This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

LATINO-PRO: LAtent consisTency INverse sOlver with PRompt Optimization

Alessio Spagnoletti[1]* Jean Prost[2]* Andrés Almansa[1] Nicolas Papadakis[3] Marcelo Pereyra[4]

[1]Université Paris Cité, CNRS, MAP5 UMR 8145, F-75006 Paris, France [2]Université de Lille, CRIStAL UMR 9189, F-59655 Villeneuve d'Ascq, France
[3]Univ. Bordeaux, CNRS, INRIA, Bordeaux INP, IMB, UMR 5251, F-33400 Talence, France
[4]Heriot-Watt University, MACS & Maxwell Institute for Mathematical Sciences, EH14 4AS, Edinburgh, United Kingdom

Abstract

Text-to-image latent diffusion models (LDMs) have recently emerged as powerful generative models with great potential for solving inverse problems in imaging. However, leveraging such models in a Plug & Play (PnP), zero-shot manner remains challenging because it requires identifying a suitable text prompt for the unknown image of interest. Also, existing text-to-image PnP approaches are highly computationally expensive. We herein address these challenges by proposing a novel PnP inference paradigm specifically designed for embedding generative models within stochastic inverse solvers, with special attention to Latent Consistency Models (LCMs), which distill LDMs into fast generators. We leverage our framework to propose LAtent consisTency INverse sOlver (LATINO), the first zero-shot PnP framework to solve inverse problems with priors encoded by LCMs. Our conditioning mechanism avoids automatic differentiation and reaches SOTA quality in as little as 8 neural function evaluations. As a result, LATINO delivers remarkably accurate solutions and is significantly more memory and computationally efficient than previous approaches. We then embed LATINO within an empirical Bayesian framework that automatically calibrates the text prompt from the observed measurements by marginal maximum likelihood estimation. Extensive experiments show that prompt self-calibration greatly improves estimation, allowing LATINO with PRompt Optimization to define new SOTAs in image reconstruction quality and computational efficiency. The code is available at latino-pro.github.io.

1. Introduction

We seek to recover on an unknown image of interest $\boldsymbol{x}$, taking values in $\mathbb{R}^n$, from a measurement

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

where $\mathcal{A}$ is a linear measurement operator, and $\boldsymbol{n}$ is additive Gaussian noise with covariance $\sigma_n^2\mathrm{Id}$. We consider situations in which the recovery of $\boldsymbol{x}$ from $\boldsymbol{y}$ is ill-conditioned or ill-posed, leading to significant uncertainty about the solution. Adopting a Bayesian approach, we leverage prior knowledge about $\boldsymbol{x}$ in order to regularize the problem and deliver meaningful inferences that are well-posed. This is achieved by specifying the marginal distribution $p(\boldsymbol{x})$, so-called prior distribution, together with Bayes' theorem to obtain the posterior distribution $p(\boldsymbol{x}|\boldsymbol{y}) = p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})/p(\boldsymbol{y})$ which models our knowledge about $\boldsymbol{x}$ after observing $\boldsymbol{y}$.

Modern imaging techniques rely increasingly on pre-trained foundational generative models as image priors. Developing methodology to leverage such models in a so-called Plug & Play (PnP) manner, in combination with a likelihood function $p(\boldsymbol{y}|\boldsymbol{x})$ specified during test time, is a highly active research area. In particular, strategies relying on diffusion

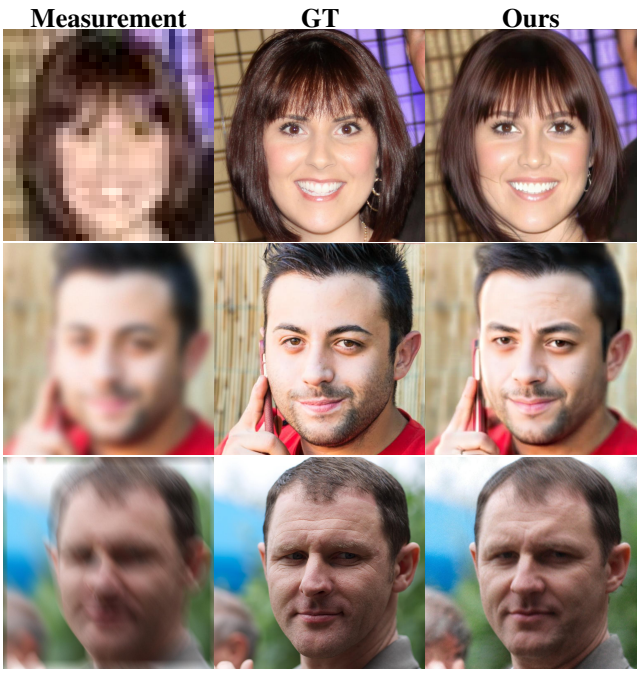

Figure 1. Qualitative comparison of LATINO-PRO on the FFHQ-1024 val dataset. Tasks: $\times 32$ super-resolution, Gaussian deblur $\sigma = 20.0$ pixels, Motion deblur.

