Learning few-step posterior samplers by unfolding and distillation of diffusion models

Anonymous authors
Paper under double-blind review

Abstract

Diffusion models (DMs) have emerged as powerful image priors in Bayesian computational imaging. Two primary strategies have been proposed for leveraging DMs in this context: Plug-and-Play methods, which are zero-shot and highly flexible but rely on approximations; and specialized conditional DMs, which achieve higher accuracy and faster inference for specific tasks through supervised training. In this work, we introduce a novel framework that integrates deep unfolding and model distillation to transform a DM image prior into a few-step conditional model for posterior sampling. A central innovation of our approach is the unfolding of a Markov chain Monte Carlo (MCMC) algorithm—specifically, the recently proposed LATINO Langevin sampler (Spangnoletti et al., 2025)—representing the first known instance of deep unfolding applied to a Monte Carlo sampling scheme. We demonstrate our proposed unfolded and distilled samplers through extensive experiments and comparisons with the state of the art, where they achieve excellent accuracy and computational efficiency, while retaining the flexibility to adapt to variations in the forward model at inference time.

1 Introduction

We seek to infer an unknown image of interest $x^\star \in \mathbb{R}^d$ from a noisy measurement $\mathbf{y} = y$, related to $x^\star$ by a statistical observation model of the form

$$\mathbf{y} \sim P(\boldsymbol{A}x^\star)\,, \tag{1}$$

where $\boldsymbol{A}$ is a linear measurement operator describing deterministic instrumental aspects of the data acquisition process and P is a statistical model describing measurement noise. This includes, for instance, the canonical linear Gaussian observation model $\mathbf{y} = \boldsymbol{A}x^\star + \mathbf{n}$, where $\mathbf{n}$ is additive Gaussian noise with covariance $\sigma_n^2\boldsymbol{I}$. Such models arise frequently in problems related to image deblurring, inpainting, super-resolution, compressive sensing, and tomographic reconstruction (see, e.g., (Kaipio & Somersalo, 2010; Ongie et al., 2020)). Within this context, we are particularly interest in imaging problems where recovering $x^\star$ from y is severely ill-conditioned or ill-posed, leading to significant uncertainty about the solution (Kaipio & Somersalo, 2010).

Adopting a Bayesian approach, we leverage prior knowledge available in order to regularize the problem and deliver meaningful inferences that are well-posed. This is achieved by treating $x^\star$ as a realization of a random variable $\mathbf{x}_0$ and using Bayes' theorem to obtain the posterior distribution of $\mathbf{x}_0|\mathbf{y} = y$, with density given for all $x_0 \in \mathbb{R}^d$ by (Robert, 2007)

$$p(x_0|y) = \frac{p(y|x_0)p(x_0)}{\int_{\mathbb{R}^d} p(y|\tilde{x}_0)p(\tilde{x}_0)\mathrm{d}\tilde{x}_0}\,,$$

where $p(y|x_0)$ denotes the likelihood function associated to Equation 1 and $p(x_0)$ the marginal density of $\mathbf{x}_0$.

The choice of $p(x_0)$, so-called prior, is crucial and greatly impacts the obtained posterior $p(x_0|y)$, especially in problems that are ill-conditioned or ill-posed. Bayesian imaging methods traditionally used analytical priors designed to promote solutions with expected structural properties (e.g., smoothness, sparsity, piecewise regularity). However, modern Bayesian imaging methods rely predominantly on machine learning in order to harness vast amounts of prior knowledge available in the form of training data and deliver unprecedented accuracy (Mukherjee et al., 2023). In particular, state-of-the-art Bayesian imaging techniques often use deep

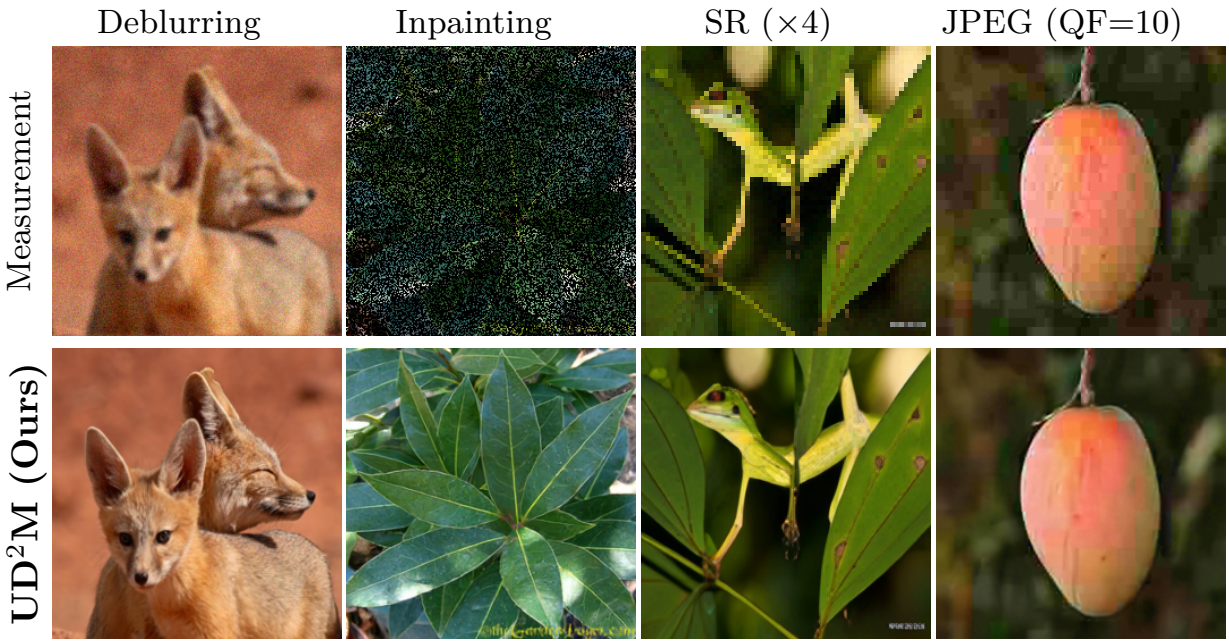


Figure 1: Qualitative comparison of the proposed Unfolded and Distilled Diffusion Model (UD^{2}M) for posterior sampling on the ImageNet 256 dataset. Tasks: Gaussian Deblurring, random inpainting (70%), super-resolution (4×), and restoration of JPEG compression artifacts (QF=10).

generative models as image priors, notably denoising diffusion models (DMs). For an introduction to DMs, we refer the reader to the seminal DM papers Song & Ermon (2019; 2020); Song et al. and to the recent survey on DMs as image priors for solving for inverse problems (Daras et al., 2024).

The literature on DM-based Bayesian imaging methodology has two main strands. On the one hand, Plug & Play (PnP) methods seek to use a pretrained DM in a zero-shot manner, in combination with a likelihood function $p(y|x_0)$ specified analytically during inference. PnP approaches are flexible by construction and generalize robustly to new measurement models by exploiting knowledge of the likelihood function $p(y|x)$ explicitly. Notable examples of PnP DM techniques include, e.g., DPS (Chung et al., 2022), DDRM (Kawar et al., 2022), DiffPIR (Zhu et al., 2023), ΠGDM (Song et al., 2023a), and mid-point (Moufad et al., 2025).

The other strand relies on fine-tuning an unconditional DM representing the prior, in order to construct a y-conditional DM for posterior sampling. This involves modifying a DM to take the measurement y as input and, through supervised training, specializing it for posterior sampling of $\mathbf{x}_0|\mathbf{y} = y$ (instead of marginal sampling of $\mathbf{x}_0$). Specializing a DM for a specific imaging task enables the development of Bayesian imaging methods that surpass PnP approaches in both accuracy and computational efficiency, assuming the task is fully defined during training. However, in the absence of additional training, such task-specific DMs often exhibit limited generalization to even slightly altered measurement conditions, as they are not inherently designed to exploit the likelihood $p(y|x_0)$ during inference. For more details and comparisons with PnP strategies, see, e.g., I^{2}SB (Liu et al., 2023a), InDI (Delbracio & Milanfar, 2023), CDDB (Chung et al., 2023).

Moreover, the most advanced DM-based imaging methods currently available -whether zero-shot or trained for conditional sampling- leverage model distillation to enhance computational efficiency and improve performance. Notably, distillation techniques such as consistency models (Song et al., 2023c; Kim et al., 2024) and flow matching (Liu et al., 2023b; Lipman et al., 2023) have significantly reduced the number of required neural function evaluations (NFEs) per posterior sample from over 10^3 to fewer than 10, while concurrently improving sampling quality and accuracy (see Spangnoletti et al. (2025); Garber & Tirer (2025); Zhao et al. (2025)).

To bridge the methodological gap between these two strands, this paper introduces a novel framework for constructing conditional DMs which offers the computational efficiency and accuracy of conditional DMs

obtained via finetuning and distillation, with the flexibility afforded by explicit likelihood functions in inference, thus bringing together the advantages of finetuning and PnP strategies. This is achieved by generalizing deep unfolding (Monga et al., 2021), a paradigm wherein iterative algorithms with a fixed number of steps are mapped onto modular deep neural network architectures. These architectures explicitly incorporate the likelihood function $p(y|x_0)$, for example through its gradient or proximal operator, in conjunction with a denoiser that is subsequently fine-tuned during training. A main novelty of this work is to present the first application of deep unfolding of a Markov chain Monte Carlo (MCMC) sampling scheme, in contrast to prior work on deep unfolding that focused on optimization-based algorithms (which, unlike MCMC schemes, are not inherently suitable for posterior sampling). Namely, we propose to unfold the state-of-the-art LATINO zero-shot imaging Langevin method (Spangnoletti et al., 2025), and train the resulting unfolded LATINO networks by using a supervised consistency trajectory models objective combining distortion, perceptual and adversarial terms (Kim et al., 2024). This leads to distilled conditional DMs that are fast (low NFEs), generate high-quality samples, are GPU-memory efficient during inference (due to the absence of automatic differentiation). We refer to the proposed method as an Unfolded and Distilled Diffusion Model (UD^{2}M). Crucially, UD^{2}Ms support joint training over a family of likelihoods, followed by instantaneous specialization to specific measurement models at inference time (i.e., specific instances of $\boldsymbol{A}$ and P). For illustration, Figure 1 shows some examples of posterior samples obtained with the proposed unfolded and distilled diffusion models (UD^{2}Ms) applied to Gaussian deblurring, random inpainting, super-resolution by a factor 4, and JPEG artifact restoration for images from ImageNet dataset.

2 Background & related works

Diffusion models (DMs) are generative models that draw samples from a distribution of interest $p(\mathbf{x}_0)$ by iteratively reversing the following "noising" process

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

which is designed to transport $\mathbf{x}_0$ to a standard normal random variable $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ as $T \to \infty$, and where β_t represents a noise schedule and $\mathbf{w}_t$ is a Brownian motion (Ho et al., 2020). The reverse process, which allows generating samples of $\mathbf{x}_0$ by transporting $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ back to $\mathbf{x}_0$, is given by (Song et al., 2021):

$$d\mathbf{x}_t = \left[-\frac{\beta_t}{2}\mathbf{x}_t - \beta_t \nabla_{x_t} \log p_t(\mathbf{x}_t)\right] dt + \sqrt{\beta_t} d\overline{\mathbf{w}_t}, \tag{3}$$

where $\overline{\mathbf{w}_t}$ is again Brownian motion, in the reverse direction. The target distribution $p(\mathbf{x}_0)$ is encoded in the so-called score function $x_t \mapsto \nabla_{x_t} \log p_t(x_t)$, which is approximated using Tweedie's identity (Efron, 2011) by a deep neural network denoiser $G_\theta(\cdot, t)$ that is trained by weighted denoising score matching (DSM), see Song et al. (2021).

Denoising diffusion probabilistic models (DDPM) (Ho et al., 2020) and denoising diffusion implicit models (DDIM) (Song et al., 2021) are the two main methodologies to implement Equation 3 in a time-discrete setting, where the process is initialized with $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ for some large finite T and solved iteratively with t decreasing progressively to $t = 0$. Accurate estimation of the scores $\nabla_{x_t} \log p_t(x_t)$ underpinning Equation 3 has been crucial to the success of DDPM and DDIM. This has been achieved using large training datasets, specialized network architectures, and significant high-performance computing resources (Ho et al., 2020; Song et al., 2021). In addition to advancing generative modeling, DMs have become central to modern strategies for solving inverse problems in computational imaging (see, e.g., the recent survey paper (Daras et al., 2024)).

