k+1})}_{\text{implicit likelihood half-step}}\,, \end{aligned} \tag{7}$$

where we identify a splitting in which each CM prior is involved separately through exact integration (these integrals will be approximated through SAE steps), and the likelihood is involved through two implicit (backward Euler) half-steps. Importantly, unlike the explicit steps used in most Langevin sampling algorithms, the implicit steps in (7) remain numerically stable for all $\delta > 0$. This allows L∀TINO to converge quickly by taking δ large, albeit with some small bias. Conversely, the widely used unadjusted Langevin algorithm (ULA) integrates only the Brownian term $\boldsymbol{w}_s$ exactly, it involves gradients via an explicit Euler step, and is explosive unless δ is sufficiently small. It is worth recalling that the Langevin diffusion is a time-homogeneous process. The iterates $\boldsymbol{x}_k$ resulting from its discrete-time approximation are asymptotically ergodic, converging to a neighborhood of $p(\boldsymbol{x}|\boldsymbol{y}, c, \lambda)$ as $k \to \infty$. Unlike DMs, these iterates do not travel backwards in time through an inhomogeneous process. Therefore, Langevin algorithms use directly the likelihood $p(\boldsymbol{y}|\boldsymbol{x}) \propto \exp\{-\|\boldsymbol{y} - \mathcal{A}\boldsymbol{x}\|_2^2/2\sigma_n^2\}$, avoiding the need to approximate the likelihood of $\boldsymbol{y}$ w.r.t. a noisy version of $\boldsymbol{x}$, as required in guided DMs like (Chung et al., 2022; Song et al., 2023a; Kwon et al., 2025).

Following Spagnoletti et al. (2025), we compute $\boldsymbol{x}_{k+1/4}$ and $\boldsymbol{x}_{k+3/4}$ approximately via SAE steps,

$$\boldsymbol{x}_{k+1/4} = \mathcal{D}^V \left(f_\vartheta^V \left(\sqrt{\alpha_{t_k^{(V)}}} \mathcal{E}_V \left(\boldsymbol{x}^{(k)} \right) + \sqrt{1 - \alpha_{t_k^{(V)}}} \boldsymbol{\epsilon}, t_k^{(V)} \right), c \right) ,$$

$$\boldsymbol{x}_{k+3/4} = \mathcal{D}^I \left(f_\theta^I \left(\sqrt{\alpha_{t_k^{(I)}}} \mathcal{E}_I \left(\boldsymbol{x}^{(k)} \right) + \sqrt{1 - \alpha_{t_k^{(I)}}} \boldsymbol{\epsilon}, t_k^{(I)}, c \right) \right) ,$$

where we recall that $(\mathcal{E}^I, \mathcal{D}^I, f^I)$ act frame-wise and that f_ϑ^V and f_θ^I have model-specific schedules.

The implicit Euler steps in (7) can be reformulated as an explicit proximal point steps as follows

$$\begin{aligned}
\tilde{\boldsymbol{x}}_{k+1/2} &= \underset{\boldsymbol{u} \in \mathbb{R}^n}{\arg\min}\, g_{\boldsymbol{y}}(\boldsymbol{u}) + \left(\inf_{\boldsymbol{u}' \in \mathbb{R}^n} \phi_\lambda(\boldsymbol{u}') + \tfrac{1}{2\gamma} \|\boldsymbol{u} - \boldsymbol{u}'\|_2^2 \right) + \tfrac{1}{2\delta\eta} \|\tilde{\boldsymbol{x}}_{k+1/4} - \boldsymbol{u}\|_2^2 \,, \\
&\approx \underset{\boldsymbol{u} \in \mathbb{R}^n}{\arg\min}\, g_{\boldsymbol{y}}(\boldsymbol{u}) + \phi_\lambda(\boldsymbol{u}) + \tfrac{1}{2\delta\eta} \|\tilde{\boldsymbol{x}}_{k+1/4} - \boldsymbol{u}\|_2^2 \,, \\
\tilde{\boldsymbol{x}}_{k+1} &= \underset{\boldsymbol{u} \in \mathbb{R}^n}{\arg\min}\, g_{\boldsymbol{y}}(\boldsymbol{u}) + \tfrac{1}{2\delta(1-\eta)} \|\tilde{\boldsymbol{x}}_{k+3/4} - \boldsymbol{u}\|_2^2 \,,
\end{aligned}$$

where $g_{\boldsymbol{y}} : \boldsymbol{x} \mapsto -\log p(\boldsymbol{y}|\boldsymbol{x})$ and where we have simplified the computation of $\tilde{\boldsymbol{x}}_{k+1/2}$ by assuming that $\gamma \ll \delta\eta$ (Pereyra, 2016). The optimization problems described above are strongly convex and can be efficiently approximated by using a small number of iterations of a specialized solver. In particular, to compute $\tilde{\boldsymbol{x}}_{k+1}$, we employ a few iterations of the conjugate gradient algorithm with warm-starting (Hestenes & Stiefel, 1952). For the computation of $\tilde{\boldsymbol{x}}_{k+1/2}$, we recommend using a proximal splitting optimizer (Chambolle & Pock, 2011), or a warm-started Adam optimizer (Kingma & Ba, 2014), both of which are effective in practice. Please see Appendix A.6 for more details.

Refer to Algorithm 1 for more details about L∀TINO, and to Figure 2 for its schematic representation.

4 EXPERIMENTS

Models. We implement L∀TINO by using `CausVid` as VCM prior. We adopt the standard bidirectional WaN architecture, fine-tuned as a CM. The model also supports an autoregressive configuration, which we do not utilize here, leaving the exploration of autoregressive priors for longer video restoration to future work. Concerning the ICM, we use `DMD2`, following Spagnoletti et al. (2025). For our experiments, we use $t_i^{(V)} \in \{757, 522, 375, 255, 125\}$ and $t_i^{(I)} \in \{374, 249, 124, 63\}$ for the VCM and ICM respectively. This results in a total of 9 NFEs, where applying the ICM across all frames counts as a single NFE. Regarding the text prompt specifying VCM and ICM, in the same spirit as Kwon & Ye (2025b), we do not perform any prompt optimization and instead use the generic prompt *"A high resolution video/image"*. Exploring prompt optimization by leveraging the maximum likelihood strategy of Spagnoletti et al. (2025) remains a key direction for future work.

Dataset and Metrics. We evaluate methods on 435 video clips of 25 frames each from the `Adobe240` dataset (Su et al., 2017), and 239 video clips of 25 frames each from the `GoPRO240` test dataset Nah et al. (2016). These datasets contain high-quality, high-frame-rate video sequences that we rescale to a spatial resolution of 1280 × 768 pixels to match our targeted resolution.

We assess reconstruction quality using peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) (Wang et al., 2004). Additionally, we evaluate two perceptual metrics: Learned Perceptual Image Patch Similarity (LPIPS) (Zhang et al., 2018), along with the recently proposed

Algorithm 1 LVTINO (LAtent Video consisTency INverse sOlver)

```
given degraded video $\boldsymbol{y}$, operator $\mathcal{A}$, initialization $\boldsymbol{x}_0 = \mathcal{A}^\dagger \boldsymbol{y}$, video lenght $T+1$, steps $N = 5$
given video CM $(\mathcal{E}_V, \mathcal{D}_V, f^V_\vartheta)$, image CM $(\mathcal{E}_I, \mathcal{D}_I, f^I_\theta)$, schedules $\{t^{(V)}_k, t^{(I)}_k, \delta_k, \eta, \lambda\}_{k=0}^{N-1}$, $g_{\boldsymbol{y}}$
for $k = 0, \ldots, N-1$ do
    # VCM prior half-step (temporal coherence)
    $\boldsymbol{\epsilon}_V \sim \mathcal{N}\big(0, \mathrm{Id}_{(1+T/4)\times H/8 \times W/8 \times C}\big)$
    $\boldsymbol{z}^{(V)}_{t^{(V)}_k} \leftarrow \sqrt{\alpha_{t^{(V)}_k}}\, \mathcal{E}_V\big(\boldsymbol{x}_{k-1}\big) \;+\; \sqrt{1-\alpha_{t^{(V)}_k}}\, \boldsymbol{\epsilon}_V$
    $\tilde{\boldsymbol{x}}_{k+1/4} \leftarrow \mathcal{D}_V\big(f^V_\vartheta\big(\boldsymbol{z}^{(V)}_{t^{(V)}_k},\, t^{(V)}_k\big)\big)$        ▷ VCM
    # First likelihood - Solved with proximal splitting or Adam iterations
    $\tilde{\boldsymbol{x}}_{k+1/2} \leftarrow \arg\min_{\boldsymbol{u}\in\mathbb{R}^{(T+1)\times H\times W\times 3}} g_y(\boldsymbol{u}) + \phi_\lambda(\boldsymbol{u}) + \frac{1}{2\delta_k \eta}\|\tilde{\boldsymbol{x}}_{k+1/4} - \boldsymbol{u}\|_2^2$
    if $k < N$ then
        # ICM prior half-step (per-frame detail)
        $\boldsymbol{\epsilon}_I \sim \mathcal{N}\big(0, \mathrm{Id}_{h/8\times w/8\times c}\big)$
        $\tilde{\boldsymbol{x}}_{k+3/4} \leftarrow \mathrm{stack}_{\tau=0}^{T}\, \mathcal{D}_I\Big(f^I_\theta\big(\sqrt{\alpha_{t^{(I)}_k}}\, \mathcal{E}_I(\tilde{\boldsymbol{x}}_{k+1/2,\tau}) + \sqrt{1-\alpha_{t^{(I)}_k}}\, \boldsymbol{\epsilon}_I,\; t^{(I)}_k\big)\Big)$        ▷ ICM
        # Likelihood prox (2nd) - Solved with conjugate gradient iterations
        $\boldsymbol{x}_k \leftarrow \arg\min_{\boldsymbol{u}\in\mathbb{R}^{(T+1)\times H\times W\times 3}} g_y(\boldsymbol{u}) + \frac{1}{2\delta_k(1-\eta)}\|\tilde{\boldsymbol{x}}_{k+3/4} - \boldsymbol{u}\|_2^2$
    else
        # Final iteration: skip ICM and second likelihood
        $\boldsymbol{x}_k \leftarrow \tilde{\boldsymbol{x}}_{k+1/2}$
    end if
end for
return $\boldsymbol{x}_N$
```

Fréchet Video Motion Distance (FVMD) (Liu et al., 2024) which is tailored for assessing motion smoothness and perceptual quality in videos.

Inverse Problems. We consider three linear inverse problems for high-resolution video restoration. Let $\boldsymbol{x} = (\boldsymbol{x}_\tau)_{\tau=0}^T \in \mathbb{R}^{(T+1)\times H\times W\times C}$ denote the unknown high-resolution video and $\boldsymbol{y} = \mathcal{A}\boldsymbol{x} + \boldsymbol{n}$ the observed degraded video with additive Gaussian noise $\boldsymbol{n}$. For fair comparisons, we consider a mild noise regime $\sigma_n = 0.001$, which addresses the noiseless case.

- **Problem A** - *Temporal SR$\times 4$ + SR$\times 4$:* here $\mathcal{A}$ first applies temporal average pooling with factor 4 (reducing the frame rate), followed by frame-wise spatial downsampling by factor 4, simulating a low frame rate and low resolution video. [2] Temporal upsampling to generate the missing frame is highly challenging here, as it requires prior knowledge of motion.
- **Problem B** - *Temporal blur + SR$\times 8$:* here $\mathcal{A}$ first applies a uniform blur kernel of size 7 pixels along the temporal dimension, followed by frame-wise spatial downsampling by a factor 8, simulating a motion-blurred and low-resolution video (Kwon & Ye, 2025a;b).
- **Problem C** - *Temporal SR$\times 8$ + SR$\times 8$:* is a harder version of **Problem A**, where $\mathcal{A}$ first applies temporal average pooling with factor 8 and then a spatial downsampling by factor 8.