*Equal contribution.

models (DM), pioneered in [56–58], superseded GANs [12] and established themselves as the default approach; with notable examples including, e.g., DPS [7], DDRM [22], DiffPIR [67], ΠGDM [55], and mid-point [37]. In order to reduce computational efficiency, modern DMs operate in the latent space of a variational autoencoder (VAE) [25, 28]; leading to Latent Diffusion Models (LDMs) [47]. LDMs simply project the data into a latent space using an encoder, $\mathcal{E}$, perform the diffusion in the latent space, and use a decoder, $\mathcal{D}$, to move back to pixel space. However, since the likelihood is specified in pixel space, naive adaptations of DM-based imaging algorithms fail when applied directly to LDMs. Instead, the specific properties of the latent space need to be considered, as proposed in PSLD [48].

Another fundamental advantage of using LDMs like the celebrated Stable Diffusion models [42, 47] as image priors is the possibility to condition on a textual prompt input. Prompts can be highly informative, so it is natural to seek to exploit them for imaging. Specifying a suitable prompt a-priori is difficult, so P2L [8] includes a strategy to calibrate the prompt automatically, while TReg [23] exploits classifier-free guidance to tune the null prompt.

A main limitation of all existing (L)DM-based imaging methods is that they rely on iterative sampling strategies that require hundreds or thousands of neural function evaluations (NFEs) to produce a single sample from $p(\boldsymbol{x}|\boldsymbol{y})$. As a result, they are very computationally expensive. Also, they cannot support Bayesian inferences that require drawing a large number of samples from $p(\boldsymbol{x}|\boldsymbol{y})$, e.g., to compute statistical estimators, quantify uncertainty, etc. This is in sharp contrast with the latest generative models, such as Consistency Models (CM) [30, 59] and Distilled Diffusion Models (DDM) [29, 35, 50], which can produce remarkable samples from the prior $p(\boldsymbol{x})$ in a few NFEs. Some concurrent works explore CMs within a PnP sampling framework [14, 27, 62], and one could also consider fine-tuning the models to learn to sample from $p(\boldsymbol{x}|\boldsymbol{y})$ instead of $p(\boldsymbol{x})$, e.g., see CoSIGN [66]. Unfortunately, again we are not aware of any existing imaging methods capable of achieving this for high-resolution latent CMs, exploiting the prompt conditioning.

We aim at developing a method that is simultaneously zero-shot, fast (low number of NFEs), light in GPU memory usage (no auto-grad required), that can sample high-resolution images, and which can be conditioned by a text prompt. In particular, the last two conditions are currently only satisfied by LDM-based methods TReg and P2L requiring ≥ 200 NFE per sample as well as the use of additional correction terms. Our two main contributions are:

1. We propose LATINO, a new PnP approach that leverages pre-trained text-to-image LCMs to sample from the posterior $p(\boldsymbol{x} \mid \boldsymbol{y}, c)$ in a gradient-free way, and with few NFEs per sample. This allows scaling to large images ($\geq 1024^2$) with low inference time and low GPU memory footprint. LATINO is by nature prompt-conditioned, allowing users to input semantic information. LATINO is derived from a careful discretization of a Langevin diffusion and a novel PnP approach tailored for LCMs, which involves stochastic auto-encoders instead of denoisers.
2. We equip LATINO with automatic prompt optimization (LATINO-PRO), via a stochastic approximation proximal gradient scheme to simultaneously find $\hat{c}(\boldsymbol{y}) = \arg\max_{c \in \mathbb{R}^k} p(\boldsymbol{y} \mid c)$ and sample from $p(\boldsymbol{x} \mid \boldsymbol{y}, \hat{c}(\boldsymbol{y}))$. This method can correct incomplete or misleading prompts while still requiring only 68 NFEs.

2. Related works

Data-driven regularization for inverse problem. Deep learning regularization methods demonstrated a significant performance improvement over classical inverse problem solvers on a large number of applications. Denoising-based regularization such as RED and Plug-and-play implicitly defines the prior model via a denoising neural network [33, 46, 65]. Deep generative model can also be used to define the prior model on the solution, including GANs [11, 36, 38], variational autoencoders [2, 15, 43], normalizing flows [6]. Denoising diffusion models can also be used repurposed to solve inverse problem by defining an additional guidance term to enforce consistency with measurements [7, 22, 55]. In this work, we instead use a latent consistency model.

Latent diffusion inverse problem solvers (LDIS). Latent diffusion models are significantly faster than classical diffusion models which operate in the pixel space. Various strategies were proposed to condition the latent diffusion process to a measurement, including optimizing the latent iterates to force consistency with the measurements [16, 48, 53], or using a second-order sampler [49]. Unlike the proposed LATINO solver, those methods necessitate to backpropagate through the decoder and the latent score network, and are thus less efficient in terms of memory usage.