Distillation of diffusion models Conventional DM approaches produce high quality samples by progressively transporting $\mathbf{x}_T$ to $\mathbf{x}_0$. However, they have a high computational cost, as generating each sample typically requires performing between 100 and 1000 evaluations of G_θ. This drawback can be addressed by distilling G_θ into a new model $\tilde{G}_\vartheta$ requiring far fewer steps (e.g., 4-8 steps); see, e.g., Song et al. (2023c); Liu et al. (2023b). Remarkably, modern distillation techniques combining DSM with adversarial training are able to further improve the quality of the samples as they dramatically reduce the number of sampling steps, while retaining the same network architecture as the original DM. This is the case, for example, of the so-called consistency trajectory models (Kim et al., 2023), which we use in our proposed method in Section 4.

Consistency models Consistency models (CMs) are derived from a probability flow formulation of Equation 3, given by (Song et al., 2023b)

$$dx_t = \left[-\frac{\beta_t}{2} x_t - \frac{\beta_t}{2} \nabla_{x_t} \log p_t(x_t) \right] dt. \tag{4}$$

This ODE is equivalent to Equation 3 in the sense that both have the same marginals $p_t(\cdot)$ of $\mathbf{x}_t$. CMs leverage this property by learning a so-called consistency function $\tilde{G}_\theta$ that maps any point x_t on a trajectory $\{x_t\}_{t\in[\eta,T]}$ of Equation 4 backwards to x_η, for some small given $\eta > 0$. Given $\tilde{G}_\theta : (x_t, t) \to x_\eta$, CMs are able to sample the r.v. $\mathbf{x}_\eta = \tilde{G}_\vartheta(\mathbf{x}_T, T)$ with $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ in a single step, see Kim et al. (2023) for details. Two-step CMs achieve superior quality by re-noising $\mathbf{x}_\eta = \tilde{G}_\vartheta(\mathbf{x}_T, T)$ following Equation 2 for some intermediate time $s \in (\eta, T)$ followed by $\tilde{G}_\vartheta(\mathbf{x}_s, s)$ to bring back $\mathbf{x}_s$ close to $\mathbf{x}_0$. Multi-step CMs apply this strategy recursively, in 4 to 8 steps, achieving top performance while retaining computational efficiency (Kim et al., 2024). Recent examples of few-step Bayesian image restoration methods based on CMs include, for instance, the fine-tuned method CoSIGN (Zhao et al., 2025), the zero-shot method CM4IR (Garber & Tirer, 2025), and the zero-shot method LATINO (Spangnoletti et al., 2025) which uses a so-called Stable Diffusion latent CM with semantic conditioning via text prompting.

LoRA fine-tuning of diffusion models Low-Rank Adaptation (LoRA) (Hu et al., 2022) is a technique designed to fine-tune large attention-based architectures such as DMs in a manner that achieves comparable performance to full fine-tuning at a fraction of the computational cost. This is achieved by adding a trainable low-rank correction Δ_θ to the model's original attention weights, which remain frozen during fine-tuning, and exploiting structural properties of attention layers in order to optimize the weights $\theta + \Delta_\theta$ at a reduced cost. LoRA fine-tuning has been widely applied to DMs. For example, in the context of DM distillation, CM-LORA strategies distill a pre-trained DM $G_\theta(\cdot, t)$ into a CM $\tilde{G}_\vartheta(\cdot, t) = G_{\theta+\Delta_\theta}(\cdot, t)$ by adjusting Δ_θ with a CM objective combining DSM and adversarial training (Luo et al., 2023).

Deep unfolding (also known as deep unrolling) is a deep learning paradigm that transforms iterative optimization algorithms with a fixed number of iterations into deep neural network architectures, whereby the algorithm's steps become trainable layers in a network that is trained end-to-end (Monga et al., 2021). This approach, also known as deep unrolling, provides a powerful template for designing interpretable network architectures that incorporate y and the operator $\boldsymbol{A}$ in a manner that is modular and explicit (e.g., via proximal operator layers), together with trainable elements (e.g. U-Net modules) that can be recognized as data-driven regularization terms. Because $\boldsymbol{A}$ is provided explicitly during training, unfolded networks can be easily trained to handle a range of operators (e.g., motion blurs, as represented by a database) and then instantiated for a specific operator during inference. To date, deep unfolding has primarily been implemented by unrolling optimization algorithms. A main innovation of this paper is the unfolding of a stochastic Langevin sampling algorithm instead (specifically, the LATINO algorithm of Spangnoletti et al. (2025)). Also, to the best of our knowledge, this is the first instance of deep unfolding in the context of DMs.

Langevin PnP methods for image restoration. DMs can serve as highly informative PnP priors within Langevin Markov chain Monte Carlo posterior samplers. Notably, Kemajou Mbakam et al. incorporates a DM within a PnP unadjusted Langevin algorithm (ULA) (Laumont et al., 2022) to estimate the posterior mean $\mathrm{E}(\boldsymbol{x}|\boldsymbol{y})$. Similarly, Coeurdoux et al. (2024b) embeds a DM within a split-Gibbs sampler (Vono et al., 2019), which is equivalent to a noisy ULA (Pereyra et al., 2023). With regards to other stragies not based on DMs, we note Coeurdoux et al. (2024a) and Melidonis et al. (2024) which explore the use of normalizing flows as PnP priors, while Holden et al. (2022) proposes a general theoretical framework for incorporating variational autoencoder (VAE) and generative adversarial network (GAN) priors.

LAtent consisTency INverse sOlver (LATINO) is a state-of-the-art zero-shot posterior sampling technique derived from embedding a CM prior $\tilde{G}_\vartheta(\cdot, t)$ within an overdamped Langevin diffusion process of the form $d\mathbf{x}_{0,s} = \nabla \log p(y|\mathbf{x}_{0,s})\mathrm{d}s + \nabla \log p(\mathbf{x}_{0,s})\mathrm{d}s + \mathrm{d}\mathbf{w}_s$, which converges to the posterior of interest $p(x_0|y)$ as $s \to \infty$, see Spangnoletti et al. (2025). In its simplest form, LATINO to sample from $p(x_0|y)$ is

based on the following recursion[1]: for all $k \in \mathbb{N}$

$$\begin{aligned}
\tilde{\mathbf{x}}_0^{(k+1)} &= \mathrm{prox}_{\delta g_y}(\mathbf{x}_0^{(k)}), && \texttt{implicit gradient step, } g_y : x_0 \mapsto -\log p(y|x_0) \\
\mathbf{x}_{t_\delta}^{(k+1)} &= \sqrt{\bar{\alpha}_{t_\delta}}\tilde{\mathbf{x}}_0^{(k+1)} + \sqrt{(1-\bar{\alpha}_{t_\delta})}\boldsymbol{\epsilon}_{k+1}, && \texttt{sample } \mathbf{x}_{t_\delta,k+1} \sim p(\cdot|\mathbf{x}_{0,k}) \texttt{ by using forward SDE 2} \\
\mathbf{x}_0^{(k+1)} &= \tilde{G}_\vartheta(\mathbf{x}_{t_\delta}^{(k+1)}, t_\delta), && \texttt{transport } \mathbf{x}_{t_\delta}^{(k+1)} \texttt{ back to } \mathbf{x}_0^{(k+1)} \texttt{ with ODE 4}
\end{aligned} \tag{5}$$

where δ, t_δ are step-sizes related to the time-discretization of the Langevin process, $\boldsymbol{\epsilon}_{k+1} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, and the proximal step $\mathrm{prox}_{\delta g_y}(s) = \arg\min_{x_0 \in \mathbb{R}^d} g_y(x_0) + \frac{1}{2\delta}\|x_0 - s\|_2^2/2$ with $g_y : x_0 \mapsto -\log p(y|x_0)$ is equivalent to an implicit Euler step on g_y, see Spangnoletti et al. (2025) for details. For linear Gaussian observation models of the form $y = \boldsymbol{A}x + \mathbf{n}$ with noise covariance $\sigma^2\boldsymbol{I}$, $\mathrm{prox}_{\delta g_y}(x) = (\delta\boldsymbol{A}^\top\boldsymbol{A} + \sigma^2\boldsymbol{I})^{-1}(\delta\boldsymbol{A}^\top y + \sigma^2 x)$, which can be computed efficiently by using a fast linear solver or a specialised scheme. Alternatively, for challenging linear operators and non-linear problems, we use an appropriate optimizer to compute an approximate solution of $\mathrm{prox}_{\delta g_y}(s) = \arg\min_{x_0 \in \mathbb{R}^d} g_y(x_0) + \frac{1}{2\delta}\|x_0 - s\|_2^2/2$, which is essentially a regularized least squares problem. It is worth mentioning that a more general form of the LATINO recursion generalizes Equation 5 by using a (deterministic) auto-encoder pair $(\mathcal{E}, \mathcal{D})$, which allows embedding CMs trained on the latent space of $(\mathcal{E}, \mathcal{D})$, notably modern stable diffusion CMs (Yin et al., 2024).

Integrating the CM $\tilde{G}_\vartheta$ within a Langevin sampler allows LATINO to use the likelihood function $x_0 \mapsto p(y|x_0)$ directly and exactly - through its proximal operator - without the need for approximations[2]. This represents a significant advantage over prevalent alternative DM-based posterior sampling approaches, which attempt to embed the likelihood into the DM itself — a process that is typically intractable and reliant on likelihood approximations because of time inhomogeneity. As a result, LATINO outperforms competing zero-shot DMs in sampling accuracy, with a computational cost comparable to state-of-the-art multi-step CMs (i.e., in the order of 4-8 iterations). The proposed UD^{2}M framework unfolds LATINO within a y-conditional distilled DM, which is trained end-to-end in a supervised manner for posterior sampling.

3 Proposed framework

3.1 UD^{2}Ms: Diffusion model unfolding and distillation for multi-step posterior sampling

We are now ready to present our proposed methodology for learning a posterior sampler for $(\mathbf{x}_0|\mathbf{y} = y)$, through unfolding and distillation of a DM that has been trained to sample $\mathbf{x}_0$ unconditionally (i.e., from the prior $p(x_0)$). We assume the availability of a DM $G_\theta(\cdot, t)$ pre-trained to approximate $G_\theta(x_t, t) \approx \mathbb{E}(\mathbf{x}_0|\mathbf{x}_t = x_t)$ with $\mathbf{x}_t = \sqrt{\overline{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\overline{\alpha}_t}\boldsymbol{\epsilon}_t$ following Equation 2. From this DM, we seek to derive a distilled model L_ϑ that, in a single step, samples approximately from the conditional distribution $p(\mathbf{x}_0|y, x_t)$. This distilled model L_ϑ is then embedded within a multi-step sampling scheme that iteratively generates $\mathbf{x}_t^{(k)} \sim p(\mathbf{x}_t|\mathbf{x}_0 = x_0^{(k-1)})$ exactly through the noising process of Equation 2, followed by approximate sampling of $\mathbf{x}_0^{(k)} \sim p(\mathbf{x}_0|y, x_t^{(k)})$ by using $L_\vartheta(y, x_t^{(k)}, t)$. The scheme starts at $t = T$ by sampling from the marginal $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, and the timestep t is reduced across iterations to reach $t = 0$ in a few steps (e.g., $N = 3$ steps). This scheme is summarized in Algorithm 1. Importantly, unlike zero-shot methods, here we consider that the likelihood function $p(y|x_0)$ associated to $p(\mathbf{x}_0|y, x_t)$ belongs a class of likelihood functions of interest that is fixed during the training of L_ϑ, while retaining the flexibility to specify the exact parameters of the likelihood during inference time. Specializing L_ϑ in this manner will lead to significant accuracy and computational cost gains when compared to zero-shot strategies that are fully likelihood agnostic during training. Note that L_ϑ samples the conditional random variable $(\mathbf{x}_0|y, \mathbf{x}_t)$, we do not seek to sample $(\mathbf{x}_0|y)$ directly.