	Problem A: Temp. SR×4 + SR×4					Problem B: Temp. blur + SR×8					Problem C: Temp. SR×8 + SR×8				
Method	NFE↓	FVMD↓	PSNR↑	SSIM↑	LPIPS↓	NFE↓	FVMD↓	PSNR↑	SSIM↑	LPIPS↓	NFE↓	FVMD↓	PSNR↑	SSIM↑	LPIPS↓
LVTINO	9	**371.1**	**27.25**	**0.837**	**0.249**	9	**42.65**	24.91	0.741	**0.370**	**7**	602.5	23.11	**0.697**	**0.411**
VISION-XL	**8**	1141	26.03	0.672	0.439	**8**	82.92	**26.18**	**0.749**	0.468	8	1604	**23.38**	0.652	0.520
VIDUE	–	–	–	–	–	–	–	–	–	–	**1**	**142.5**	21.78	0.624	0.505
ADMM-TV	–	427.6	18.04	0.767	0.297	–	128.2	21.18	0.644	0.452	–	1645	18.15	0.663	0.439

Table 1: Results on the Adobe240 dataset across the three problems. Best results are in **bold**, second best are underlined.

[2] Temporal SR×k is also a coarse (Riemann sum) approximation of motion blur due to moving objects or camera during full continuous exposure between frames (Zhang et al., 2021).

	Problem A: Temp. SR×4 + SR×4					Problem B: Temp. blur + SR×8					Problem C: Temp. SR×8 + SR×8				
Method	**NFE↓**	**FVMD↓**	**PSNR↑**	**SSIM↑**	**LPIPS↓**	**NFE↓**	**FVMD↓**	**PSNR↑**	**SSIM↑**	**LPIPS↓**	**NFE↓**	**FVMD↓**	**PSNR↑**	**SSIM↑**	**LPIPS↓**
LVTINO	9	**189.4**	24.01	0.775	**0.315**	9	**46.20**	22.46	0.687	**0.433**	7	232.6	**22.91**	**0.677**	**0.445**
VISION-XL	**8**	282.2	**26.06**	**0.792**	0.326	**8**	52.03	**24.05**	**0.697**	0.486	8	995.9	22.67	0.669	0.474
VIDUE	–	–	–	–	–	–	–	–	–	–	**1**	**84.45**	20.66	0.571	0.548
ADMM-TV	–	265.8	24.32	0.745	0.406	–	145.9	20.83	0.618	0.527	–	969.3	17.70	0.631	0.527

Table 2: Results on the GoPro240 dataset across the three problems. Best results are in **bold**, second best are underlined.

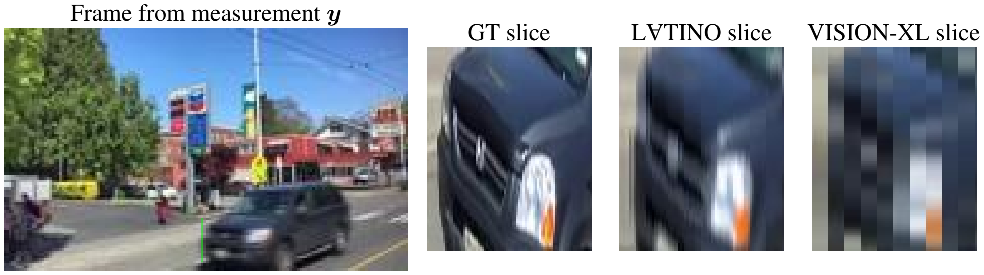


Figure 3: Comparison between slices from 81 consecutive frames for **Problem C (seq. C2)**. Slice images (i, τ) are obtained from the video tensor (i, j, τ) by fixing a column index j shown in green.

Computational Efficiency. While NFEs provide a hardware-agnostic measure of complexity, practical deployment requires considering runtime and memory footprints. Table 4 reports the wall-clock time and peak GPU memory usage for restoring a 25-frame video, measured on one A100 GPU. VISION-XL, by only loading an image model, exchanges memory usage for time, as it needs to perform sequentially each frame. LVTINO offers a competitive trade-off thanks to the VCM, which scales better for longer videos. Notably, the lighter variant LVTINO-V (see Appendix A.6 for more details) achieves the fastest runtime among deep generative approaches with a moderate memory cost, as it only loads the VCM component.

Method	**NFE ↓**	**Time (s) ↓**	**Mem. (GB) ↓**
LVTINO	9	132	35.15
VISION-XL	8	176	**15.64**
ADMM-TV	–	**13.6**	22.01
LVTINO-V	**5**	105	25.42

Table 3: Runtime and memory usage. Measured on a single video clip of 25 frames at 1280 × 768 resolution. Best results are in **bold**, second best are underlined.

Results. Experiments in Table 1 refer to **Problems A, B** and **C**, and are obtained with different numerical schemes for (7). We fix the hyperparameters per problem to better tackle the different degradations; see Table 4 in Appendix A.6 for more details and for an ablation study.

For the more challenging **Problem C**, to stabilize and warm-start LVTINO, we use the joint deblurring/interpolation network of Shang et al. (2023)[3] to produce a temporally interpolated version of $\boldsymbol{y}$, which we then upsample via bilinear spatial interpolation so that it can be used as initialization $\boldsymbol{x}_0$. This warm-start allows us to reduce the number of integration steps, bringing the NFEs to 7. The same model, referred to as VIDUE, is used as a baseline comparison in Table 1 and Table 2.

We further provide a visual analysis of motion quality using fixed vertical slices of video frames, following Cohen et al. (2024), who observed that spatiotemporal slices of natural videos resemble natural images. Figure 3 and Appendix B in Figures 11a and 11b show (i, τ) slices. These reveal that even for small motions, LVTINO more closely preserves ground truth temporal continuity.

Qualitative and quantitative evaluation. Figures 1, 4, 5, and 6 show the results of our algorithm compared to the measurements, ground truth and VISION-XL (see also the videos by following the links in the captions). Table 8 in Appendix B provides additional results. These results demonstrate that LVTINO yields more detailed and temporally coherent videos than VISION-XL. The ICM prior enhances spatial detail, while the VCM prior and TV_3^λ jointly improve temporal coherence, particularly in the challenging upsampling tasks B and C. For example, in Figure 6, LVTINO achieves noticeably sharper results with minimal motion blur and strong temporal coherence, whereas VISION-

[3] Which is trained on the GoPRO240 train dataset (Nah et al., 2016).

XL shows a staircase effect with repeated frames and unresolved blur, also evident in Figure 4. In Figure 5, VISION-XL exhibits temporal flickering, which our method eliminates via the VCM and TV models. Table 1 supports these visual findings: L∀TINO achieves strong FVMD and LPIPS scores, reflecting accurate spatiotemporal dynamics and fine spatial detail.

Other baselines. We also report comparisons with ADMM-TV, a classical optimization-based method (we use the hyperparameters of (Kwon & Ye, 2025a)). We also considered comparing with VDPS (Kwon et al., 2025), however the backpropagation through Wan's DiT and Decoder at resolution 1280 × 768 pixels required > 80 Gb of VRAM, exceeding the memory capacity of GPUs available in our academic HPC facility. Since L∀TINO's conditioning mechanism does not rely on automatic differentiation, it has significantly lower memory usage.

5 Conclusion

We introduced L∀TINO, the first VCM-based zero-shot or PnP inverse solver for Bayesian restoration of high definition videos. By combining a VCM, a frame-wise ICM and TV3 regularization, L∀TINO can recover subtle spatial temporal dynamics, as evidenced by its strong performance on challenging tasks and datasets involving both moving objects and camera shake. Moreover, L∀TINO's conditioning mechanism ensures strong measurement consistency and perceptual quality, while requiring as few as 8 NFEs and no automatic differentiation. We anticipate that upcoming advancements in distillation of VCMs will further improve the accuracy and computational efficiency of L∀TINO.

Future research will explore sequential and auto-regressive Bayesian strategies for the restoration of long videos, as well as better Langevin sampling scheme through the use of more sophisticated numerical integrators. Another promising research direction is the incorporation of automatic prompt optimization by maximum likelihood estimation, as considered in Spagnoletti et al. (2025) for image restoration tasks. Furthermore, it would be interesting to specialize L∀TINO for particular tasks through the unfolding and distillation framework of Kemajou Mbakam et al. (2025).

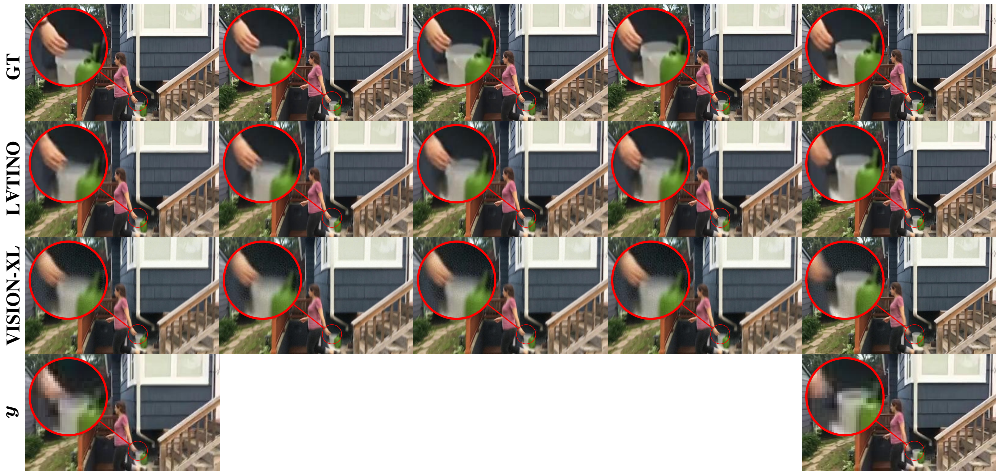


Figure 4: Visual comparison for **Problem A (seq. A1)**. The continuity of the motion is retrieved as the hand moves from right to left. See full videos: **L∀TINO** and **VISION-XL**.

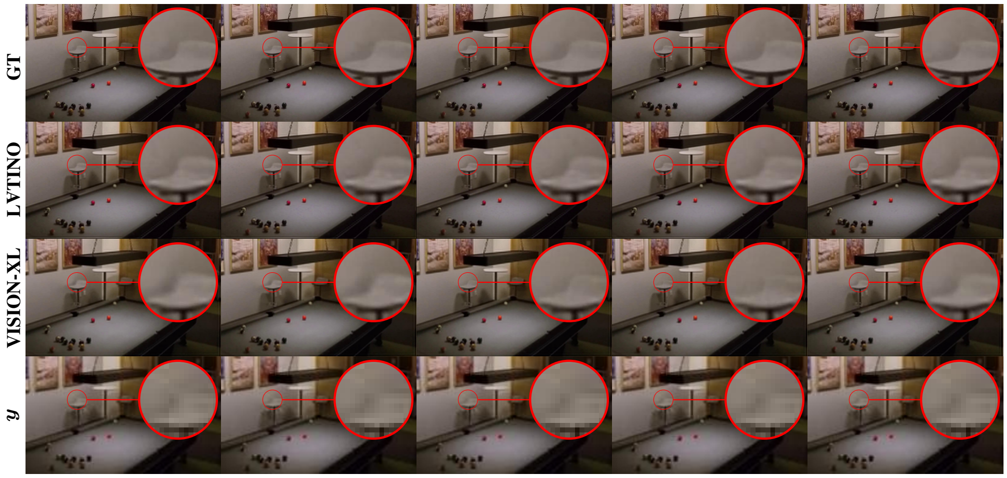


Figure 5: Visual comparison for **Problem B (seq. B2)**. The flickering problem is solved by L∀TINO (see darker and lighter area behind the chair). See full videos: **L∀TINO** and **VISION-XL**.

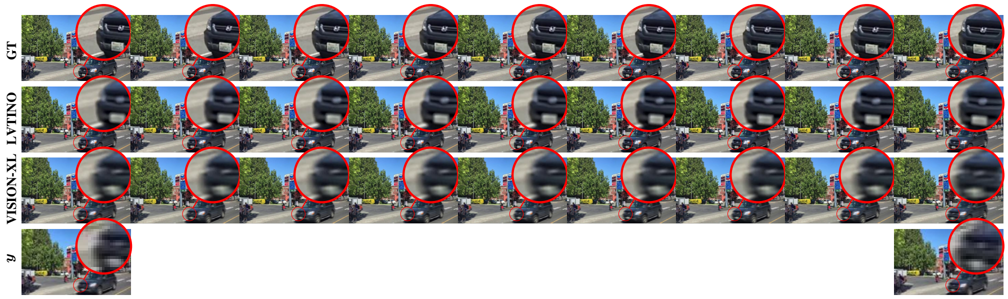


Figure 6: Visual comparison for **Problem C (seq. C2)**. The motion is retrieved by the reconstruction. See full videos (81 frames for a better direct comparison): **L∀TINO** and **VISION-XL**.

Reproducibility statement

To allow complete reproducibility, we commit to publishing the full code on GitHub upon acceptance. Furthermore, the L∀TINO algorithm is fully described as pseudo-code in Algorithm 1 and the details contained in Table 4 and Sections 4, A.1, A.2 describe the implementations of the key components.

References

Joe Benton, George Deligiannidis, and Arnaud Doucet. Error bounds for flow matching methods, 2024. URL https://arxiv.org/abs/2305.16860.

A. Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, and Dominik Lorenz. Stable video diffusion: Scaling latent video diffusion models to large datasets. *ArXiv*, abs/2311.15127, 2023a. URL https://api.semanticscholar.org/CorpusID:265312551.

A. Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22563–22575, 2023b. URL https://api.semanticscholar.org/CorpusID:258187553.

A. Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40:120–145, 2011. URL https://api.semanticscholar.org/CorpusID:261281173.

Haoxin Chen, Menghan Xia, Yin-Yin He, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, Chao-Liang Weng, and Ying Shan. Videocrafter1: Open diffusion models for high-quality video generation. *ArXiv*, abs/2310.19512, 2023. URL https://api.semanticscholar.org/CorpusID:264803867.

Hyung Won Chung, Noah Constant, Xavier García, Adam Roberts, Yi Tay, Sharan Narang, and Orhan Firat. Unimax: Fairer and more effective language sampling for large-scale multilingual pretraining. *ArXiv*, abs/2304.09151, 2023. URL https://api.semanticscholar.org/CorpusID:258187051.

Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*, 2022.

Nathaniel Cohen, Vladimir Kulikov, Matan Kleiner, Inbar Huberman-Spiegelglas, and Tomer Michaeli. Slicedit: Zero-shot video editing with text-to-image diffusion models using spatio-temporal slices. *ArXiv*, abs/2405.12211, 2024. URL https://api.semanticscholar.org/CorpusID:269921890.

Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106:1602 – 1614, 2011. URL https://api.semanticscholar.org/CorpusID:23284154.

Tomer Garber and Tom Tirer. Zero-shot image restoration using few-step guidance of consistency models (and beyond). In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pp. 2398–2407, June 2025.

Magnus R. Hestenes and Eduard Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of research of the National Bureau of Standards*, 49:409–435, 1952. URL https://api.semanticscholar.org/CorpusID:2207234.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. *ArXiv*, abs/2204.03458, 2022. URL https://api.semanticscholar.org/CorpusID:248006185.

Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *ArXiv*, abs/2205.15868, 2022. URL https://api.semanticscholar.org/CorpusID:249209614.

Samuel Hurault, Arthur Leclaire, and Nicolas Papadakis. Gradient step denoiser for convergent plug-and-play. In *International Conference on Learning Representations (ICLR)*, 2022.

Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. *Advances in Neural Information Processing Systems*, 35:23593–23606, 2022.

Charlesquin Kemajou Mbakam, Jonathan Spence, and Marcelo Pereyra. Learning few-step posterior samplers by unfolding and distillation of diffusion models, 2025. URL https://arxiv.org/abs/2507.02686.

Beomsu Kim, Jaemin Kim, Jeongsol Kim, and Jong Chul Ye. Generalized consistency trajectory models for image manipulation. *ArXiv*, abs/2403.12510, 2024. URL https://api.semanticscholar.org/CorpusID:268532278.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *preprint arXiv:1412.6980*, 2014. URL https://api.semanticscholar.org/CorpusID:6628106.

Nikita Kornilov, Alexander Gasnikov, and Alexander Korotin. Optimal flow matching: Learning straight trajectories in just one step. *arXiv preprint arXiv:2403.13117*, 2024.

Taesung Kwon and Jong Chul Ye. Solving video inverse problems using image diffusion models, 2025a. URL https://arxiv.org/abs/2409.02574.

Taesung Kwon and Jong Chul Ye. Vision-xl: High definition video inverse problem solver using latent image diffusion models, 2025b. URL https://arxiv.org/abs/2412.00156.

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*, PP, 2025. URL https://api.semanticscholar.org/CorpusID:280648146.

Rémi Laumont, Valentin De Bortoli, Andrés Almansa, Julie Delon, Alain Durmus, and Marcelo Pereyra. Bayesian imaging using plug & play priors: when langevin meets tweedie. *SIAM Journal on Imaging Sciences*, 15(2):701–737, 2022.

Xiang Li, Soo Min Kwon, Shijun Liang, Ismail R. Alkhouri, Saiprasad Ravishankar, and Qing Qu. Decoupled data consistency with diffusion purification for image restoration, 2025. URL https://arxiv.org/abs/2403.06054.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023. URL https://arxiv.org/abs/2210.02747.

Jiahe Liu, Youran Qu, Qi Yan, Xiaohui Zeng, Lele Wang, and Renjie Liao. Fréchet video motion distance: A metric for evaluating motion consistency in videos. *ArXiv*, abs/2407.16124, 2024. URL https://api.semanticscholar.org/CorpusID:271334698.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow, 2022. URL https://arxiv.org/abs/2209.03003.

Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *ArXiv*, abs/2310.04378, 2023a. URL https://api.semanticscholar.org/CorpusID:263831037.

Simian Luo, Yiqin Tan, Suraj Patil, Daniel Gu, Patrick von Platen, Apolinário Passos, Longbo Huang, Jian Li, and Hang Zhao. Lcm-lora: A universal stable-diffusion acceleration module, 2023b. URL https://arxiv.org/abs/2311.05556.

Zhengyao Lv, Chenyang Si, Tianlin Pan, Zhaoxi Chen, Kwan-Yee K. Wong, Yu Qiao, and Ziwei Liu. Dcm: Dual-expert consistency model for efficient and high-quality video generation. *ArXiv*, abs/2506.03123, 2025. URL https://api.semanticscholar.org/CorpusID:279119323.

Ségolène Martin, Anne Gagneux, Paul Hagemann, and Gabriele Steidl. Pnp-flow: Plug-and-play image restoration with flow matching, 2025. URL https://arxiv.org/abs/2410.02423.

Tim Meinhardt, Michael Moeller, Caner Hazirbas, and Daniel Cremers. Learning proximal operators: Using denoising networks for regularizing inverse imaging problems. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 1799–1808. IEEE Computer Society, 2017.

Antoine Monod, Julie Delon, Matias Tassano, and Andrés Almansa. Video restoration with a deep plug-and-play prior. *arXiv preprint arXiv:2209.02854*, 2022.

Badr Moufad, Yazid Janati, Lisa Bedin, Alain Oliviero Durmus, randal douc, Eric Moulines, and Jimmy Olsson. Variational diffusion posterior sampling with midpoint guidance. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=6EUtjXAvmj.

Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 257–265, 2016. URL https://api.semanticscholar.org/CorpusID:8671030.

William S. Peebles and Saining Xie. Scalable diffusion models with transformers. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182, 2022. URL https://api.semanticscholar.org/CorpusID:254854389.

Marcelo Pereyra. Proximal markov chain monte carlo algorithms. *Statistics and Computing*, 26: 745–760, 2016.

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*.

Aram-Alexandre Pooladian, Heli Ben-Hamu, Carles Domingo-Enrich, Brandon Amos, Yaron Lipman, and Ricky T. Q. Chen. Multisample flow matching: Straightening flows with minibatch couplings, 2023. URL https://arxiv.org/abs/2304.14772.

Robin Rombach, A. Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10674–10685, 2021. URL https://api.semanticscholar.org/CorpusID:245335280.

Wonyong Seo, Jihyong Oh, and Munchurl Kim. Bim-vfi: Bidirectional motion field-guided frame interpolation for video with non-uniform motions, 2025. URL https://arxiv.org/abs/2412.11365.

Wei Shang, Dongwei Ren, Yi Yang, Hongzhi Zhang, Kede Ma, and Wangmeng Zuo. Joint video multi-frame interpolation and deblurring under unknown exposure time. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13935–13944, 2023. URL https://api.semanticscholar.org/CorpusID:257767169.

Jascha Narain Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. *ArXiv*, abs/1503.03585, 2015. URL https://api.semanticscholar.org/CorpusID:14888175.

Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *International Conference on Learning Representations*, 2023a. URL https://api.semanticscholar.org/CorpusID:259298715.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Neural Information Processing Systems*, 2019. URL https://api.semanticscholar.org/CorpusID:196470871.

Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33:12438–12448, 2020.

Yang Song, Jascha Narain Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *ArXiv*, abs/2011.13456, 2020. URL https://api.semanticscholar.org/CorpusID:227209335.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning*, 2023b. URL https://api.semanticscholar.org/CorpusID:257280191.

Alessio Spagnoletti, Jean Prost, Andrés Almansa, Nicolas Papadakis, and Marcelo Pereyra. Latino-pro: Latent consistency inverse solver with prompt optimization, 2025. URL https://arxiv.org/abs/2503.12615.

Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 237–246, 2017. URL https://api.semanticscholar.org/CorpusID:5872410.

Yu Sun, Brendt Wohlberg, and Ulugbek S Kamilov. An online plug-and-play algorithm for regularized image reconstruction. *IEEE Transactions on Computational Imaging*, 5(3):395–408, 2019.

Julián Tachella, Matthieu Terris, Samuel Hurault, Andrew Wang, Dongdong Chen, Minh-Hai Nguyen, Maxime Song, Thomas Davies, Leo Davy, Jonathan Dong, Paul Escande, Johannes Hertrich, Zhiyuan Hu, Tobías I. Liaudat, Nils Laurent, Brett Levac, Mathurin Massias, Thomas Moreau, Thibaut Modrzyk, Brayan Monroy, Sebastian Neumayer, J'er'emy Scanvic, Florian Sarron, Victor Sechaud, Georg Schramm, Romain Vo, and Pierre Weiss. Deepinverse: A python package for solving imaging inverse problems with deep learning. 2025. URL https://api.semanticscholar.org/CorpusID:278910576.

Hong Ye Tan, Subhadip Mukherjee, Junqi Tang, and Carola-Bibiane Schönlieb. Provably convergent plug-and-play quasi-Newton methods. *SIAM Journal on Imaging Sciences*, 17(2):785–819, 2024.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport, 2024. URL https://arxiv.org/abs/2302.00482.

Singanallur V Venkatakrishnan, Charles A Bouman, and Brendt Wohlberg. Plug-and-Play priors for model based reconstruction. In *2013 IEEE Global Conference on Signal and Information Processing*, pp. 945–948. IEEE, dec 2013. ISBN 978-1-4799-0248-4. doi: 10.1109/GlobalSIP.2013.6737048. URL http://brendt.wohlberg.net/publications/pdf/venkatakrishnan-2013-plugandplay2.pdf.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23:1661–1674, 2011. URL https://api.semanticscholar.org/CorpusID:5560643.

Maxime Vono, Nicolas Dobigeon, and Pierre Chainais. Split-and-augmented Gibbs sampler - Application to large-scale inference problems. *IEEE Transactions on Signal Processing*, 67(6): 1648–1661, 2019.

Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Xiaofeng Meng, Ningying Zhang, Pandeng Li, Ping Wu, Ruihang Chu, Rui Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wen-Chao Zhou, Wente Wang, Wen Shen, Wenyuan Yu, Xianzhong Shi, Xiaomin Huang, Xin Xu, Yan Kou, Yan-Mei Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhengbin Han, Zhigang Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *ArXiv*, abs/2503.20314, 2025. URL https://api.semanticscholar.org/CorpusID:277321639.