Our UD^{2}M approach to construct L_ϑ combines distillation with deep-unfolding, whereby we obtain a trainable architecture from unfolding an iterative algorithm into a sequential scheme (Monga et al., 2021). As mentioned previously, a main novelty of this work is to unfold a Langevin sampler to construct a

[1]The steps of the recursion in Spangnoletti et al. (2025) are presented in a different order, first sampling using 2, then the CM step, and the proximal step last. We find that starting with the proximal step performs better when unrolling few iterations.

[2]The Langevin diffusion is a time-homogeneous process and the resulting iterates $\mathbf{x}_0^{(k)}$ are asymptotically ergodic, converging to a δ-neighbourhood of $p(x_0|y)$ as $k \to \infty$. The iterates are <u>not</u> traveling backwards in time through an inhomogeneous process and therefore there is no need to compute the likelihood of y w.r.t. a noisy version of $\mathbf{x}_0$, as is the case in conditional DMs.

generative model, which we train end-to-end. More precisely, given $t \in [0, T]$ and $\mathbf{x}_t = x_t$, we implement $L_\vartheta(\cdot, x_t, t)$ by unfolding K iterations of the LATINO Langevin sampling algorithm of Equation 5, modified to target the conditional distribution $p(\mathbf{x}_0|y, x_t)$. Within each LATINO module, the prior is represented by a distilled DM $G_{\theta+\Delta_\theta}$ derived from G_θ by LORA fine-tuning, where Δ_θ denotes LORA adaptation weights which we will train end-to-end for conditional sampling (with θ frozen), and where the likelihood $p(y, x_t|x_0) = p(y|x_0)p(x_t|x_0)$ associated with the posterior $p(\mathbf{x}_0|y, x_t)$ is involved via its proximal operator $\text{prox}_{\delta g_{y,x_t}}(s) = \arg\min_{x_0 \in \mathbb{R}^d} g_{y,x_t}(x_0) + \frac{1}{2\delta}\|x_0 - s\|_2^2/2$ where $g_{y,x_t} = -\log p(y, x_t|x_0)$. As mentioned previously, for linear Gaussian observation models of the form $y = \boldsymbol{A}x + \mathbf{n}$ with noise variance σ^2, this operator is given for all $x \in \mathbb{R}^d$ by

$$\text{prox}_{\delta g_{y,x_t}}(x) = \Sigma^{-1}\left(\frac{\boldsymbol{A}^\top y}{\sigma^2} + \frac{x_t}{\sqrt{\overline{\alpha}_t}\sigma_t^2} + \frac{x}{\delta}\right), \tag{6}$$

where $\sigma_t^2 = (1 - \overline{\alpha}_t)/\overline{\alpha}_t$ and the matrix Σ is defined as follows

$$\Sigma = \left(\frac{\boldsymbol{A}^\top \boldsymbol{A}}{\sigma^2} + \frac{\boldsymbol{I}}{\sigma_t^2} + \frac{\boldsymbol{I}}{\delta}\right) \tag{7}$$

For other models, we solve $\text{prox}_{\delta g_{y,x_t}}(s) = \arg\min_{x_0 \in \mathbb{R}^d} g_{y,x_t}(x_0) + \frac{1}{2\delta}\|x_0 - s\|_2^2/2$ with $g_{y,x_t} = -\log p(y, x_t|x_0)$ approximately by using Adam (Kingma & Ba, 2014) with x_0 as warm-starting.

Because of the low-rank structure of Δ_θ, and because we used tied weights across LATINO modules, the number of trainable parameters is usually of order of 0.1% of the number contained in θ, which makes adversarial training significantly more stable. Operating with a reduced number of parameters also greatly simplifies learning and storing adaptations of G_θ tailored for specific classes of problems of interest, and allows us to easily instantiate problem-specific variants of L_ϑ during inference. Moreover, Equation 6 can be computed cheaply when a singular value decomposition of $\boldsymbol{A}$ is available. Otherwise, if computing an exact solution efficiently is not possible, we use a specialised sub-iteration, a randomized linear solver, or some steps of a convex optimization scheme (Chambolle & Pock, 2016).

Lastly, to keep the number of LATINO modules K small without compromising accuracy, we warm-start the first LATINO module by using auxiliary neural estimator $\hat{x}_0 = f_\psi(\mathbf{y}, \mathbf{x}_t)$ as input module, parametrized by a set of weights ψ. For example, in our experiments we use the recently proposed foundational model RAM (Terris et al., 2025), with stacked operators $(\boldsymbol{A}, \boldsymbol{I})$ to take (y, x_t) as input. Alternatively, one can also initialize with $\boldsymbol{A}^\dagger y$. Accordingly, the set of trainable parameters of L_ϑ is $\vartheta = (\Delta_\theta, \psi)$ if f_ψ is finetuned for additional performance; otherwise we have $\vartheta = \Delta_\theta$ if ψ is frozen or if the first LATINO module is initialized directly with $\boldsymbol{A}^\dagger y$. The proposed architecture is summarized in Figure 2 below.

Before discussing our strategy for training L_ϑ, it is worth noting that one could in principle design L_ϑ to target the posterior $p(\mathbf{x}_0|y)$ directly, by using unfolding and distillation but bypassing fully the multi-step scheme. However, one-step samplers are notoriously harder to train, and the resulting samples are often of inferior quality (Luhman & Luhman, 2021; Meng et al., 2023; Song et al., 2023c; Salimans & Ho, 2022). Also, sampling $p(\mathbf{x}_0|y)$ directly with LATINO would require unfolding a larger number of iterations compared to sampling $p(\mathbf{x}_0|y, x_t)$, hence offsetting the computational benefits of targeting $p(\mathbf{x}_0|y)$ directly. This is due to the fact that the additional information provided by $\mathbf{x}_t = x_t$ leads to a posterior distribution that is much better conditioned and thus easier to sample by Langevin sampling. By using multi-step sampling, we obtain excellent sample quality with a reduced number of NFEs (e.g., we use $K = 3$ and $N = 3$ steps).

Algorithm 1 Conditional Diffusion Sampling

Require: Observation y, Time-grid $0 = t_0 < t_1 < \cdots < t_N = T$,

1: Sample $x_{t_N} \sim \mathcal{N}(0, \boldsymbol{I})$ ▷ Initialize reversed diffusion
2: **for** $n = N, \ldots, 1$ **do**
3: Set $\hat{x}_0 \leftarrow \tilde{x}^{\Delta_\theta}_{t_n,K}(x_{t_n}, y)$ using L_ϑ (see Figure 2) ▷ Unfolded sample targeting $p_0(x_0 \,|\, x_{t_n}, y)$
4: Sample $x_{t_{n-1}} \sim p_{t_{n-1}}(x_{t_{n-1}} \,|\, \hat{x}_0, x_{t_n})$ ▷ Reverse DDIM step
5: **end for**
6: **return** x_{t_0}

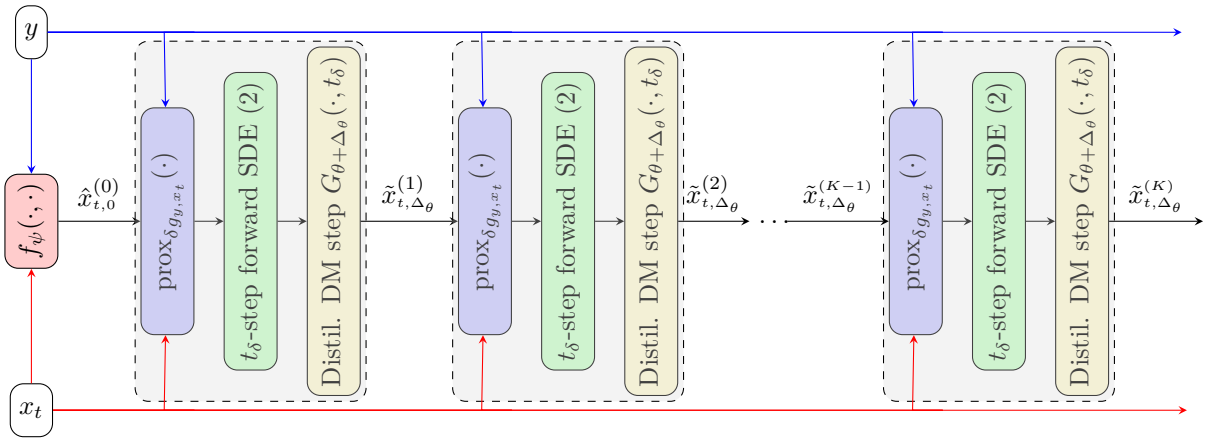


Figure 2: Diagram of the proposed conditional sampling architecture derived by deep unfolding K LATINO iterations (Spangnoletti et al., 2025). The prior is introduced via a pre-trained unconditional DM G_θ with LoRA adaptation Δ_θ, while the observation model, measurement y and noisy state x_t are involved via $g_{y,x_t} = -\log p(y, x_t|x_0)$. The first module is initialized by an estimator f_ψ such as RAM (Terris et al., 2025), or by setting $f_\psi(x_t, y) = A^\dagger y$. The unfolded network is finetuned and distilled to sample from $(\mathbf{x}_0|y, x_t)$.

3.2 Training objective

To train our network to sample the conditional random variable $(\mathbf{x}_0|y, \mathbf{x}_t)$, we take inspiration from Kim et al. (2024); Bendel et al. (2023) and combine reconstruction and adversarial losses, which has been shown to greatly improve for sample generation quality as opposed to using a reconstruction loss alone. More precisely, we adapt the objective of Kim et al. (2024) and use the following objective

$$\arg\min_{\Delta_\theta,\psi} \quad \mathcal{L}^G(\Delta_\theta, \psi) \triangleq \mathcal{L}_{\mathrm{Adv}}(\Delta_\theta, \psi, \phi) + \omega_{\ell_2}\mathcal{L}_{\ell_2}(\Delta_\theta, \psi) + \omega_{\mathrm{PS}}\mathcal{L}_{\mathrm{PS}}(\Delta_\theta, \psi)\,, \tag{8}$$

$$\arg\max_{\phi} \quad \mathcal{L}^D(\phi) \triangleq \mathcal{L}_{\mathrm{Adv}}(\Delta_\theta, \vartheta, \phi) + \omega_{\mathrm{GS}}\mathcal{L}_{\mathrm{GS}}(\phi)\,, \tag{9}$$

where $\omega_{\ell_2}, \omega_{\mathrm{PS}}, \omega_{\mathrm{GS}} > 0$ are regularization hyper-parameters and the remaining loss terms are defined as follows, where we let t be uniformly distributed on $[1, T]$. First, $\mathcal{L}_{\ell_2}(\Delta_\theta, \vartheta)$ is the ℓ_2^2-regression loss commonly encountered in DSM on the pixel space,

$$\mathcal{L}_{\ell_2}(\Delta_\theta, \psi) = \mathbb{E}_{t,\mathbf{x}_t,\mathbf{y},\mathbf{x}_0}\Big[||\mathbf{x}_0 - L_{\Delta_\theta,\psi}(\mathbf{x}_t, \mathbf{y})||_2^2\Big]\,. \tag{10}$$

Second, $\mathcal{L}_{\mathrm{PS}}(\Delta_\theta, \vartheta)$ promotes perceptual quality via the Learned Perceptual Image Patch Similarity (LPIPS) loss (Zhang et al., 2018)

$$\mathcal{L}_{\mathrm{PS}}(\Delta_\theta, \psi) = \mathbb{E}_{t,\mathbf{x}_t,\mathbf{y},\mathbf{x}_0}\Big[\mathrm{LPIPS}(\mathbf{x}_0, L_{\Delta_\theta,\psi}(\mathbf{x}_t, \mathbf{y}))\Big]. \tag{11}$$

Third, the adversarial loss, $\mathcal{L}_{\mathrm{Adv}}$, is based on Goodfellow et al. (2014)

$$\mathcal{L}_{\mathrm{Adv}}(\Delta_\theta, \vartheta, \phi) = \mathbb{E}_{\mathrm{x}_0,\mathbf{y}}\Big[\log\big(\varsigma(D_\phi(\mathbf{x}_0; \mathbf{y}))\big)\Big] + \mathbb{E}_{t,\mathbf{x}_t,\mathbf{y}}\Big[\log\big(1 - \varsigma(D_\phi(L_{\Delta_\theta,\psi}(\mathbf{x}_t, \mathbf{y}), \mathbf{y}))\big)\Big] \tag{12}$$

where $\varsigma : \mathbb{R} \to \mathbb{R}$ is the sigmoid function. The discrimination D_ϕ is a CNN adapted from Radford et al. (2016); Bendel et al. (2023), which we train to maximize the discriminator loss $\mathcal{L}^D(\phi)$. To regularize the discriminator, we follow Mescheder et al. (2018) and introduce a gradient-sparsity regularizer designed to promote Lipschitz regularity of D_ϕ, defined by

$$\mathcal{L}_{\mathrm{GS}}(\phi) = \mathbb{E}_{t,\mathbf{x}_t,\mathbf{y},\mathbf{x}_0,\mathbf{u}}\Big[\Big\|\nabla_x D_\phi\big(\mathbf{u}\mathbf{x} + (1-\mathbf{u})L_{\Delta_\theta,\psi}(\mathbf{x}_t, \mathbf{y}), \mathbf{y}\big)\Big\|^2\Big], \tag{13}$$

where $\mathbf{u} \sim \mathcal{U}(0,1)$ is independent of $\mathbf{x}_0$ and $\mathbf{y}$. Note that our presentation of the training objective assumes RAM initialization, so the trainable parameters of L_ϑ are $\vartheta = (\Delta_\theta, \psi)$. If ψ is frozen or if the first LATINO module is initialized directly with $\boldsymbol{A}^\dagger y$ and x_t, then we use the same training objective reduced to $\vartheta = \Delta_\theta$.