Xiang Wang, Shiwei Zhang, Han Zhang, Yu Liu, Yingya Zhang, Changxin Gao, and Nong Sang. Videolcm: Video latent consistency model. *ArXiv*, abs/2312.09109, 2023. URL https://api.semanticscholar.org/CorpusID:266209871.

Zhou Wang, Alan Conrad Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13:600–612, 2004. URL https://api.semanticscholar.org/CorpusID:207761262.

Ping Wu, Kai Zhu, Yu Liu, Liming Zhao, Wei Zhai, Yang Cao, and Zhengjun Zha. Improved video vae for latent video diffusion model. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18124–18133, 2024. URL https://api.semanticscholar.org/CorpusID:273962787.

Tongda Xu, Ziran Zhu, Jian Li, Dailan He, Yuanyuan Wang, Ming Sun, Ling Li, Hongwei Qin, Yan Wang, Jingjing Liu, and Ya-Qin Zhang. Consistency model is an effective posterior sample approximation for diffusion inverse solvers, 2024. URL https://arxiv.org/abs/2403.12063.

Tianwei Yin, Michael Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6613–6623, 2023. URL https://api.semanticscholar.org/CorpusID:265506768.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and Bill Freeman. Improved distribution matching distillation for fast image synthesis. *Advances in Neural Information Processing Systems*, 37:47455–47487, 2024a.

Tianwei Yin, Qiang Zhang, Richard Zhang, William T. Freeman, Frédo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22963–22974, 2024b. URL `https://api.semanticscholar.org/CorpusID:274610175`.

Yuanhao Zhai, Kevin Qinghong Lin, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Chung-Ching Lin, David S. Doermann, Junsong Yuan, and Lijuan Wang. Motion consistency model: Accelerating video diffusion with disentangled motion-appearance distillation. *ArXiv*, abs/2406.06890, 2024. URL `https://api.semanticscholar.org/CorpusID:270379579`.

K. Zhang, Yawei Li, Wangmeng Zuo, Lei Zhang, Luc Van Gool, and Radu Timofte. Plug-and-play image restoration with deep denoiser prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44:6360–6376, 2020. URL `https://api.semanticscholar.org/CorpusID:221377171`.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018. URL `https://api.semanticscholar.org/CorpusID:4766599`.

Youjian Zhang, Chaoyue Wang, Stephen J Maybank, and Dacheng Tao. Exposure trajectory recovery from motion blur. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11): 7490–7504, 2021.

Yuanzhi Zhu, K. Zhang, Jingyun Liang, Jiezhang Cao, Bihan Wen, Radu Timofte, and Luc Van Gool. Denoising diffusion models for plug-and-play image restoration. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1219–1229, 2023. URL `https://api.semanticscholar.org/CorpusID:258714952`.

A Appendix

A.1 Implementation of the forward operators

For all the inverse problems considered, we use the following formulation

$$\mathcal{A} = \texttt{SpatialSR} \circ \texttt{TemporalSR}.$$

For *Temporal SR*$\times 4$ + *Spatial SR*$\times 4$, we apply a temporal average pooling with factor 4 (with end padding if T is not divisible), followed by frame-wise spatial downsampling with factor 4 (`DeepInv.Downsampling` Tachella et al. (2025)). The adjoint $\mathcal{A}^\top$ first applies the spatial adjoint (back-projection to HR) and then the adjoint of temporal averaging (nearest upsample by 4 divided by 4, with folding of the padded tail back to the last frame when T is not a multiple of 4). The same approach, but with $\times 8$, is adopted for the *Temporal SR*$\times 8$ + *Spatial SR*$\times 8$ problem. For the *Temporal blur* + *Spatial SR*$\times 8$ task, we use a 1D temporal uniform convolution with circular boundary conditions via FFT of window size of 7, followed by frame-wise spatial downsampling with factor 8; the adjoint corresponds to spatial back-projection and time-reversed temporal filtering via FFT.

A.2 Implementation of Likelihood Proximal Steps

We will now describe the implementation of the likelihood updates in the splitting scheme (Equation(7)) instantiated by task-specific linear operators $\mathcal{A}$ over videos $\boldsymbol{x} \in \mathbb{R}^{(T+1)\times H\times W\times 3}$. We remind that we have to solve the following problems:

$$\underset{\boldsymbol{u}\in\mathbb{R}^{(T+1)\times H\times W\times 3}}{\arg\min}\; g_{\boldsymbol{y}}(\boldsymbol{u}) + \phi_\lambda(\boldsymbol{u}) + \tfrac{1}{2\delta\eta}\|\tilde{\boldsymbol{x}}_{k+1/4} - \boldsymbol{u}\|_2^2, \tag{8}$$

and

$$\underset{\boldsymbol{u}\in\mathbb{R}^{(T+1)\times H\times W\times 3}}{\arg\min}\; g_{\boldsymbol{y}}(\boldsymbol{u}) + \tfrac{1}{2\delta(1-\eta)}\|\tilde{\boldsymbol{x}}_{k+3/4} - \boldsymbol{u}\|_2^2, \tag{9}$$

where $g_{\boldsymbol{y}}(\cdot) = \frac{1}{2\sigma_n^2}\|\mathcal{A}\cdot - \boldsymbol{y}\|_2^2$.

Starting from Equation (9), we notice that this is exactly the shape of the $\mathrm{prox}_{\delta(1\,-\,\eta)/2\|\mathcal{A}\cdot-\boldsymbol{y}\|_2^2}(\boldsymbol{u})$, we thus provide details about the computation of this step.

Quadratic proximal (ℓ_2 data term). Given $\epsilon > 0$ (which may include δ, η as well as the noise variance σ_n^2), the quadratic likelihood proximal operator

$$\mathrm{prox}_{\frac{\epsilon}{2}\|\mathcal{A}\,\cdot - \boldsymbol{y}\|_2^2}(\boldsymbol{u}) = \arg\min_{\boldsymbol{x}} \tfrac{\epsilon}{2}\|\mathcal{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \tfrac{1}{2}\|\boldsymbol{x} - \boldsymbol{u}\|_2^2$$

reduces to the normal equations

$$\big(\mathrm{Id} + \epsilon\,\mathcal{A}^\top\mathcal{A}\big)\,\boldsymbol{x} \;=\; \boldsymbol{u} + \epsilon\,\mathcal{A}^\top\boldsymbol{y},$$

where Id is the identity operator. The exact solution is computationally tractable in high dimensions when $\mathcal{A}$ admits a closed-form and fast SVD (Zhang et al., 2020)[4], but to make our method applicable to general operators, we solve this linear system approximately using ~ 10 *Conjugate Gradient* (CG) (Hestenes & Stiefel, 1952) iterations.

CG is a Krylov-subspace method that iteratively refines an approximate solution $\boldsymbol{x}^{(k)}$ without explicitly inverting $\mathrm{Id} + \epsilon\mathcal{A}^\top\mathcal{A}$. Starting from the initial guess $\boldsymbol{x}^{(0)} = \boldsymbol{u}$, we iteratively update:

$$\begin{aligned}
&\boldsymbol{r}^{(k)} = \boldsymbol{b} - \big(\mathrm{Id} + \epsilon\mathcal{A}^\top\mathcal{A}\big)\boldsymbol{x}^{(k)}, && \boldsymbol{b} := \boldsymbol{u} + \epsilon\,\mathcal{A}^\top\boldsymbol{y},\\
&\boldsymbol{p}^{(k)} = \boldsymbol{r}^{(k)} + \beta^{(k)}\boldsymbol{p}^{(k-1)}, && \beta^{(k)} := \frac{\|\boldsymbol{r}^{(k)}\|_2^2}{\|\boldsymbol{r}^{(k-1)}\|_2^2},\\
&\alpha^{(k)} = \frac{\|\boldsymbol{r}^{(k)}\|_2^2}{\big\langle \boldsymbol{p}^{(k)},\, (\mathrm{Id} + \epsilon\mathcal{A}^\top\mathcal{A})\boldsymbol{p}^{(k)}\big\rangle}, &&\\
&\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha^{(k)}\boldsymbol{p}^{(k)}, && \boldsymbol{r}^{(k+1)} = \boldsymbol{r}^{(k)} - \alpha^{(k)}\big(\mathrm{Id} + \epsilon\mathcal{A}^\top\mathcal{A}\big)\boldsymbol{p}^{(k)}.
\end{aligned}$$

The algorithm terminates after a fixed number of iterations or once the residual norm $\|\boldsymbol{r}^{(k)}\|_2$ falls below a tolerance (e.g. 10^{-6}). Because $\mathrm{Id} + \epsilon\mathcal{A}^\top\mathcal{A}$ is symmetric positive definite, CG converges rapidly.

This iterative scheme is memory-efficient, requiring only matrix–vector products with $\mathcal{A}$ and $\mathcal{A}^\top$, and avoids the explicit computation of $\mathcal{A}^\top\mathcal{A}$, making it suitable for large-scale inverse problems and long video sequences.

Spatio-temporal TV$_3$ proximal (PDHG). For the regularised subproblem (8), we solve

$$\min_{\boldsymbol{u}}\; \underbrace{\tfrac{1}{2\sigma_n^2}\|\mathcal{A}\boldsymbol{u} - \boldsymbol{y}\|_2^2 + \tfrac{1}{2\delta\eta}\|\boldsymbol{u} - \tilde{\boldsymbol{x}}_{k+1/4}\|_2^2}_{f(\boldsymbol{u})} \;+\; \underbrace{\phi_\lambda(\boldsymbol{u})}_{g(D_\lambda\boldsymbol{u})}, \tag{10}$$

where

$$\phi_\lambda(\boldsymbol{u}) \;=\; \mathrm{TV}_{3,\lambda}(\boldsymbol{u}) \;:=\; \sum_{\tau,c,i,j} \sqrt{\lambda_h^2\big(D_h\boldsymbol{u}_{\tau,c,i,j}\big)^2 + \lambda_v^2\big(D_v\boldsymbol{u}_{\tau,c,i,j}\big)^2 + \lambda_t^2\big(D_\tau\boldsymbol{u}_{\tau,c,i,j}\big)^2}\,,$$

and $D_\lambda := \big[\,\lambda_h D_h,\; \lambda_v D_v,\; \lambda_\tau D_\tau\,\big]$, so that $g(D_\lambda\boldsymbol{u}) = \|D_\lambda\boldsymbol{u}\|_2$.

[4] For **Problems A, B, C**, the SVD of $\mathcal{A}$ can be expressed in terms of Fourier transforms, only if convolutions are periodic, which is not always the case for the kind of spatial and temporal blur we have in our case.

The associated subproblem in (10) is convex and can be solved using the *primal–dual hybrid gradient* (PDHG, Chambolle–Pock) algorithm Chambolle & Pock (2011). Let $\boldsymbol{p} = (p_h, p_v, p_\tau)$ denote the dual variable with three components per voxel. Given stepsizes $\rho, \sigma > 0$ such that $\rho\sigma\|D_\lambda\|^2 < 1$ and extrapolation $\theta \in [0, 1]$, the iterations read:

$$\boldsymbol{p}^{k+1} = \mathrm{prox}_{\sigma g^*}\big(\boldsymbol{p}^k + \sigma D_\lambda \bar{\boldsymbol{u}}^k\big) = \frac{\boldsymbol{p}^k + \sigma D_\lambda \bar{\boldsymbol{u}}^k}{\max\big(1,\ \|\boldsymbol{p}^k + \sigma D_\lambda \bar{\boldsymbol{u}}^k\|_2\big)} \quad \text{(projection onto unit } \ell_2 \text{ ball)},$$

$$\boldsymbol{u}^{k+1} = \mathrm{prox}_{\rho f}\big(\boldsymbol{u}^k - \rho D_\lambda^\top \boldsymbol{p}^{k+1}\big),$$

$$\qquad \text{obtained by solving } \big(I + \rho(\mathcal{A}^\top\mathcal{A} + \tfrac{1}{\delta\eta} I)\big)\boldsymbol{u}^{k+1} = \boldsymbol{z} + \rho\Big(\mathcal{A}^\top \boldsymbol{y} + \tfrac{1}{\delta\eta}\tilde{\boldsymbol{x}}_{k+1/4}\Big),$$

$$\qquad \text{with } \boldsymbol{z} = \boldsymbol{u}^k - \rho D_\lambda^\top \boldsymbol{p}^{k+1},$$

$$\bar{\boldsymbol{u}}^{k+1} = \boldsymbol{u}^{k+1} + \theta(\boldsymbol{u}^{k+1} - \boldsymbol{u}^k).$$

Here $D_\lambda^\top \boldsymbol{p} = \lambda_h D_h^\top p_h + \lambda_v D_v^\top p_v + \lambda_\tau D_\tau^\top p_\tau$ is the weighted divergence, and the proximal step for $f(\boldsymbol{u}) = \frac{1}{2\sigma_n^2}\|\mathcal{A}\boldsymbol{u} - \boldsymbol{y}\|_2^2 + \frac{1}{2\delta\eta}\|\boldsymbol{u} - \tilde{\boldsymbol{x}}_{k+1/4}\|_2^2$ is implemented by solving the normal equations. As in our implementation $\delta\eta$ is often $\geq 10^5$, to simplify the computations we remove the regularization term $\frac{1}{2\delta\eta}\|\boldsymbol{u} - \tilde{\boldsymbol{x}}_{k+1/4}\|_2^2$. Around 10 iterations of the CG algorithm can be used to solve the normal equations, as they are warm-started with $\boldsymbol{u}^k$.

In practice, we apply Chambolle–Pock (~ 200 iterations) only in the *pure temporal TV* case ($\lambda_h = \lambda_v = 0$). When spatial weights are nonzero ($\lambda_h > 0$ or $\lambda_v > 0$), we instead minimise (8) directly with ADAM (Kingma & Ba, 2014) (learning rate 10^{-3}, 100 iterations), which proved more robust in this setting.

A.3 THE LATINO ALGORITHM

In order to clarify the practical implementation of the splitting scheme introduced in Equation (5), we provide here the pseudo-code to implement LATINO as described in Spagnoletti et al. (2025).

Algorithm 2 LATINO

```
given $\boldsymbol{x}_0 = \mathcal{A}^\dagger \boldsymbol{y}$, text prompt $c$, number of steps $N$, latent consistency model $f_\theta$, latent space
    decoder $\mathcal{D}$, latent space encoder $\mathcal{E}$, sequences $\{t_k, \delta_k\}_{k=0}^{N-1}$.
for $k = 0, \dots, N-1$ do
    $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathrm{Id})$
    $\boldsymbol{z}_{t_k}^{(k)} \leftarrow \sqrt{\alpha_{t_k}}\mathcal{E}(\boldsymbol{x}_k) + \sqrt{1 - \alpha_{t_k}}\boldsymbol{\epsilon}$        ▷ Encode
    $\boldsymbol{u}^{(k)} \leftarrow \mathcal{D}(f_\theta(\boldsymbol{z}_{t_k}^{(k)}, t_k, c))$        ▷ Decode
    $\boldsymbol{x}_{k+1} \leftarrow \mathrm{prox}_{\delta_k g_y}(\boldsymbol{u}^{(k)})$        ▷ $g_{\boldsymbol{y}} : \boldsymbol{x} \mapsto -\log p(\boldsymbol{y}|\boldsymbol{x})$
end for
return $\boldsymbol{x}_N$
```

A.4 THE VISION-XL ALGORITHM

VISION-XL Kwon & Ye (2025b) (Video Inverse-problem Solver using latent diffusION models) is a SOTA framework for high-resolution video inverse problems, LDMs such as SDXL to restore videos from measurements affected by spatio-temporal degradations.

Components VISION-XL integrates three main contributions: (i) *Pseudo-batch inversion*, which initializes the sampling process from latents obtained by DDIM-inverting the measurement frames. (ii) *Pseudo-batch sampling*, which splits latent video frames and samples them in parallel using Tweedie's formula Efron (2011), reducing memory requirements to that of a single frame. (iii) *Pixel-space data-consistency updates*, where each denoised batch $\hat{\boldsymbol{x}}_t$ is refined using l iterations of a quadratic proximal step

$$\bar{\boldsymbol{x}}_t = \arg\min_{\boldsymbol{x} \in \hat{\boldsymbol{x}}_t + K_l} \|\boldsymbol{y} - \mathcal{A}(\boldsymbol{x})\|_2^2,$$

typically solved via conjugate gradient (CG). This enforces alignment with the measurement before re-encoding to the latent space and re-noising for the next step.

Overall Algorithm. Starting from $\boldsymbol{z}_\rho = \mathrm{DDIM}^{-1}(E_\theta(\boldsymbol{y}))$ with $\rho \approx 0.3T$, VISION-XL alternates denoising in latent space and proximal data-consistency refinement in pixel space. After decoding the denoised latent batch $\hat{\boldsymbol{x}}_t = D_\theta(\hat{\boldsymbol{z}}_t)$, a low-pass filter is applied to suppress high-frequency inconsistencies before re-encoding and re-noising, yielding $\boldsymbol{z}_{t-1}$. This process is repeated until $t = 0$, as shown in Algorithm 3.

Algorithm 3 VISION-XL

```
Require: Pretrained VAE encoder $\mathcal{E}_\theta$, decoder $\mathcal{D}_\theta$, denoiser $E_\theta^{(t)}$, measurement $\boldsymbol{x}$, forward operator $\mathcal{A}$, initial DDIM inversion step $\rho$, CG iterations $l$, low-pass filter widths $\{\sigma_t\}$, noise schedule $\{\bar\alpha_t\}_{t=1}^T$
$\boldsymbol{z}_0 \leftarrow \mathcal{E}_\theta(\boldsymbol{y})$
$\boldsymbol{z}_\rho \leftarrow \mathrm{DDIM}^{-1}(\boldsymbol{z}_0)$    ▷ Step 1: **Pseudo-batch inversion** (informative latent initialization)
for $t = \rho, \dots, 2$ do
    $\hat{\boldsymbol{z}}_t \leftarrow \dfrac{\boldsymbol{z}_t - \sqrt{1-\bar\alpha_t}\, E_\theta^{(t)}(\boldsymbol{z}_t)}{\sqrt{\bar\alpha_t}}$    ▷ Step 2: **Pseudo-batch sampling** (Tweedie's formula)
    $\hat{\boldsymbol{x}}_t \leftarrow \mathcal{D}_\theta(\hat{\boldsymbol{z}}_t)$
    $\bar{\boldsymbol{x}}_t \leftarrow \arg\min_{\boldsymbol{x} \in \hat{\boldsymbol{x}}_t + \mathcal{K}_l} \| \boldsymbol{y} - \mathcal{A}(\boldsymbol{x}) \|_2^2$    ▷ Step 3: **Data-consistency refinement** (multi-step proximal via $l$ CG steps)
    $\bar{\boldsymbol{x}}_t \leftarrow \bar{\boldsymbol{x}}_t * h_{\sigma_t}$    ▷ Step 4: **Scheduled low-pass filtering** (mitigate VAE error accumulation)
    $\bar{\boldsymbol{z}}_t \leftarrow \mathcal{E}_\theta(\bar{\boldsymbol{x}}_t)$
    $\boldsymbol{z}_{t-1} \leftarrow \sqrt{\bar\alpha_{t-1}}\bar{\boldsymbol{z}}_t + \sqrt{1-\bar\alpha_{t-1}}\mathcal{E}_t$    ▷ Step 5: **Renoising** (batch-consistent noise)
end for
$\boldsymbol{z}_0 \leftarrow \dfrac{\boldsymbol{z}_1 - \sqrt{1-\bar\alpha_1}\, E_\theta^{(1)}(\boldsymbol{z}_1)}{\sqrt{\bar\alpha_1}}$
return $\boldsymbol{x}_0 \leftarrow \mathcal{D}_\theta(\boldsymbol{z}_0)$
```

A.5 Connection with PnP-Flow algorithms

The PnP-Flow Martin et al. (2025) algorithm designed to leverage Flow Matching image priors has some direct connections to LATINO Spagnoletti et al. (2025). For this reason, we now briefly introduce their setting and state how this idea can be extended to Video Flow models.

Let $X_0 \sim P_0$ denote a latent variable and $X_1 \sim P_1$ a data variable, with joint law $(X_0, X_1) \sim \pi$. Assume we are given a pre-trained Flow Matching model with velocity field

$$v_\theta : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d, \qquad (t, \boldsymbol{x}) \mapsto v_\theta(t, \boldsymbol{x}),$$

learned by minimizing the Conditional Flow Matching (CFM) Lipman et al. (2023) loss along the straight-line interpolation Liu et al. (2022); Benton et al. (2024)

$$X_t := e_t(X_0, X_1) := (1-t)X_0 + tX_1, \qquad t \in [0,1].$$

Time-dependent denoiser from Flow Matching. From the velocity field v_θ we define a family of time-dependent denoisers

$$D_t(\boldsymbol{x}) \ := \ \boldsymbol{x} + (1-t)\, v_\theta(t, \boldsymbol{x}), \qquad t \in [0,1]. \tag{11}$$

To motivate this choice, recall that for each $t \in [0,1]$ the population minimizer $v_t^\star$ of the CFM loss satisfies

$$v_t^\star(\boldsymbol{x}) \ = \ \mathbb{E}[X_1 - X_0 \,|\, X_t = \boldsymbol{x}]\,,$$

so that in the ideal case $v_\theta(t, \cdot) = v_t^\star(\cdot)$ one has

$$D_t(\boldsymbol{x}) \ = \ \boldsymbol{x} + (1-t)\, v_t^\star(x) \ = \ \mathbb{E}[X_1 \,|\, X_t = \boldsymbol{x}]\,. \tag{12}$$

Thus D_t coincides with the minimum mean-square-error (MMSE) estimator of the clean variable X_1 given a noisy point X_t on the interpolation path. Equivalently, D_t solves the regression problem

$$D_t \in \arg\min_g \mathbb{E}\big[\|X_1 - g(X_t)\|^2\big] ,$$

and can be interpreted as a time-indexed denoiser that projects points lying along the straight path $(X_t)_{t\in[0,1]}$ onto the target distribution P_1.

In particular, if the FM flow is *straight-line* in the sense that $X_t = (1-t)X_0 + tX_1$ is realized by the associated flow ODE, then D_t can perfectly recover X_1 from X_t. Under mild regularity assumptions, one can show that the mean-squared error $\mathbb{E}\big[\|D_t(X_t) - X_1\|^2\big]$ vanishes for all $t \in [0,1]$ if and only if the learned flow forms a straight-line Flow Matching pair between X_0 and X_1.[5] This highlights the particular suitability of straight-line FM models (e.g. OT-FM Pooladian et al. (2023); Tong et al. (2024)) as building blocks for PnP priors.

PnP Flow Matching algorithm. Martin et al. (2025) incorporates the denoisers $\{D_t\}_{t\in[0,1]}$ into a Forward–Backward Splitting (FBS) scheme for solving imaging inverse problems of the form

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} F(\boldsymbol{x}) + R(\boldsymbol{x}),$$

where F is a differentiable data-fidelity term (e.g. negative log-likelihood), and R is an implicit prior induced by the generative model. Classical PnP-FBS Meinhardt et al. (2017); Sun et al. (2019); Hurault et al. (2022); Tan et al. (2024) replace the proximal operator of R by a *time-independent* denoiser, applied directly after the gradient step on F.

In contrast, PnP-Flow introduces two key modifications:

1. A *time-dependent* denoiser D_t as in (11), indexed by a schedule $(t_n)_n \subset [0,1]$ with $t_n \nearrow 1$.
2. An intermediate *interpolation/reprojection step* that maps the gradient iterate back onto the straight FM path before denoising.