As mentioned previously, using Equation 8 is closely related to training $L_{\Delta_\theta,\psi}$ as a consistency trajectory model (Kim et al., 2024) to sample from $p(x|y, x_t)$. Similarly to CMs, our distilled model is trained for sampling, not for denoising. However, CMs are deterministic whereas $L_{\Delta_\theta,\psi}$ is a stochastic map, and we do not seek to enforce consistency of paths as a result. Also, we use a discriminator D_ϕ that conditions on y only, whereas training a CM for $(\mathbf{x}_0|y, \mathbf{x}_t)$ by following Kim et al. (2024) would require using a discriminator that conditions on both y and x_t. This would also require a larger and more complex discriminator architecture with a time embedding, which we do not consider here, as we find that a lighter discriminator that conditions on y only suffices for correctly training the proposed UD^{2}M sampler.

Before concluding this section, we present two alternative interpretations of the proposed deep unfolded architecture. Both are based on deep unfolding of discretizations of the same overdamped Langevin process that underlies LATINO, specifically the PnP-ULA scheme introduced by Laumont et al. (2022). In the first interpretation, our architecture is viewed as a deep unfolding of PnP-ULA using a pre-trained SNORE denoiser (Renaud et al., 2024) adapted via LoRA. In the second, it is interpreted as a deep unfolding of the DM-within-ULA scheme proposed by Mbakam et al. (2024). In both cases, the likelihood is incorporated through its proximal operator (Durmus et al., 2018), rather than through the more conventional gradient-based approach. Among these interpretations, we find the deep unfolding of LATINO to be the most meaningful. LATINO is itself a distilled few-step scheme that incorporates the likelihood through its proximal operator and generalizes naturally to image-text latent DMs such as Stable Diffusion (Yin et al., 2024), which are seeing increasing adoption. Developing UD^{2}M samplers by finetuning latent DMs is a natural extension of this work and main direction for future research.

4 Experiments

4.1 Experimental setup.

Preliminaries. To demonstrate the effectiveness of the proposed method, we now report a series numerical experiments and comparisons with competing approaches from the state of the art. In our experiments, we consider three forms of image deblurring (Gaussian, uniform and motion deblurring), two forms of image inpainting (box and uniformly randomly missing pixels), image super-resolution (SR) by a factor 4, as well as the removal of artifacts from aggressive JPEG compression (as an example of a non-linear restoration problem). Moreover, we assess the performance of our method qualitatively and quantitatively by computing the following quantitative metrics: the Peak Signal-to-Noise Ratio (PSNR) as distortion metric (Wang et al., 2004), the Learned Perceptual Image Patch Similarity (LPIPS) for perceptual quality (Zhang et al., 2018), and the Frechet Inception Distance (FID) for sampling accuracy (Heusel et al., 2017).

Datasets and experimental conditions. We use the following two public datasets in our experiments - ImageNet (Russakovsky et al., 2015) and LSUN Bedroom (Yu et al., 2015) - which have been used extensively in prior work related to image restoration with DMs and CMs in particular. For both datasets, we crop images to a size of 256×256 pixels and normalize their pixel amplitude to the range of $[0, 1]$. For our experiments with the ImageNet dataset, we used one million images from the training set for model training, whereas for the LSUN Bedroom dataset, we used 1.2 million images for training. For computing performance metrics, we use a test set of 1500 images from ImageNet and a test set of 300 images from LSUN bedroom. Regarding the parameters of each experiment, for ImageNet restoration tasks we follow Chung et al. (2023), with the exception of motion deblurring where we follow Zhu et al. (2023). For LSUN bedroom restoration tasks, for random inpainting, Gaussian deblurring and SR ($\times 4$), we use the setup as Garber & Tirer (2025) with a noise-level $\sigma = 0.025$; we follow Zhao et al. (2025) for box inpainting. For more details about the considered experiments, please see Appendix B.3.

Comparisons with the state of the art We compare our proposed method with an extensive selection of zero-shot and fine-tuned diffusion models from the state of the art. For the experiments with ImageNet,

we compare with the diffusion bridges I^2SB (Liu et al., 2023a) and CDDB (Chung et al., 2023), which are trained specifically for posterior sampling for ImageNet and for particular tasks. In addition, we compare with the two zero-shot methods DiffPIR (Zhu et al., 2023) and DDRM (Kawar et al., 2022), which rely on the same pretrained DM for ImageNet as our method, but differ in how they incorporate the data fidelity term.

For the experiments with LSUN bedroom, we use the same methods mentioned previously and two additional ones that rely on a CM trained and distilled for this dataset: the zero-shot method CM4IR (Garber & Tirer, 2025), and CoSIGN (Zhao et al., 2025), a learning-based method which upgrades an unconditional CM into a conditional CM for posterior sampling by learning an operator-specific ControlNet guidance that is attached onto the CM (Zhao et al., 2025). Our experiments show that CoSIGN's ControlNet guidance is more computationally efficient than our unfolding approach, but lacks flexibility and leads to less accurate results as it does not use an explicit likelihood function.

Implementation of the proposed method We implement our method as follows. For our deep unfolded architecture, we use $K = 3$ LATINO modules with a pre-trained diffusion model denoiser[3]; we use the U-NET architecture of Ho et al. (2020) with time-embeddings and attention layers trained on a linear noise schedule $\beta_1 = 10^{-4}$ up to $\beta_{1000} = 0.02$. With regards to training, we use the training objective detailed in Section 3.2 and LoRA adaptation with rank 5, while keeping original weights θ frozen. Moreover, for each experiment, with the exception of JPEG artifact removal, we implement and train two versions of our model. In one version, the first LATINO module is initialized directly with a weighted average of the inputs $A^\dagger y$ and x_t. In the other, the first LATINO module is initialized by a RAM module (Terris et al., 2025) with stacked operators $(\boldsymbol{I}, \boldsymbol{A})$ that take x_t and y as input, and which we train end-to-end together with he LoRA adaptation Δ_θ by using the objective of Section 3.2. We do not consider RAM initialization for the JPEG artifact removal task because RAM is not suitable for non-linear image restoration problems. For more details about the model's architecture, training parameters, GPU and time consumption, please see Appendix B.

Extension to non-linear inverse problems Similarly to Spangnoletti et al. (2025), to apply our method to JPEG artifact restoration and other non-linear image restoration problems we solve $\mathrm{prox}_{\delta g_y}(s) = \arg\min_{x_0 \in \mathbb{R}^d} g_y(x_0) + \frac{1}{2\delta}\|x_0 - s\|_2^2/2$ with $g_{y,x_t} = -\log p(y, x_t|x_0)$ approximately by using an optimizer -we use Adam (Kingma & Ba, 2014) with s as warm-starting. We do not use RAM initialization.

4.2 Results

Table 1 and Table 2 below summarize the performance metrics for our proposed method, with (w/) and without (wo) RAM initialization, for the considered ImageNet and LSUN Bedroom experiments respectively. For comparison, we also report the performance metrics for the alternative approaches from the state of the art. The results for the ImageNet motion deblurring task are reported separately in Table 7, as there are no publicly available finetuned models for posterior sampling for that task for comparison. As mentioned previously, we do not use RAM initialization for JPEG artifact removal because RAM does not support non-linear problems, and we do not use RAM initialization for box inpainting because it is a lightweight model with a receptive field that is not sufficiently large to meaningfully inpaint large regions.

We observe from Table 1 and Table 2 that the proposed approach performs very strongly relative to the state of the art across all tasks and all metrics, especially when RAM initialization is used. Notably, our approach very clearly outperforms the alternative methods from the state-of-the-art in terms of FID and LPIPS scores, indicating excellent perceptual and sampling quality. For example, for the SR ($\times 4$) task, our method with RAM initialization achieves a FID score of 11.9 for ImageNet and 19.48 for LSUN Bedrooms, largely outperforming the second best methods (CDDB and CoSIGN respectively) which achieve scores of 19.88 and 40.48. In addition, the proposed approach is highly computationally efficient, as it only requires in the order of 10 NFEs per posterior sample, a key computational advantage relative to non-distilled methods. This illustrates the benefit of training with the CM objective of Equation 8 that induces distillation. Having said that, out method is not as efficient as CoSIGN, which does not have the cost of evaluating $K = 3$ unfolded LATINO modules per step of the multi-step scheme.

[3]For ImageNet, we use the un-(class)conditional model `https://github.com/openai/guided-diffusion/` and for LSUN bedroom the model `https://github.com/pesser/pytorch_diffusion/`

Table 1: Quantitative results for the deblurring and super-resolution tasks. PSNR [dB] (↑), LPIPS (↓) and FID (↓) results on ImageNet dataset with noise level 0.01.

		Deblurring						SR			JPEG		
		Gaussian			Uniform			×4			QF=10		
Methods	NFEs	PSNR	LPIPS	FID	PSNR	LPIPS	FID	PSNR	LPIPS	FID	PSNR	LPIPS	FID
Ours (wo RAM)	9	**38.77**	0.02	4.61	35.57	0.02	11.14	24.42	0.15	20.69	**27.52**	**0.18**	35.16
Ours (w/ RAM)	12	35.97	**0.01**	**3.30**	**36.96**	**0.01**	**2.69**	**26.70**	**0.08**	**11.9**	-	-	-
CDDB	1000	37.02	0.06	5.01	31.26	0.19	23.15	26.41	0.2	19.88	26.34	0.26	**19.48**
I2SB	1000	36.01	0.07	5.8	30.75	0.2	23.01	25.22	0.26	24.13	26.12	0.27	20.35
DiffPIR	100	28.10	0.13	21.53	31.44	0.10	20.20	20.39	0.36	70.45	-	-	-
DDRM	20	36.73	0.07	4.34	29.21	0.21	19.97	26.05	0.27	46.49	26.33	0.33	47.02

Table 2: Deblurring, inpainting and super-resolution. PSNR [dB] (↑), LPIPS (↓) and FID (↓) results on LSUN bederoom dataset with noise level 0.025. Comparisons with I^2SB and, CDDB and CoSIGN on the Box inpainting and SR (×4) problems are taken from Zhao et al. (2025). The remaining comparisons are obtained from Garber & Tirer (2025).