Given an initial guess $\boldsymbol{x}_0 \in \mathbb{R}^d$, a sequence of times $(t_n)_n$ with $t_n \in [0,1]$ and $t_n \to 1$, and stepsizes $(\gamma_n)_n$, each PnP-Flow iteration at time t_n proceeds as follows:

1. Gradient step. Move towards data consistency by a gradient descent step on F:

$$\boldsymbol{z}_n = \boldsymbol{x}_n - \gamma_n \nabla F(\boldsymbol{x}_n).$$

2. Interpolation (reprojection) step. The denoiser D_{t_n} is trained to act on points distributed as X_{t_n}, i.e. lying on the straight-line FM path. The output $\boldsymbol{z}_n$ of the gradient step does not follow this distribution, so we "reproject" it onto the FM trajectory by drawing a latent sample $\varepsilon \sim P_0$ and forming

$$\tilde{\boldsymbol{z}}_n = (1-t_n)\,\varepsilon + t_n\,\boldsymbol{z}_n. \tag{13}$$

Intuitively, $\tilde{\boldsymbol{z}}_n$ mimics a point at time t_n on a straight path between a latent sample from P_0 and the current gradient iterate.

3. PnP denoising step. Finally, we apply the FM-induced denoiser at time t_n,

$$\boldsymbol{x}_{n+1} = D_{t_n}(\tilde{\boldsymbol{z}}_n) = \tilde{\boldsymbol{z}}_n + (1-t_n)\,v_\theta\big(t_n, \tilde{\boldsymbol{z}}_n\big), \tag{14}$$

which pushes $\tilde{\boldsymbol{z}}_n$ towards the data distribution while still respecting the measurement model encoded in F.

The resulting discrete-time algorithm, summarized in Algorithm 4, alternates between a data-fidelity gradient step, an interpolation onto FM trajectories, and a generative PnP denoising step. The time parameter t_n controls the relative weight of the prior: for small t_n, the denoiser has a strong effect (large factor $1-t_n$ in (14)), while as $t_n \to 1$ the updates gradually become more likelihood-driven. Comparing Algorithm 4 to Algorithm 2, it is clear that both adapt the same core idea: data-term → add noise → denoise and repeat. Both LATINO and PnP-Flow reproject the intermediate step $\boldsymbol{x}_n$ to a point in the Flow ODE, to which is applied, in one case, the CM, and in the other, the FM denoiser.

[5] See Proposition 1 in Martin et al. (2025) for a precise statement and proof.

Algorithm 4 PnP–Flow Matching

```
Input: Pre-trained Flow Matching network $v_\theta$, time sequence $(t_n)_n$ with $t_n \in [0,1]$ and $t_n \nearrow 1$, step sizes $(\gamma_n)_n$, data-fidelity $F : \mathbb{R}^d \to \mathbb{R}$, prior $P_0$ (e.g. standard Gaussian), initial iterate $\boldsymbol{x}_0 \in \mathbb{R}^d$
for $n = 0, 1, 2, \dots$ do
    $\boldsymbol{z}_n \leftarrow \boldsymbol{x}_n - \gamma_n \nabla F(\boldsymbol{x}_n)$    ▷ gradient step on data-fidelity
    Sample $\varepsilon \sim P_0$    ▷ latent noise
    $\tilde{\boldsymbol{z}}_n \leftarrow (1 - t_n)\, \varepsilon + t_n\, \boldsymbol{z}_n$    ▷ interpolation along the flow path
    $\boldsymbol{x}_{n+1} \leftarrow \tilde{\boldsymbol{z}}_n + (1 - t_n)\, v_\theta\big(t_n, \tilde{\boldsymbol{z}}_n\big)$    ▷ PnP denoising with FM-induced denoiser $D_{t_n}$
end for
Output: Reconstruction $\boldsymbol{x}_{n+1}$
```

The other difference is in the type of data-fidelity term adopted; in one case, it is a proximal step as a result of an implicit Euler step, while in the other, it is a gradient one, which is equivalent to an explicit Euler step and requires many more iterations to converge due to the limitations on γ_n.

Given these similarities, it is natural to think about merging the two frameworks by leveraging few-step FMs Liu et al. (2022); Kornilov et al. (2024) in place of CMs. This would lead to a Flow-SAE that could be plugged into the LATINO algorithm and provide a different way to integrate the prior term in Equation (5). As a direct consequence, given a video FM prior, it can be deployed in place of the VCM in our Algorithm 1, and benefit from the modular framework introduced in this work, as it can be coupled with an ICM, or an image FM prior, and the TV3 term. We believe that future research may benefit from this Flow-L∀TINO formulation to improve the quality of restorations and further generalize our setting.

A.6 Ablation study

To better understand the impact of the data-consistency updates in L∀TINO, we perform an ablation study comparing different strategies for the likelihood *proximal steps* appearing in Equation (7). Furthermore, we provide results on **Problem A** and **Problem B** obtained with a lighter version of L∀TINO that only includes the VCM prior. We call this version L∀TINO-V and we provide in Algorithm 5 its implementation.

In Table 4 we find the hyperparameters used to get Table 1 in Section 4. These values were chosen after an extensive grid search on $\lambda = (\lambda_h, \lambda_w, \lambda_\tau), \eta, \gamma$; nevertheless, other combinations also produced satisfactory results, and we want to illustrate some alternative choices in this section.

Problem	$(\lambda_h,\, \lambda_v,\, \lambda_\tau)$	$\eta\delta$	$(1-\eta)\delta$
A	$(0,\ 0,\ 0.005)$	10^5	10^5
B	$(0,\ 0,\ 0)$	10^5	$2{\times}10^3$
C	$(10^{-4},\ 10^{-4},\ 10^{-6})$	10^5	10^5

Table 4: Hyperparameters used in (7).

L∀TINO: w\ and w\o TV. As we can see from Table 4, it seems better to keep the TV prior term ϕ_λ when we solve **Problem A**, while it is better to fall back on the prox-only case (*i.e.* $\lambda = (0,0,0)$) when we tackle **Problem B**. We then show in Table 5 what happens in the two symmetric cases, meaning when we switch the optimal configurations of **Problem A** with those of **Problem B**. We can observe how the metrics do not change much for **Problem B**, as we are still able to beat the SOTA VISION-XL method in half of the metrics (in particular, we focus on the FVMD that tells us how temporally consistent the reconstruction is). As opposed to this, we see that we lose a lot of precision for **Problem A** in all the metrics. This can be explained by the fact that the TV prior is crucial when dealing with temporal interpolation, as it prevents the ICM from creating flickering effects.

L∀TINO-V as a lighter alternative. As anticipated, we also provide some results when we turn off the ICM part of the L∀TINO algorithm, meaning that we set $\eta = 1$. This solution, described in Algorithm 5, only presents choices in one data-fidelity step, which we can again tune as a TV-regularized step or as a classical prox-only step. We provide in Table 5 both cases. The values of λ and δ are the same as Table 4, meaning that the TV case will follow the **Problem A** row and the prox case the **Problem B** row. We see how this lighter version can still beat VISION-XL in almost all metrics with only 5 NFEs. In particular, since we no longer have the ICM, the TV prior loses its importance, and the prox case emerges as the best option. L∀TINO-V is capable of getting highly temporally coherent reconstructions, as shown by the low FVMD values, only losing to LVTINO, especially in LPIPS, as its single frame quality suffers from the limitations of the VCM. We believe that further research could fill the gap between LVTINO and LVTINO-V, developing new SOTA methods that solely use VCMs, without the need for its image counterpart, to increase spatial quality.

		Temp. SR×4 + SR×4				Temp. blur + SR×8			
Method (Data-Consistency Config)	**NFE↓**	**FVMD↓**	**PSNR↑**	**SSIM↑**	**LPIPS↓**	**FVMD↓**	**PSNR↑**	**SSIM↑**	**LPIPS↓**
LVTINO-V (prox)	**5**	<u>425.2</u>	25.00	<u>0.811</u>	<u>0.270</u>	**31.70**	23.80	0.737	<u>0.375</u>
LVTINO (ICM: prox, **VCM:** prox)	9	607.5	22.59	0.614	0.475	<u>42.65</u>	<u>24.91</u>	<u>0.741</u>	**0.370**
LVTINO-V (TV)	**5**	503.3	24.44	0.776	0.338	578.0	22.01	0.684	0.441
LVTINO (ICM: prox, **VCM: TV)**	9	**371.1**	**27.25**	**0.837**	**0.249**	51.52	23.18	0.725	0.418
VISION-XL	<u>8</u>	1141	<u>26.03</u>	0.672	0.439	82.92	**26.18**	**0.749**	0.468
ADMM-TV	–	427.6	18.04	0.767	0.297	128.2	21.18	0.644	0.452

Table 5: **Ablation study on data-consistency schemes.** Left block: results for *temporal SR*×4 + *SR*×4, **Problem A**. Right block: results for *temporal blur* + *SR*×8, **Problem B**.

Algorithm 5 LVTINO-V

```
given degraded video $\boldsymbol{y}$, operator $\mathcal{A}$, initialization $\boldsymbol{x}_0 = \mathcal{A}^\dagger \boldsymbol{y}$, video lenght $T+1$, steps $N=5$
given video CM $(\mathcal{E}_V, \mathcal{D}_V, f_\vartheta^V)$, schedules $\{t_k, \delta_k, \lambda\}_{k=0}^{N-1}$, $g_{\boldsymbol{y}}$
for $k = 0, \dots, N-1$ do
    $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathrm{Id}_{(1+T/4)\times H/8 \times W/8 \times C})$
    $\boldsymbol{z}_{t_k}^{(k)} \leftarrow \sqrt{\alpha_{t_k}}\, \mathcal{E}_V(\boldsymbol{x}_k) \;+\; \sqrt{1-\alpha_{t_k}}\, \boldsymbol{\epsilon}$    ▷ encode & diffuse to $t_k$
    $\tilde{\boldsymbol{x}}_{k+1/2} \leftarrow \mathcal{D}_V\big(f_\vartheta^V(\boldsymbol{z}_{t_k}^{(k)}, t_k)\big)$    ▷ VCM prior contraction
    $\boldsymbol{x}_{k+1} \leftarrow \arg\min_{\boldsymbol{u} \in \mathbb{R}^{(T+1)\times H \times W \times 3}} g_{\boldsymbol{y}}(\boldsymbol{u}) + \phi_\lambda(\boldsymbol{u}) + \frac{1}{2\delta_k} \|\tilde{\boldsymbol{x}}_{k+1/2} - \boldsymbol{u}\|_2^2$    ▷ data-consistency
    Solved with a few CG iters;  TV-in-time can be used here.
end for
return $\boldsymbol{x}_N$
```

A.7 Additional Experiments and Analyses

Comparisons to other baselines. To provide a more comprehensive evaluation, we extend our comparison to include non-zero-shot methods, such as VIDUE Shang et al. (2023), which is explicitly trained for joint motion-blur removal and frame interpolation. This makes it a highly relevant baseline for the combined blur and interpolation tasks of **Problem C**, whereas standard Video Frame Interpolation (VFI) methods often fail to address motion blur. We indeed specifically compare VIDUE against the recent BiM-VFI Seo et al. (2025). As shown in Figure 7, because BiM-VFI is trained specifically for interpolation, it fails to remove the degradation caused by motion blur. In contrast, VIDUE addresses the joint problem more effectively. As VIDUE does not perform super-resolution, we apply bicubic upsampling (×8) to its output for fair comparison to LVTINO. The results are shown in Table 1 and Table 2.

We also acknowledge that DiffIR2VR is a relevant competitor to VISION-XL, and thus to LVTINO. However, the specific `Stable Diffusion v2.1` checkpoint required to reproduce their method is no longer publicly available, which prevents a fair comparison.

Figure 7: Visual comparison on **Problem C**. Left: BiM-VFI preserves blur artifacts. Right: VIDUE removes some motion blur.

Noisier cases. We now show results computed on the Adobe240 dataset for a higher noise scenario with $\sigma_y = 0.01$. As expected, the optimization step in VISION-XL fails to properly restore the video sequences in this case, as VISION-XL is not conceived to deal with noisy measurements, yielding `NaN` values. In contrast, LVTINO and ADMM-TV handle this case without difficulty. Their results are reported in Table 6, together with VIDUE for **Problem C**.

	Problem A: Temp. SR×4 + SR×4					Problem B: Temp. blur + SR×8					Problem C: Temp. SR×8 + SR×8				
Method	**NFE↓**	**FVMD↓**	**PSNR↑**	**SSIM↑**	**LPIPS↓**	**NFE↓**	**FVMD↓**	**PSNR↑**	**SSIM↑**	**LPIPS↓**	**NFE↓**	**FVMD↓**	**PSNR↑**	**SSIM↑**	**LPIPS↓**
LVTINO	9	**256.5**	**24.95**	**0.782**	**0.331**	9	**62.6**	**21.91**	**0.671**	**0.448**	7	310.6	**23.42**	**0.688**	**0.428**
VIDUE	–	–	–	–	–	–	–	–	–	–	**1**	**121.5**	21.33	0.603	0.511
ADMM-TV	–	424.1	17.85	0.758	0.373	–	145.5	21.35	0.646	0.471	–	1665	18.12	0.652	0.475

Table 6: Results on the Adobe240 dataset with noise $\sigma_y = 0.01$ across the three problems. Best results are in **bold**, second best are underlined.

Non-linear Inverse Problems. Although for presentation clarify we present LVTINO in the context of linear inverse problems, LVTINO can be applied to non-linear problems too. The main requirement is the ability to evaluate the proximal operator of the log-likelihood, which is feasible for many non-linear degradations, as already shown in Spagnoletti et al. (2025).

To demonstrate this, we consider a non-linear degradation: Additive Gaussian noise ($\sigma = 0.01$) followed by JPEG compression (quality=10) applied to each frame independently. Figure 8 shows frames extracted from the reconstruction results. LVTINO successfully recovers high-frequency details and suppresses compression artifacts, confirming its applicability to non-linear inverse problems.

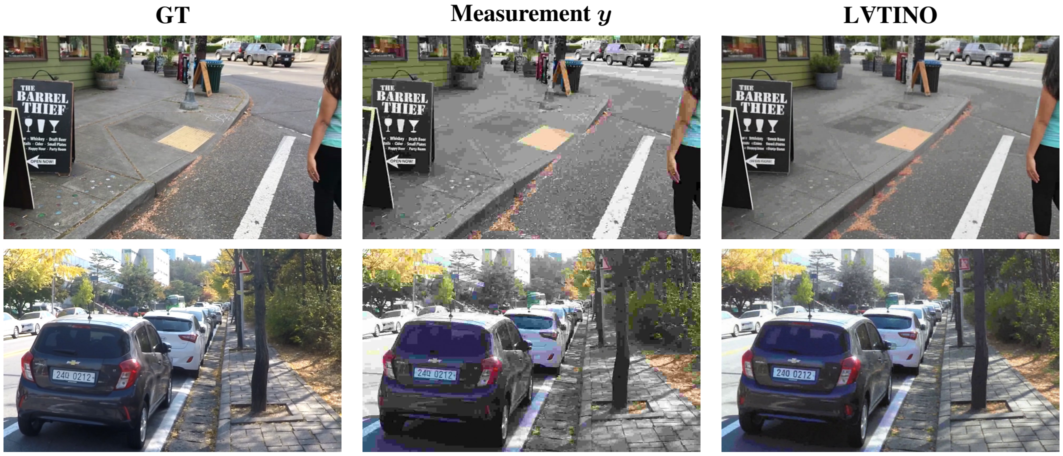


Figure 8: Results on a non-linear inverse problem (Gaussian Noise + JPEG compression). Top row: Example from Adobe240. Bottom row: Example from GoPRO240. LVTINO effectively removes blocking artifacts and noise in both cases.

Hyperparameter Sensitivity. We analyze the stability of LVTINO with respect to the step size δ and the regularization weight λ. Figure 9 plots PSNR and LPIPS metrics on a representative sequence from the challenging **Problem C**. We observe that performance remains stable across a reasonable range of values (e.g., $\delta \in [2 \cdot 10^4, 2 \cdot 10^5]$). This indicates that the parameters reported in Table 4 are not brittle, and ϵ-good hyperparameters can be found without exhaustive fine-tuning.

In a similar way, it is also possible to analyse the parameter η, which controls the balance between the VCM and the ICM in our theoretical framework (see equation (6)). It must be translated into practice by choosing the corresponding evaluation times t_V and t_I. In particular, when η increases, the t_V is larger, and the ICM is evaluated at a smaller t_I. Because pretrained Consistency Models are only accurate on a restricted subset of timesteps, this severely limits how finely we can tune η in practice.

To approximate different effective values of η, we therefore perform an ablation in which we vary the possible video timesteps t_V and image timesteps t_I within the valid finetuned ranges of the two backbones. Operationally, we choose among the subsets:

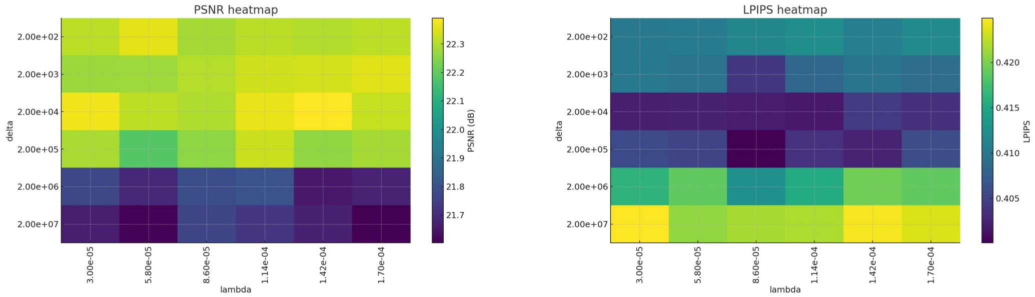


Figure 9: Sensitivity analysis for **Problem C**. The method shows robust performance across a wide range of step sizes (δ) and regularization weights (λ).

- **VCM (video model):** ($[757, 522, 375, 255, 125]$)
- **ICM (image model):** ($[749, 624, 499, 374, 249, 124, 63]$)

and pairing them to simulate "larger η" (larger VCM steps + smaller ICM steps) and "smaller η" (smaller VCM steps + larger ICM steps).

For clarity, Table 7 shows some configurations evaluated to provide a comparison for **Problem B** (Temporal SR×4 + SR×4). Each configuration is evaluated on the same sample sequence:

Experiment	t_V **(video)**	t_I **(image)**	**PSNR** ↑	**SSIM** ↑	**LPIPS** ↓
EXP 1	[375, 255, 125]	[499, 374]	22.90	0.752	0.296
EXP 2	[757, 522, 375]	[249, 124]	22.56	0.714	0.308
EXP 3	[522, 375, 255]	[374, 249]	22.80	0.762	0.290
EXP 4	[522, 255, 125]	[749, 624]	22.36	0.720	0.317
BASELINE	[757, 522, 375, 255, 125]	[374, 249, 124, 63]	23.96	0.770	0.272
VISION-XL	—	—	24.36	0.667	0.488

Table 7: Ablation study on scheduling strategies (t_V and t_I) for **Problem B**. EXP 1-4 represent varying balances of η, while BASELINE represents the configuration used in the main paper.

For comparison, the values used for the experiments shown in the other tables are: $t_V \in [757, 522, 375, 255, 125]$ and $t_I \in [374, 249, 124, 63]$. We notice how, even with fewer steps and varying the configurations, the metrics remain stable.

Error Map Analysis. To better visualize the nature of the residuals, we provide L_2 error maps in Figure 10 for **Problem C** on an example sequence. Comparing LVTINO against VISION-XL and ADMM-TV, we observe that our method yields lower residuals, particularly around motion boundaries and fine structural details where competing methods exhibit larger errors due to unresolved blur or temporal inconsistencies.

B Additional Examples

We provide in Table 8 qualitative video comparisons for **Problem A**, **Problem B**, and **Problem C**. Each triplet corresponds to the Ground Truth (GT), the observed degraded input ($\boldsymbol{y}$), and the restored sequence. For **Problem C**, we provide longer sequences (81 frames) to better appreciate the results.

Additional examples are shown in Figures 12,13,14,15,16. We also include additional sliced images in Figures 11a and 11b.

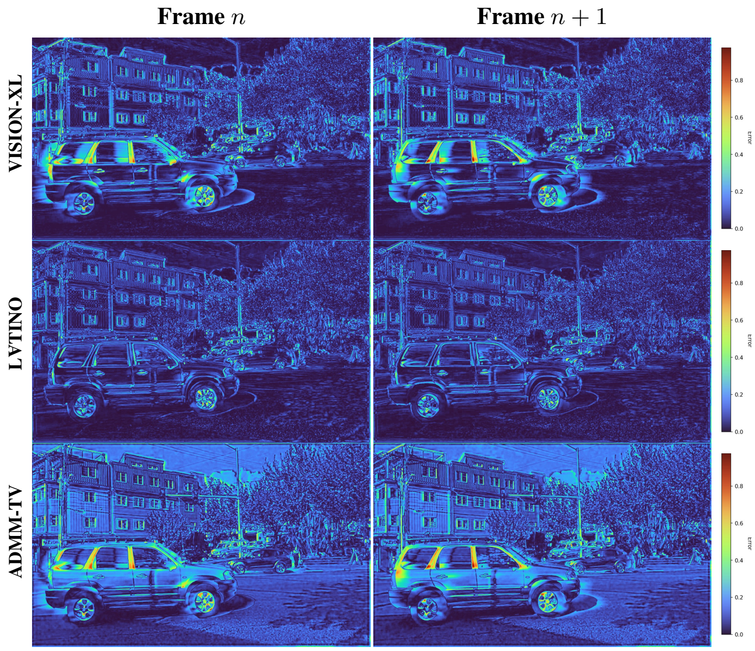
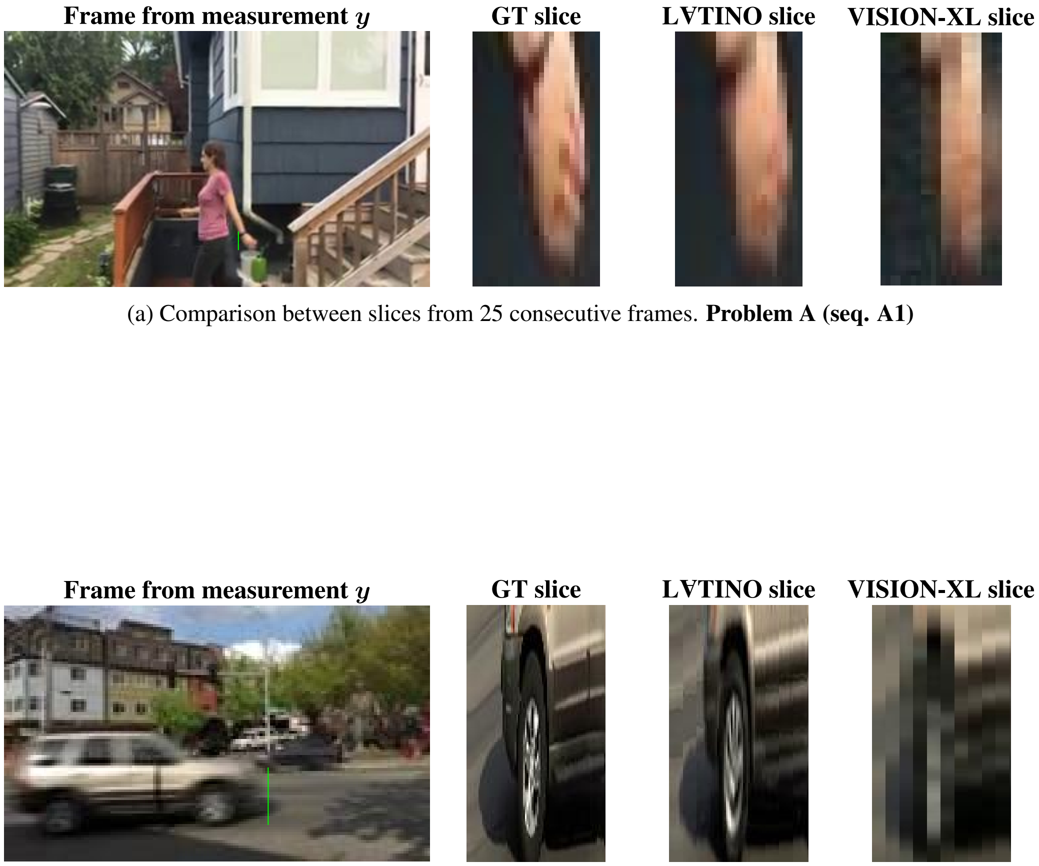


Figure 10: L_2 error maps between reconstructions and Ground Truth. L∀TINO (middle row) demonstrates lower error magnitude compared to VISION-XL and ADMM-TV, particularly in dynamic regions.