		Inpainting						Deblurring			SR		
		Box			Random			Gaussian			×4		
Methods	NFEs	PSNR	LPIPS	FID	PSNR	LPIPS	FID	PSNR	LPIPS	FID	PSNR	LPIPS	FID
Ours (wo RAM)	9[a]	22.22	**0.11**	**8.45**	22.32	0.19	43.3	26.73	0.26	88.97	24.51	0.19	32.71
Ours (w/ RAM)	12	-	-	-	**27.75**	**0.06**	**8.61**	**29.67**	**0.08**	**13.23**	25.48	**0.12**	**19.48**
CoSIGN	2	22.61	0.14	38.64	23.22	0.37	-	19.74	0.342	-	26.13	0.22	40.84
CDDB	1000	**23.74**	0.13	45.20	-	-	-	-	-	-	**27.31**	0.24	54.20
I^2SB	1000	23.21	0.28	55.10	-	-	-	-	-	-	27.23	0.24	53.40
DiffPIR	100	13.42	0.34	66.79	23.8	0.36	97.69	27.48	0.32	64.81	23.83	0.45	101.92
DDRM	20	18.90	0.22	51.50	19.16	0.55	-	28.94	0.22	-	25.09	0.37	-
CM4IR	4	21.98	0.24	42.83	25.28	0.33	159.43	29.00	0.21	53.19	26.14	0.30	124.22

[a]We have implemented our method with $K = 3$ and $N = 3$ (so 9 NFEs) for all experiments, except for box inpainting where we have used $N = 5$ instead of $N = 3$ (so 20 NFEs), as this is a more challenging task.

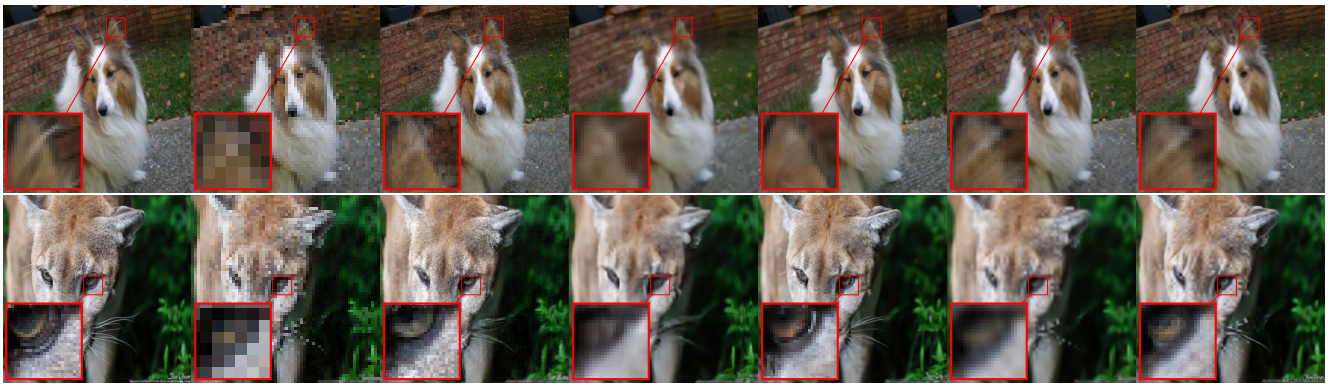
For illustration and qualitative evaluation, Figure 3, Figure 4, and Figure 5 depict examples of posterior samples for the ImageNet SR (×4), Gaussian deblurring and JPEG restoration tasks, respectively. We observe that our proposed approach (w/ RAM initialization) delivers sharp posterior samples that recover a significant amount of the fine detail in the ground truth images, and without noticeable noise or color bias. With regards to the LSUN experiments, Figure 6 and Figure 7 present results for the SR (×4) and box inpainting tasks, respectively. Again, we observe that our proposed approach (w/ RAM initialization) delivers sharp posterior samples with fine detail and no noticeable noise or color bias, in agreement with the strong performance metrics reported in Table 2.

4.3 Ablation Studies

We now proceed to examine and quantify the role of different key elements of our proposed method. This is achieved by appropriately modifying or deactivating these elements and quantifying changes in performance.

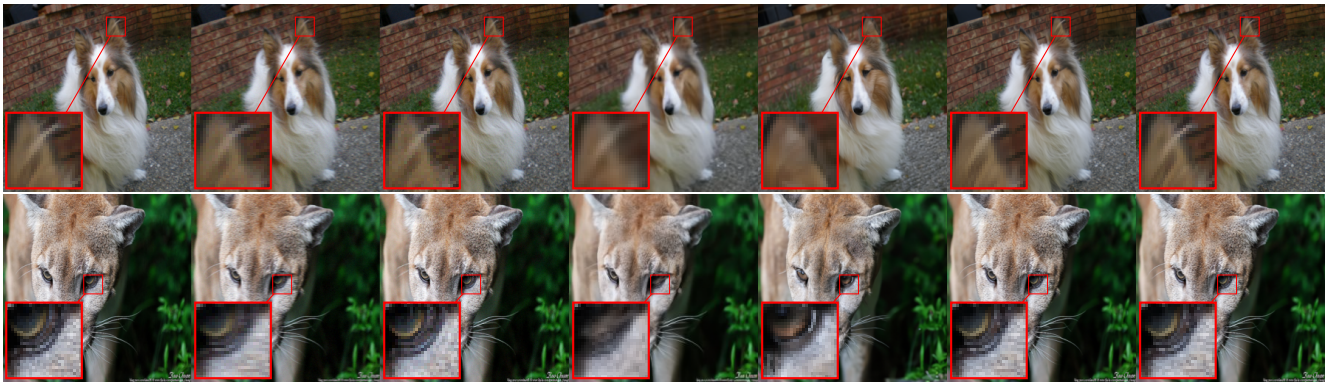
4.3.1 Choice of the number of sampling steps N

First we assess the effect of modifying the number of sampling steps N in the multi-step scheme. Note that N is specified during test time, so modifying it does not requires retraining the model. Table 3(a) shows the performance metrics for $N \in \{1, 3, 9, 27\}$ for the LSUN SR (×4) model with $K = 3$ LATINO modules and RAM initialization (recall that in our previous SR (×4) experiments we have used $K = 3$ and $N = 3$). We observe in Table 3(a) that the performance metrics do not change significantly with variations of N, indicating that the proposed approach is robust to different choices of N. Low values of N lead to some small amount of estimation bias towards the mean, and hence to a small increase in PNSR at the expense of LPIPS



Ground Truth Degraded DiffPIR DDRM CDDB Ours (wo RAM) Ours (w/ RAM)

Figure 3: Comparison of posterior samples for the task SR (×4) with noise level $\sigma = 0.01$ on ImageNet 256.



Ground Truth Degraded DiffPIR DDRM CDDB Ours (wo RAM) Ours (w/ RAM)

Figure 4: Comparison of posterior samples for the task Gaussian deblurring with noise level $\sigma = 0.01$ on ImageNet 256.

and FID performance, whereas larger values of N lead to a mild decrease in overall performance due to accumulation of errors (see Kim et al. (2023) for details) as well as to an increase in computational cost from additional NFEs. As mentioned previously, in our experiments we used $N = 3$, which we found to reliably provide good performance with a low computational cost across all the considered tasks (with the exception of box inpainting, when we use $N = 5$ as the task is challenging and we don't use RAM initialization).

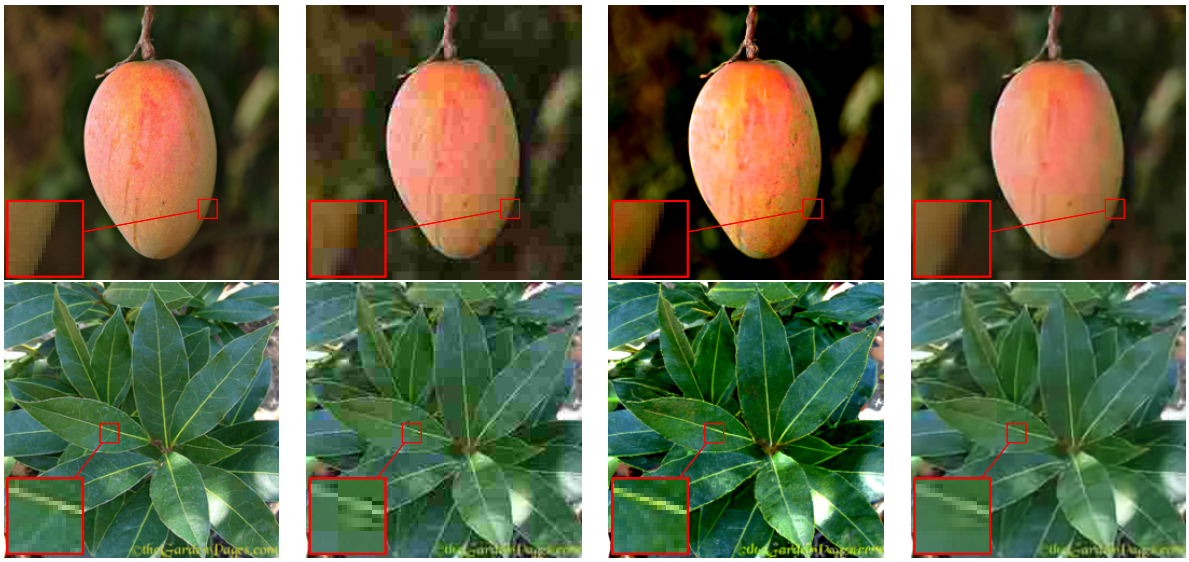
Table 3: Reconstruction metrics for varying number of conditional diffusion steps (N) and unfolded iterations (K).

N	PSNR ↑	LPIPS ↓	FID ↓	NFEs ↓
1	25.93	0.11	14.07	4
3	25.88	0.10	13.15	12
9	25.54	0.10	15.10	36
27	25.02	0.11	17.47	108

(a) K=3

N	PSNR ↑	LPIPS ↓	FID ↓	NFEs ↓
1	25.60	0.11	17.95	2
3	25.61	0.11	16.59	6
9	24.72	0.12	20.42	18
27	22.82	0.12	21.93	54

(b) K=1



Ground Truth Degraded CDDB Ours (wo RAM)

Figure 5: Comparison of posterior samples for the task JPEG artifact removal (QF=10) with noise level $\sigma = 0.01$ on ImageNet 256.

Ground Truth Degraded CM4IR CoSIGN Ours (wo RAM) Ours (w/ RAM)

Figure 6: Comparison of posterior samples for the task SR ($\times 4$) with noise level $\sigma = 0.025$ on LSUN Bedroom.

4.3.2 Effect of unfolding

We now assess the benefit of unfolding multiple LATINO modules per sampling step, as opposed to using a single LATINO module, with the same or possibly a larger number of sampling steps. Table 3(b) reports the performance metrics when retraining the network with a single LATINO module ($K = 1$) and RAM initialization. By comparing Table 3(b) with Table 3(a), we observe a noticeable deterioration in performance sampling quality as measured by FID, even if the reduction the number of LATINO modules per step is compensated by additional sampling steps. We conclude that there is a clear benefit in distilling conditional DM architectures from deep unfolding, which is consistent with the literature on deep unfolding of optimization algorithms (Monga et al., 2021). It is also worth mentioning that modifying K without retraining the DM leads to severe visual artifacts (results not reported here), so it is not possible to modify K during inference.

Ground Truth | Degraded | DDRM | CDDB | CoSIGN | Ours (wo RAM)

Figure 7: Comparison of posterior samples for the task box inpainting with noise level $\sigma = 0.025$ on LSUN Bedroom.

4.3.3 The choice of the rank of the LoRA adaptation Δ_θ

We now assess the effect of the LoRA fine-tuning rank parameter. As explained previously, rather than fine-tuning the full model, we use a LoRA approach whereby we train a low-rank correction Δ_θ that is added to the original model weights θ, which remain frozen. The number of trainable parameters in Δ_θ is directly related to the choice of the rank of Δ_θ. Setting the rank too low deteriorates performance because it constrains the model too much, whereas setting the rank too high increases the computational cost of fine-tuning the model and can lead to some overfitting. In order to illustrate this, we train again our model for ImageNet SR ($\times 4$) by setting the rank to 2 and to 20 (in our previous results we have used rank 5). The number of trainable parameters in Δ_θ is 165888 when the rank of Δ_θ is 2, 414720 when the rank is 5, and 1658880 when it is 20; this represents 0.03%, 0.08% and 0.3% of the total number of parameters in θ, respectively. The results for this study are summarized in Table 4. We observe that the model's performance, as measured by PSNR and LPIPS, is robust to large variations in the rank of Δ_θ. However, the rank of Δ_θ has a noticeable effect on FID performance, which drops significantly if it is set too low, indicating a clear deterioration in sampling quality (e.g., in Table 4 the FID for rank 2 is almost twice that of rank 5).

Table 4: Ablation study evaluating the impact of varying LoRA ranks on the ImageNet dataset for the super-resolution task.

rank(Δ_θ)	2	5	20
PSNR	26.05	26.70	26.32
LPIPS	0.11	0.08	0.09
FID	19.75	11.9	13.03
$\lvert\Delta_\theta\rvert$	165888	414720	1658880

4.3.4 Generalization w.r.t. to the noise variance σ^2

Models obtained by deep unfolding often generalize better to different levels of measurement noise, by comparison to other end-to-end training strategies that are not aware of the value of the noise variance σ^2 during test time. To demonstrate this capability, we follow (Garber & Tirer, 2025) and evaluate our LSUN bedroom SR ($\times 4$) model trained for $\sigma = 0.025$ (with $N = 3$ and $K = 3$) in a more challenging situation ($\sigma = 0.05$). The results are summarized in Table 5. For comparison, we also report the results obtained with the CoSIGN and CM4IR models applied to the same problem (results from (Garber & Tirer, 2025, Table 2)), where CoSIGN is also finetuned for $\sigma = 0.025$ and where CM4IR, as a zero-shot method, is provided the correct value of $\sigma = 0.05$ during test time. We observe that our proposed method is able to use the value of $\sigma = 0.05$ effectively during test time, even if training was performed using $\sigma = 0.025$, similarly to zero-shot methods like CM4IR. Conversely, CoSIGN struggles to generalize to the higher noise level and exhibits poor PSNR and LPIPS metrics as a result. Again, by comparison to CM4IR, we achieve significantly better perceptual quality as reflected by a lower LPIPS metric. For completeness, Figure 8 depicts examples of observations and samples for our method when $\sigma = 0.0125$, $\sigma = 0.025$ (in-distribution), and $\sigma = 0.05$.

Table 5: Reconstruction statistics for our model applied to the SR (×4) task on LSUN bedroom, with varying noise level in the observation space.

	Ours		CoSIGN		CM4IR	
Noise level σ	PSNR	LPIPS	PSNR	LPIPS	PSNR	LPIPS
0.025 (in-distribution)	25.12	0.178	26.10	0.205	26.14	0.295
0.05 (out-of-distribution)	24.11	0.214	20.35	0.509	25.60	0.320

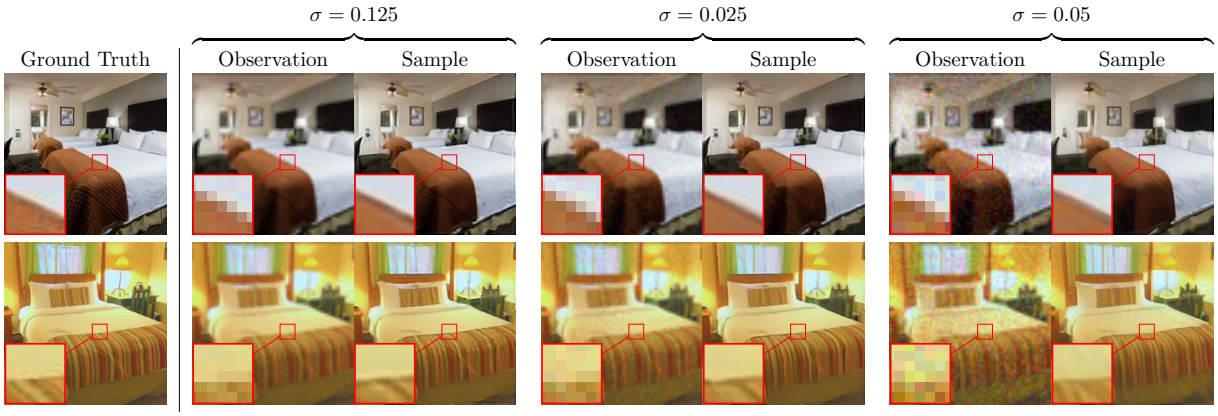


Figure 8: Comparison of posterior samples for the task SR (×4) on LSUN Bedroom, for three levels of noise.

4.3.5 Generalization w.r.t. to more challenging degradations

We now explore the capacity of our proposed approach to deal with more challenging forward models not seen during training. For this exploratory analysis, we apply our LSUN Bedroom SR (×4) model to a more challenging SR ×8 task (we keep $\sigma = 0.025$, as in training). Again, we report comparisons with the LSUN Bedroom SR ×4 CoSIGN model. Although both models are trained for SR (×4), our method receives the correct SR (×8) forward operator during inference. Conversely, CoSIGN does not have access to the SR (×8) forward operator, as it relies on a pre-trained ControlNet guidance. Through this exploratory experiment, we seek to show that models obtained by deep unfolding are more robust to challenging degradations (or forward operators) not encountered during training, by comparison to other end-to-end training strategies. A summary of these results is presented in Table 6 and an example in Figure 9. We observe that, while both methods struggle with this difficult restoration task, our method clearly outperforms CoSIGN which is unable to use the forward operator $\boldsymbol{A}$ during test time and exhibits strong reconstruction artifacts as a result. For completeness, Figure 9 also presents the result obtained with the CM4IR zero-shot method, which also outperforms CoSIGN by using the forward operator $\boldsymbol{A}$, but suffers from some smoothing because of the approximations involved in zero-shot inference.

4.3.6 Task specific versus universal image restoration models

In our previous experiments, we considered that $\boldsymbol{A}$ is unique and known a priori, or it is not unique but belongs to a known class of operators specific for the considered task (e.g., motion blurs). In principle,

Table 6: Reconstruction metrics computed on the LSUN Bedroom validation set using out-of-distribution forward operators for the model trained on LSUN Bedroom SR (×4).

	CoSIGN		Ours	
Task / Operator	PSNR ↑	LPIPS ↓	PSNR ↑	LPIPS ↓
SR (×8) - out-of-distribution	20.11	0.55	20.40	0.32
SR (×4) - in-distribution	26.13	0.22	25.40	0.12

Ground Thruth Degraded CoSIGN CM4IR Ours (w/ RAM)

Figure 9: Comparison of posterior samples for the task SR ($\times 8$), with noise level $\sigma = 0.025$, on LSUN Bedroom with models trained for LSUN Bedroom SR ($\times 4$), to illustrate the models' capacity to generalize to more challenging tasks.

one could also consider training a single common model for a much wider range of tasks. In particular, it would be interesting to train a "universal" Bayesian image restoration model, as considered in Terris et al. (2025) for MMSE estimation. Unfortunately, developing such general models requires using a much more sophisticated architecture than the one we consider here. For example, Terris et al. (2025) combines unfolding with several additional key elements not considered here (e.g., a multiscale decomposition of the operators, Krylov updates). Without these substantial improvements, our proposed approach suffers from a drop in performance when we try to generalize too widely. To illustrate this phenomenon, we trained a single posterior sampling model for the ImageNet dataset, simultaneously on Gaussian, uniform and motion deblurring, random inpainting and super-resolution tasks. The corresponding results are summarized in Table 7, where we also report the results obtained with our task specific models for comparison. We observe that the task specific models outperform the model trained jointly on all the considered tasks, and in all the considered performance metrics (the difference is most noticeable for FID performance). The extension of the UD^{2}Ms framework to support the development of general purpose posterior samplers that can tackle a wide range of image restoration tasks and a wide range of datasets, without retraining or compromising performance, is a main perspective for future work.

Table 7: Qualitative results of generalized models evaluated on the ImageNet dataset. The reported metrics include the FID score, average PSNR, and average LPIPS.

	Gaussian Deb			Motion Deb		Uniform Deb			Inp 70%		SR ($\times 4$)		
	Spec	Univ	DiffPIR	Spec	Univ	Spec	Univ	DiffPIR	Spec	Univ	Spec	Univ	DiffPIR
PSNR	35.97	34.80	28.1	33.40	31.10	36.96	31.35	31.44	31.15	30.16	26.70	25.74	20.39
LPIPS	0.01	0.03	0.13	0.04	0.04	0.01	0.08	0.1	0.03	0.06	0.08	0.12	0.36
FID	3.30	4.94	21.53	4.08	6.56	2.69	19.6	20.2	7.87	13.21	11.9	21.34	70.45

5 Conclusion

5.1 Concluding remarks and perspectives for future work

Diffusion models are powerful image priors for Bayesian computational imaging (Daras et al., 2024). The literature offers two main strategies to construct Bayesian imaging techniques based on diffusion models: Plug & Play methods that are zero-shot and hence highly flexible but reliant on approximations, and y-conditional diffusion models that achieve superior accuracy and speed through supervised specialization for a particular task. The framework introduced in this paper, unfolded and distilled diffusion models (UD^{2}Ms), leverages deep unfolding and distillation in order to upgrade a diffusion model representing the image prior into a few-step consistency model for posterior sampling. UD^{2}M samplers offer excellent accuracy and high computational efficiency, while retaining the flexibility required to generalize to variations in the forward operator and noise distribution during inference. A main novelty of UD^{2}M is to unfold a Markov chain Monte

Carlo algorithm, namely the recently proposed LATINO Langevin sampler (Spangnoletti et al., 2025), the first example of deep unfolding of a Monte Carlo scheme.

With regards to limitations of our proposed framework and perspectives for future work. We are cognizant that diffusion models can reproduce biases stemming from unbalanced or biased training data and distribution shift, diminishing the reliability of our framework. Future work should develop appropriate guardrails so that DM-based posterior samplers can be deployed safely and reliably. Also, we hope and anticipate that these issues will be partially mitigated by progress and democratization of deep generative modeling technology and the advancement of self-supervised deep generative models. Moreover, while our samplers are more computationally efficient than some alternatives, they still rely on very large models and therefore consume a significant amount of energy and compute resources. The development of more frugal architectures and sampling strategies is an important perspective for future work. Furthermore, as mentioned previously, it would be interesting to study architectures and training strategies to develop UD^{2}M samplers that can be applied to a wide range of image restoration tasks and datasets without the need for additional training, or in a few-shot manner, as is the case of the foundational image reconstruction model of Terris et al. (2025). In addition, as mentioned previously, many state-of-the-art DMs leverage the latent space representation of an autoencoder. For example, Stable Diffusion models (Yin et al., 2024) that allow semantic conditioning through text prompting. Extending UD^{2}M samplers to these DMs is a promising direction for future work.

Lastly, in parallel with this work, the concurrent works Wang et al. (2025); Elata et al. (2025) also seek to bridge supervised and zero-shot strategies for Bayesian imaging with diffusion models. Wang et al. (2025) relies on unfolding a conventional proximal gradient optimization scheme and uses an ℓ_2 loss in order to obtain a conditional diffusion model, without distillation. Elata et al. (2025) proposes a novel approach to incorporate the forward operator $\boldsymbol{A}$ and the measurement y directly within Transformer or UNet architectures commonly used for diffusion models. Contrasting and combining UD^{2}M with these alternative strategies is an important perspective for future work.

5.1.1 Broader impact statement

Computational imaging research generally poses some ethical dilemmas, especially when it involves significant methodological innovation. Although the innovations studied in the project do not raise any direct concerns, we are aware that the proposed Bayesian imaging framework could be adapted and transferred towards military applications by a hostile actor and for violating citizen privacy, for example. In addition, our framework relies on deep generative modeling technology that currently suffers from biases, which could lead to leading to unfair or discriminatory outcomes. We hope that these critical issues will be resolved by progress and democratization of generative modeling technology.

Moreover, the innovations presented in this paper fall within the "limited risk" level of the EU AI Act. We have followed the Act's recommendations for this level; i.e., a focus on the full transparency of the training data and the methods used. This also links in with our open research philosophy - we will release publicly available open-source implementations for research codes, with clear documentation and demonstrations. We see this as an essential requirement for conducting research in a democratic knowledge society.

References

Matthew Bendel, Rizwan Ahmad, and Philip Schniter. A regularized conditional GAN for posterior sampling in image recovery problems. *Advances in neural information processing systems*, 36:68673–68684, 2023.

Antonin Chambolle and Thomas Pock. An introduction to continuous optimization for imaging. *Acta Numerica*, 25:161–319, 2016.

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.

Hyungjin Chung, Jeongsol Kim, and Jong Chul Ye. Direct diffusion bridge using data consistency for inverse problems. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=497CevPdOg.