	GT	y	L∀TINO	VISION-XL
Problem A (seq. A1)	link	link	link	link
Problem B (seq. B1)	link	link	link	link
Problem B (seq. B2)	link	link	link	link
Problem C (seq. C1)	link	link	link	link
Problem C (seq. C2)	link	link	link	link

Table 8: Results of our method compared to those obtained by VISION-XL, ground truth, and measurements (input sequence). Click the links to see the videos.



(a) Comparison between slices from 25 consecutive frames. **Problem A (seq. A1)**

(b) Comparison between slices from 81 consecutive frames. **Problem C (seq. C1)**

Figure 11: Slice comparisons across two sequences. In green, the sliced column. Slice images are obtained from the three-dimensional video tensor (i,j,τ) by fixing a column index j. This leads to a 2D tensor with indices (i,τ) that is represented as an image, where the i index represents the row and the t index represents the column.

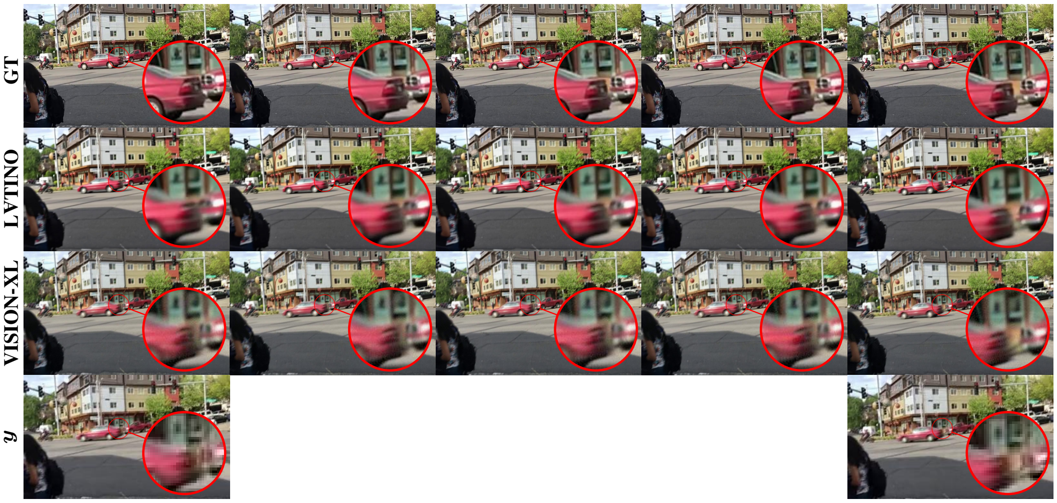


Figure 12: Visual comparison for **Problem A**.

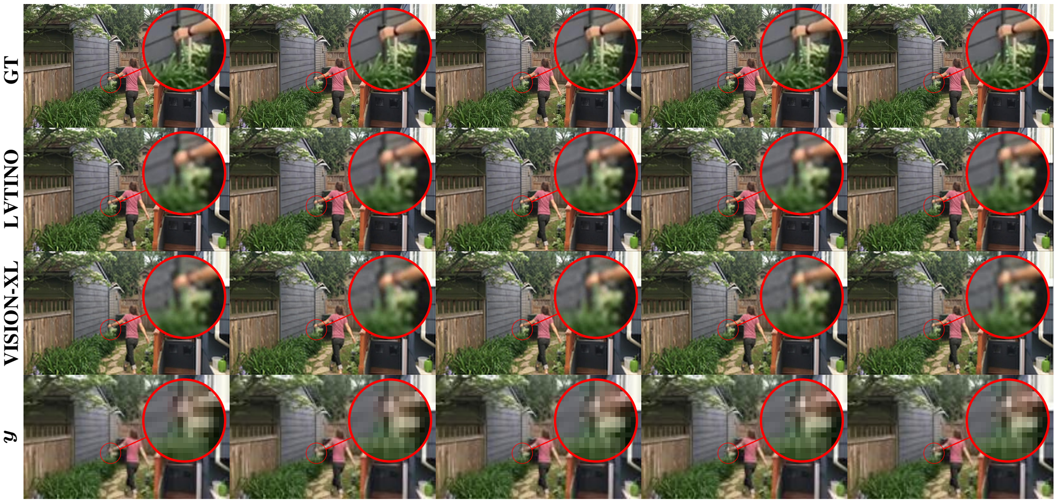


Figure 13: Visual comparison for **Problem B**.

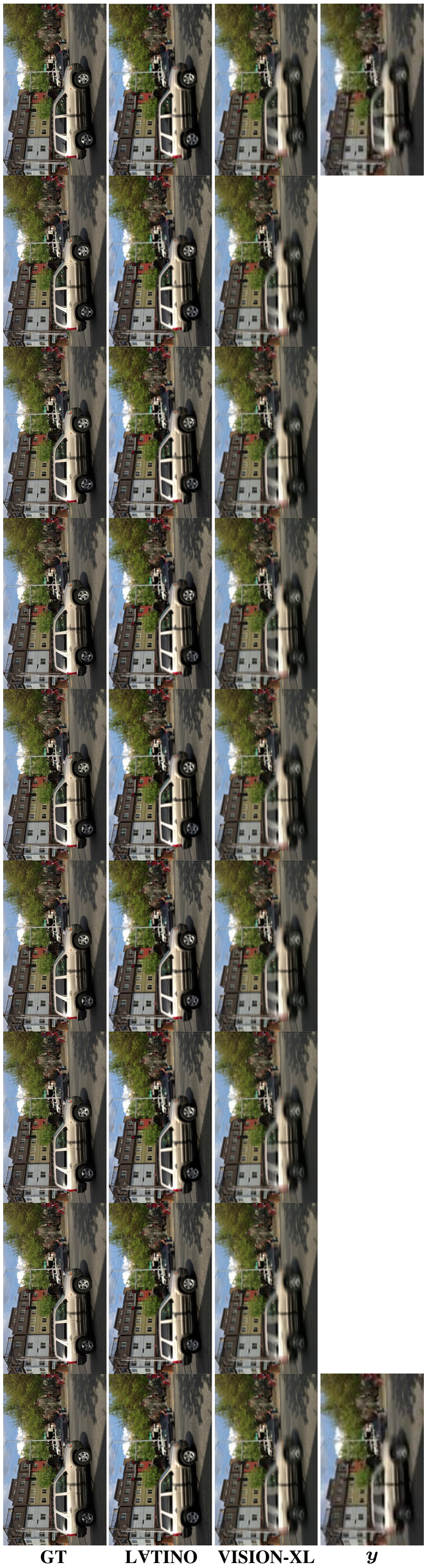


Figure 14: Visual comparison for **Problem C**.

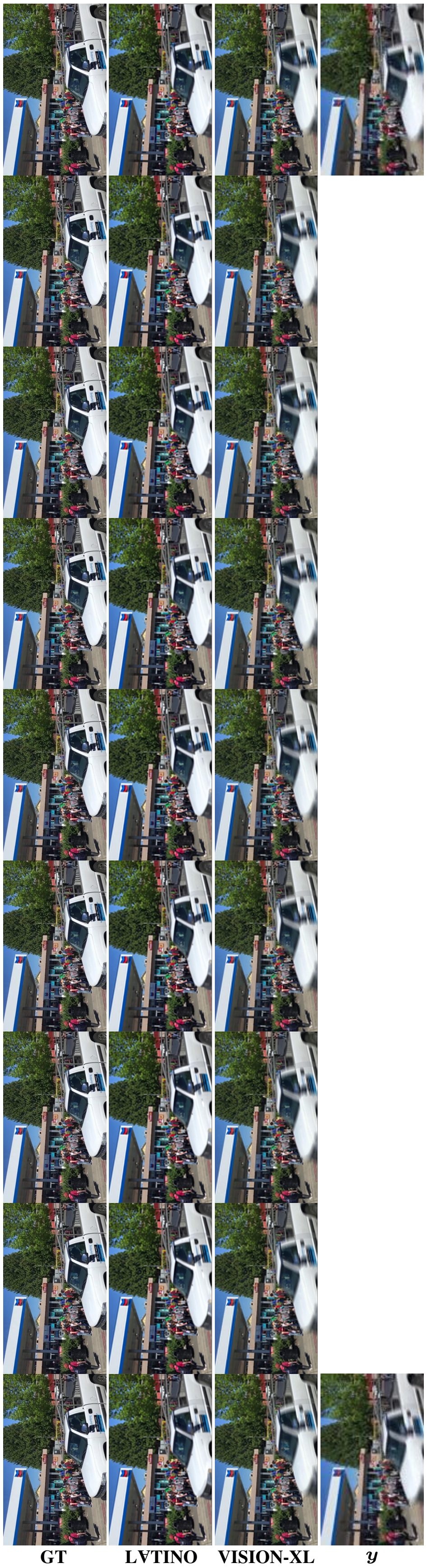


Figure 15: Visual comparison for **Problem C**.

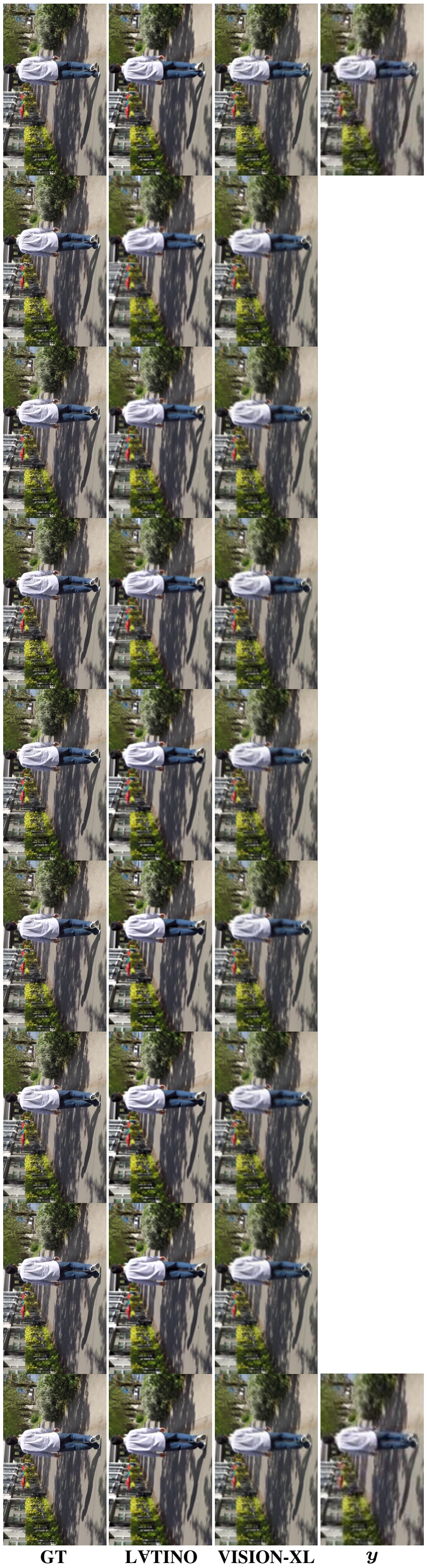


Figure 16: Visual comparison for **Problem C**.