Florentin Coeurdoux, Nicolas Dobigeon, and Pierre Chainais. Normalizing flow sampling with Langevin dynamics in the latent space. *Machine Learning*, 113(11):8301–8326, 2024a. doi: 10.1007/s10994-024-06623-x. URL https://doi.org/10.1007/s10994-024-06623-x.

Florentin Coeurdoux, Nicolas Dobigeon, and Pierre Chainais. Plug-and-play split Gibbs sampler: Embedding deep generative priors in Bayesian inference. *IEEE Transactions on Image Processing*, 2024b.

Giannis Daras, Hyungjin Chung, Chieh-Hsin Lai, Yuki Mitsufuji, Jong Chul Ye, Peyman Milanfar, Alexandros G. Dimakis, and Mauricio Delbracio. A survey on diffusion models for inverse problems, 2024. URL https://arxiv.org/abs/2410.00083.

Mauricio Delbracio and Peyman Milanfar. Inversion by direct iteration: An alternative to denoising diffusion for image restoration. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=VmyFF5lL3F. Featured Certification, Outstanding Certification.

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

Alain Durmus, Eric Moulines, and Marcelo Pereyra. Efficient Bayesian computation by proximal Markov chain Monte Carlo: When Langevin meets Moreau. *SIAM Journal on Imaging Sciences*, 11(1):473–506, 2018.

Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106 (496):1602–1614, 2011.

Noam Elata, Hyungjin Chung, Jong Chul Ye, Tomer Michaeli, and Michael Elad. Invfussion: Bridging supervised and zero-shot diffusion for inverse problems, 2025. URL https://arxiv.org/abs/2504.01689.

Tomer Garber and Tom Tirer. Zero-shot image restoration using few-step guidance of consistency models (and beyond). *arXiv preprint arXiv:2412.20596*, 2025.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information processing systems*, 27, 2014.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. *Advances in neural information processing systems*, 10, 2017.

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

Matthew Holden, Marcelo Pereyra, and Konstantinos C. Zygalakis. Bayesian imaging with data-driven priors encoded by neural networks. *SIAM Journal on Imaging Sciences*, 15(2):892–924, 2022. doi: 10.1137/21M1406313. URL https://doi.org/10.1137/21M1406313.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Jari Kaipio and E Somersalo. *Statistical and computational inverse problems*. Applied Mathematical Sciences. Springer, New York, NY, October 2010.

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, Marcelo Pereyra, and Jean-Francois Giovannelli. A stochastic optimisation unadjusted Langevin method for empirical Bayesian estimation in semi-blind image deblurring problems. *SIAM Journal on Imaging Sciences*. to appear.

Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ode trajectory of diffusion. *arXiv preprint arXiv:2310.02279*, 2023.

Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ODE trajectory of diffusion. In *12th International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=ymjI8feDTD`.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=PqvMRDCJT9t`.

G.H. Liu, A. Vahdat, D.A. Huang, E. Theodorou, W. Nie, and A. Anandkumar. I2SB: Image-to-image Schrödinger bridge. In *40th International Conference on Machine Learning (ICML)*, 2023a.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023b. URL `https://openreview.net/forum?id=XVjTT1nw5z`.

Eric Luhman and Troy Luhman. Knowledge distillation in iterative generative models for improved sampling speed, 2021.

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, 2023. URL `https://arxiv.org/abs/2311.05556`.

Pierluigi Maponi. The solution of linear systems by using the Sherman–Morrison formula. *Linear algebra and its applications*, 420(2-3):276–294, 2007.

Charlesquin Kemajou Mbakam, Jean-Francois Giovannelli, and Marcelo Pereyra. Empirical Bayesian image restoration by Langevin sampling with a denoising diffusion implicit prior, 2024. URL `https://arxiv.org/abs/2409.04384`.

S Melidonis, M Holden, Y Altmann, M Pereyra, and K C Zygalakis. Empirical Bayesian imaging with large-scale push-forward generative priors. *IEEE Signal Process. Lett.*, 31:631–635, 2024.

Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14297–14306, 2023. doi: 10.1109/CVPR52729.2023.01374.

Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for GANs do actually converge? In *International conference on machine learning*, pp. 3481–3490. PMLR, 2018.

Vishal Monga, Yuelong Li, and Yonina C. Eldar. Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *IEEE Signal Processing Magazine*, 38:18–44, 3 2021. ISSN 15580792. doi: 10.1109/MSP.2020.3016905.

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.

Subhadip Mukherjee, Andreas Hauptmann, Ozan Öktem, Marcelo Pereyra, and Carola-Bibiane Schönlieb. Learned reconstruction methods with convergence guarantees: A survey of concepts and applications. *IEEE Signal Processing Magazine*, 40(1):164–182, 2023. doi: 10.1109/MSP.2022.3207451.

Gregory Ongie, Ajil Jalal, Christopher A. Metzler, Richard G. Baraniuk, Alexandros G. Dimakis, and Rebecca Willett. Deep learning techniques for inverse problems in imaging. *IEEE Journal on Selected Areas in Information Theory*, 1(1):39–56, 2020. doi: 10.1109/JSAIT.2020.2991563.

Marcelo Pereyra, Luis A. Vargas-Mieles, and Konstantinos C. Zygalakis. The split Gibbs sampler revisited: Improvements to its algorithmic structure and augmented target distribution. *SIAM Journal on Imaging Sciences*, 16(4):2040–2071, 2023. doi: 10.1137/22M1506122. URL https://doi.org/10.1137/22M1506122.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *Proceedings of the International Conference on Learning Representations*, 2016.

Marien Renaud, Jean Prost, Arthur Leclaire, and Nicolas Papadakis. Plug-and-play image restoration with stochastic denoising regularization. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Christian P. Robert. *The Bayesian choice: From decision-theoretic foundations to computational implementation*, volume Second edition. Springer, 2007.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *The Tenth International Conference on Learning Representations*, 2022.

C. J. Schuler, M. Hirsch, S. Harmeling, and B. Schölkopf. Learning to deblur. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(7):1439–1451, July 2016. doi: 10.1109/TPAMI.2015.2481418.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising Diffusion Implicit Models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=St1giarCHLP.

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

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. 2023b.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32211–32252. PMLR, 23–29 Jul 2023c.

A. Spangnoletti, J. Prost, A. Amansa, N. Papadakis, and M. Pereyra. LATINO-PRO: LAtent consisTency INverse sOlver with PRompt Optimization. *arXiv preprint (arXiv:2503.12615)*, 2025.

Matthieu Terris, Samuel Hurault, Maxime Song, and Julian Tachella. Reconstruct anything model: A lightweight foundation model for computational imaging. *arXiv preprint arXiv:2503.08915*, 2025.

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.

Yuanhao Wang, Shirin Shoushtari, and Ulugbek S. Kamilov. Diff-unfolding: A model-based score learning framework for inverse problems, 2025. URL `https://arxiv.org/abs/2505.11393`.

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

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T Freeman. Improved distribution matching distillation for fast image synthesis. In *NeurIPS*, 2024.

Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

Kai 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 (10):6360–6376, 2021.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018.

Jiankun Zhao, Bowen Song, and Liyue Shen. CoSIGN: Few-step guidance of consistency model to solve general inverse problems. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision – ECCV 2024*, pp. 108–126. Springer Nature Switzerland, 2025.

Ningning Zhao, Qi Wei, Adrian Basarab, Nicolas Dobigeon, Denis Kouamé, and Jean-Yves Tourneret. Fast Single Image Super-Resolution Using a New Analytical Solution for $\ell_2 - -\ell_2$ Problems. *IEEE Transactions on Image Processing*, 25(8):3683–3697, 2016.

Yuanzhi Zhu, Kai Zhang, Jingyun Liang, Jiezhang Cao, Bihan Wen, Radu Timofte, and Luc Van Gool. Denoising Diffusion Models for Plug-and-Play Image Restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1219–1229, 2023.

A Solving the proximal step in Figure 2

This section outlined the approach for solving the proximal step in Figure 2 tailored to each image restoration task in our study.

Deblurring The degradation model for deblurring is generally express as $y = \boldsymbol{A}x + \boldsymbol{n}$, where y is the observation, $\boldsymbol{n}$ is zero mean Gaussian noise with variance σ^2. The forward operator $\boldsymbol{A}$ represents a circular block matrix, and can be transformed as $\boldsymbol{A} = \mathcal{F}^\dagger \Sigma_{\boldsymbol{A}} \mathcal{F}$, where $\mathcal{F}$ and $\mathcal{F}^\dagger$ denote the fast Fourier transform (FFT) and its inverse, respectively, while $\Sigma_{\boldsymbol{A}}$ is a diagonal matrix containing the Fourier coefficients of the blurring kernel. A fast solution of the proximal step in Figure 2 is given by

$$\hat{x}_{t,k} = \mathcal{F}^\dagger \left(\left[\frac{\Sigma_{\boldsymbol{A}}^\top \mathcal{F} y}{\sigma_y^2} + \frac{\mathcal{F} x_t}{\sqrt{\overline{\alpha}_t}\sigma_t^2} + \frac{\mathcal{F}\tilde{x}_{t,k}^{\Delta_\theta}}{\delta} \right] \Big/ \left[\frac{\Sigma_{\boldsymbol{A}}^2}{\sigma^2} + \frac{\boldsymbol{I}}{\sigma_t^2} + \frac{1}{\delta} \right] \right). \tag{14}$$

Inpainting For inpainting task, the degradation process is formulated as $y = \boldsymbol{A} \circ x + \boldsymbol{n}$, where $\boldsymbol{A}$ denotes the mask operator and $\circ$ represents the elementwise multiplication operator. By using the well known Sherman–Morrison formula (Maponi, 2007), we obtain the following fast solution for the proximal step in Figure 2

$$\hat{x}_{t,k} = \kappa_k \left(\boldsymbol{I} - \kappa_k \frac{\boldsymbol{A}}{\sigma^3 + \kappa_k} \right) \circ \left(\frac{\boldsymbol{A}^\top y}{\sigma^2} + \frac{\tilde{x}_{t,k}^{\Delta_\theta}}{\delta} + \frac{x_t}{\sqrt{\overline{\alpha}_t}\sigma_t^2} \right), \tag{15}$$

where $\kappa_k = \frac{\delta \sigma_t^2}{\delta + \sigma_t^2}$ and $\circ$ denotes the elementwise multiplication operator.

Super-resolution In super-resolution task, we formulate the problem by assuming that the low-resolution image is a blurred, downsampled, and noisy version of the high-resolution image. Mathematically, the model is given by $y = \boldsymbol{S}_s \boldsymbol{A} x + \boldsymbol{n}$, where $\boldsymbol{S}_s$ represents the downsampler operator with factor s and $\boldsymbol{A}$ represents the circular blur operator. According to Zhao et al. (2016); Zhang et al. (2021) a fast solution for the proximal step in Figure 2 is given by

$$\hat{x}_{t,k} = \mathcal{F}^\dagger \left(\frac{1}{\kappa_k} \big(\boldsymbol{r} - (\Sigma_{\boldsymbol{A}}^\top) \downarrow_s \circ_s (\frac{\Sigma_{\boldsymbol{A}} \boldsymbol{r}) \downarrow_s}{(\Sigma_{\boldsymbol{A}}^2) \downarrow_s + \kappa_k}) \right), \tag{16}$$

where $\boldsymbol{r} = \Sigma_{\boldsymbol{A}}^\top \boldsymbol{S}^\top y + \kappa_k \left(\delta x_t + \sqrt{\overline{\alpha}_t}\sigma_t^2 \tilde{x}_{t,k}^{\Delta_\theta} \right)$, $\kappa_k = \frac{\sigma^2}{\overline{\alpha}_t \sigma_t^2 \delta}$ and $\circ_s$ represents distinct block processing operator with element-wise multiplication. The operator $\downarrow_s$ denotes distinct block downsampler. For more details, see Zhao et al. (2016); Zhang et al. (2021).

Extension to non-linear tasks: JPEG For the non-linear JPEG restoration task, the observation model is given by $y = \mathcal{A}(x) + \boldsymbol{n}$, where $\mathcal{A}$ denotes the non-linear compression operator and $\boldsymbol{n}$ is the zero-mean Gaussian noise with variance σ^2. In this case, computing the exact solution of the proximal step $\text{prox}_{\delta g_y}(s) = \arg\min_{x_0 \in \mathbb{R}^d} g_y(x_0) + \frac{1}{2\delta}\|x_0 - s\|_2^2/2$ with $g_{y,x_t} = -\log p(y, x_t|x_0)$ is not feasible. To address this challenge, we approximate the solution by using Adam optimizer (Kingma & Ba, 2014) with s as warm-starting and we do not use RAM initialization.

B Implementation details

B.1 Models

In this work, we used two distinct pre-trained models for our experiments. The first model is derived from ADM (Dhariwal & Nichol, 2021), originally pre-trained on the ImageNet dataset at the resolution of 256×256. The second model is based on a pre-trained model from DDPM (Ho et al., 2020), which was pre-trained on LSUN dataset at the resolution of 256×256. The architectures of these models differ significantly and a detailed comparison of their architecture, parameter counts and computational requirements is provided in Table 8.

Regarding LoRA configuration, we implemented exclusively on the attention layers of the pre-trained model to improved its adaptability for specific image restoration tasks while reducing the computational overhead.

Table 8: Models summarize: $|\theta|$ denotes the total number of frozen parameters, $|\Delta_\theta|$ indicates the total LoRA parameters and $|\psi|$ represents the total RAM parameters. **Model 1** refers to the ADM pre-trained on the ImageNet dataset, while **Model 2** denotes the DDPM pre-trained on the LSUN dataset.

Models	rank(Δ_θ)	#Attn layers	$\lvert\theta\rvert$	$\lvert\Delta_\theta\rvert$
Model 1	-	16	552814086	0
Model 2	-	5	113673219	0
Model 1 w/ LoRA	5	16	552814086	414720
Model 2 w/ LoRA	5	5	113673219	122880
RAM	-	0	35618953	0

B.2 Training procedure

We adopted a training procedure that is carefully designed to ensure robust performance across tasks such as deblurring, inpainting and super-resolution. The training objective integrates reconstruction and adversarial losses to enhance sample generation quality. For the reconstruction loss, we used a consistency term to enforce data fidelity and a LPIPS term to enhance perceptual quality. Conversely, the adversarial term is designed to promote the model to generate realistic sample. Additionally, the training loss incorporates gradient penalty term to stabilize training by enforcing Lipschitz continuity on the discriminator. To facilitate reproducibility, comprehensive details regarding batch size (bs), learning rate (lr), optimizer and specific weights for loss term are summarized in Table 9.

Table 9: **Model 1** refers to the ADM pre-trained on the ImageNet dataset, while **Model 2** denotes the DDPM pre-trained on the LSUN dataset.

	lr	bs	optimizer	weight decay	ω_{ℓ_2}	ω_{GP}	ω_{GS}
Model 1	$1e^{-4}$	2	AdamW	0.01	1	0.1	0.01
Model 2	$1e^{-4}$	4	AdamW	0.01	1	0.1	0.01
RAM$_\psi$	$1e^{-5}$	-	AdamW	0.01	-	-	-

B.3 Setting experiments

Box inpainting We consider a fixed and centered square mask of size 128×128 pixels, $K = 5$ and $\sigma = 0.025$.

Random inpainting In this experiment, a random binary mask is applied with missing pixel rates of 80% and 70%. We use $K = 3$ iterations for algorithm unrolling and a noise level of $\sigma = 0.025$.

Super-resolution: For SR experiments with a scaling factor m, images from the LSUN dataset are downsampled by convolving with a bicubic filter of size $m^2 \times m^2$. In contrast, for the ImageNet dataset, images are first convolved with a 3×3 Gaussian kernel[4] and bandwidth of 10, followed by a downsampling operation with factor m. Unless otherwise stated, we consider downsampling factor $m = 4$. The number of iterations in algorithm unrolling is set to $K = 3$, and experiments are conducted with noise levels $\sigma \in \{0.01, 0.025\}$.

[4]The Gaussian kernel act as a low-pass filter to reduce aliasing before downsampling.

Deblurring For this experiment, consider 3 types of blur.

- **Gaussian blur**: On Imagenet experiments, we consider kernel sizes of 5×5 and 25×25 pixels and a bandwidth of 10. For the LSUN bedroom dataset, we use consider the anisotropic Gaussian blur used in Garber & Tirer (2025); Kawar et al. (2022) of size 9×9 and with bandwidth (20, 1).
- **Motion blur**: We generate random motion blur kernels with kernel sizes of 5×5 using the approach described in Schuler et al. (2016)[5],
- **Uniform blur** with kernel sizes of 9×9.

We conduct experiments under noise levels $\sigma \in \{0.01, 0.025, 0.05\}$ and using $K = 3$ iterations for algorithm unrolling.

JPEG restoration For this experiment, we applied JPEG compression with a quality factor of **QF=10** and introduced Gaussian noise with variance 0.01^2.

B.4 Memory and time consumption

Table 10 presents a comparative analysis of memory consumption and computational time for the models evaluated in this work. The results demonstrate that fine-tuning pre-trained models using LoRA technique reduces GPU memory usage and inference time. The table also includes the memory usage and inference time for RAM.

For the JPEG restoration task, we obtained an inference time of **4.01** seconds. This increased computational time is because the proposed method requires a sub-iterative method to solve the proximal step in Figure 2.

Table 10: Models summarize: K denotes the number of unfolding steps. **Model 1** refers to the ADM pre-trained on the ImageNet dataset, while **Model 2** denotes the DDPM pre-trained on the LSUN dataset.

Models	K	GPU (GB)	Time (s)	Resolution
Model 1	-	16.49	-	256^2
Model 2	-	3.39	-	256^2
Model 1 w/ LoRA	3	0.013	0.89	256^2
Model 2 w/ LoRA	3	0.004	0.53	256^2
RAM	1	1.06	0.31	256^2

C Additional results

In this appendix, we provide additional qualitative results between UD^{2}M applied to the problems in Section 4. Qualitative comparisons to similar methods for deblurring and super-resolution on ImageNet may be found in Figures 10 and 11. Qualitative reconstructions obtained via the universal UD^{2}M scheme discussed in Section 4.3.6 can be found in Figure 12.

Additional qualitative comparisons of UD^{2}M (with RAM) on the LSUN bedroom validation set can be found in Figures 13 and 14 for the gaussian deblurring and random inpainting tasks, each with in-distribution noise level $\sigma = 0.025$. To illustrate qualitatively the extensibility of the unfolded architecture to out-of-distribution noise, additional comparisons are presented in Figure 15 using noise level $\sigma = 0.05$ which was not used in training UD^{2}M. For the comparison methods on LSUN bedroom, we use the reconstructed images provided in Garber & Tirer (2025). To illustrate the generative capabilities of the UD^{2}M prior, additional reconstructions for box inpainting on LSUN bedroom are provided in Figure 16

[5]An implementation of this operator can be found in Deepinverse.

Figure 10: Examples of posterior samples for the task Gaussian deblurring on ImageNet with noise level $\sigma = 0.05$ and Gaussian kernel bandwidth of 10.

Figure 11: Examples of posterior samples for the task SR(×4) with noise level $\sigma = 0.05$ on ImageNet 256

.

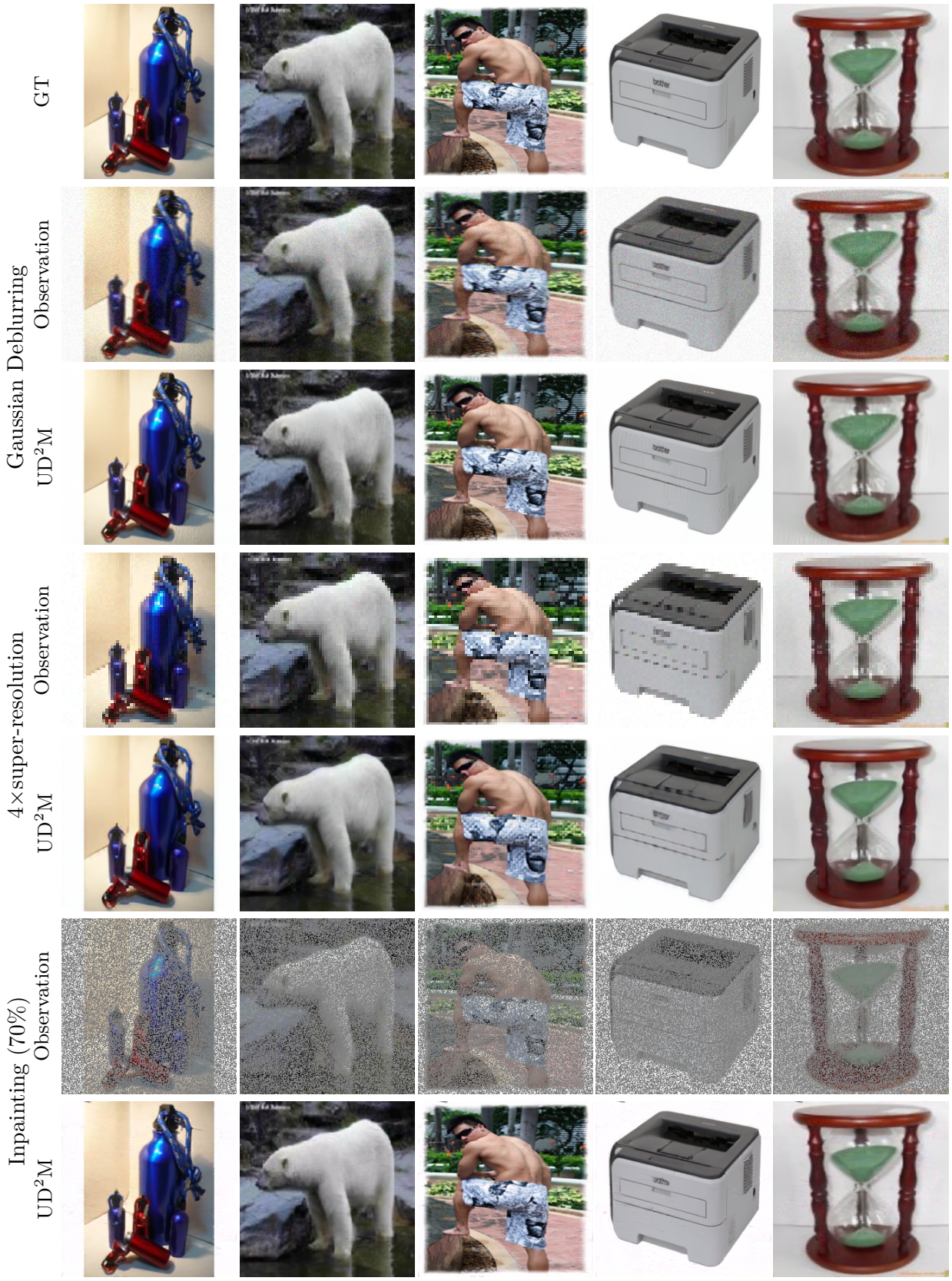


Figure 12: Examples of posterior samples generated by a "universal" UD^{2}M sampler on the following tasks on ImageNet 256: Gaussian Deblurring with a kernel bandwidth of 10, random inpainting (70%), super-resolution (×4). Noise level $\sigma = 0.01$.

Figure 13: Qualitative results from the LSUN bedroom validation set applied to anisotropic Gaussian deblurring with bandwidth $(20, 1)$ and additive Gaussian noise with standard deviation $\sigma = 0.025$.

Figure 14: Qualitative results from the LSUN bedroom validation set applied to random inpainting with 80% masked pixels and additive Gaussian noise with standard deviation $\sigma = 0.025$.

Figure 15: Qualitative results from the LSUN bedroom validation set applied to anisotropic Gaussian deblurring with bandwidth $(20, 1)$ (top) and random inpainting with 80% masked pixels (bottom). Each with additive Gaussian noise with standard deviation $\sigma = 0.05$, which is out of the training distribution for our model which was trained for $\sigma = 0.025$.

Figure 16: Examples of posterior samples for the task box inpainting with noise level $\sigma = 0.05$ on LSUN Bedroom. From top to bottom: Ground truth, degraded observation, UD^{2}M reconstruction (without RAM).