Inverse problem solvers with text-to-image generative models. Latent diffusion inverse solvers (LDIS) originally relied on null text embeddings $c_\varnothing$, while P2L [8] has been the first to try to optimize it to improve the reconstruction. TReg [23] instead adopts the CFG [17] framework to optimize the null text embedding while still enabling the user to input a custom conditioning prompt. Both TReg and P2L use the proximal operator of the data fidelity, similar to our LATINO solver, although in those works the proximal operator is motivated by connection with ADMM [5, 39]. In Appendix A we detail (text-based) LDIS algorithms.

Langevin PnP inverse problem solvers. DMs can be used as PnP priors within Langevin Markov chain Monte Carlo

samplers. Notably, [32] embeds a DM within a PnP unadjusted Langevin algorithm (ULA) [26] that computes the posterior mean $\mathrm{E}(\boldsymbol{x}|\boldsymbol{y})$, whereas [10] embeds a DM within a split-Gibbs sampler [61] that is equivariant to a noisy ULA [41]. Moreover, [9] and [34] consider PnP priors encoded by a normalizing flow, whereas [19] provides a general theoretical framework for using VAE and GAN priors.

3. Background

In this section, we will first recall the main properties of DMs and LDMs. Then, we will introduce the Consistency Model formulation that is used by methods like Distribution Matching Distillation (DMD) to sample from the prior distribution in as few as 4 steps. This last ingredient will be crucial for our implementation, allowing for fast IS.

Diffusion Models are generative models designed to learn the reverse of a forward noising process [18, 52, 57, 58]. Starting from an initial distribution $p_0(\boldsymbol{x}), \boldsymbol{x} \in \mathbb{R}^n$, the forward process gradually transforms it into a standard Gaussian distribution $p_T(\boldsymbol{x}) = \mathcal{N}(0, \mathrm{Id})$ as $T \to \infty$. In the variance-preserving (VP) framework proposed by Ho et al. [18], both forward and reverse processes are described using Ito stochastic differential equations (SDEs) [58]:

$$
\begin{aligned}
d\boldsymbol{x}_t &= -\frac{\beta_t}{2}\boldsymbol{x}_t dt + \sqrt{\beta_t} d\boldsymbol{w}, \\
d\boldsymbol{x}_t &= \left[-\frac{\beta_t}{2}\boldsymbol{x}_t - \beta_t \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)\right] dt + \sqrt{\beta_t} d\overline{\boldsymbol{w}}, \quad (1)
\end{aligned}
$$

where β_t represents the noise schedule, and $\boldsymbol{w}, \overline{\boldsymbol{w}}$ are Brownian motions for forward and reverse directions, respectively. The term $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)$ is typically approximated using a score network $s_\theta(\cdot)$, trained with denoising score matching (DSM) [60]. The main property of (1) is to possess also an equivalent Probability Flow ODE formulation [58], in which all the stochasticity is concentrated in the initial step $p_T(\boldsymbol{x})$:

$$
d\boldsymbol{x}_t = \left[-\frac{\beta_t}{2}\boldsymbol{x}_t - \frac{\beta_t}{2}\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)\right] dt. \quad (2)
$$

While image diffusion models in pixel space $\boldsymbol{x}$ are computationally intensive, a more efficient alternative is to operate in a lower-dimensional latent space using an autoencoder [25, 47]. This approach is represented as follows:

$$
\mathcal{E} : \mathbb{R}^n \mapsto \mathbb{R}^d, \quad \mathcal{D} : \mathbb{R}^d \mapsto \mathbb{R}^n, \quad \boldsymbol{x} \approx \mathcal{D}(\mathcal{E}(\boldsymbol{x})),
$$

where $\mathcal{E}$ and $\mathcal{D}$ denote the encoder and decoder, respectively, with $d \ll n$. By encoding images into a latent space $\boldsymbol{z} = \mathcal{E}(\boldsymbol{x})$ [47], one can train a diffusion model on the reduced representation, significantly decreasing computational costs and making it feasible to model high-resolution images (e.g., $\geq 512^2$ pixels). Latent diffusion models (LDMs) have thus become the standard for generative image models, notably under the name Stable Diffusion (SD).

A key distinction between SD and traditional image diffusion models [12] lies in the integration of text conditioning $s_\theta(\cdot, c)$, where c is the prompt conditioning, which in practice is continuously embedded as a vector by the CLIP text encoder [44]. Since SD is trained on the large-scale LAION-5B dataset [51], which consists of image-text pairs, it can be conditioned at inference time to generate text-aligned images through $s_\theta(\cdot, c)$ or classifier-free guidance (CFG) [17].

Consistency Models are obtained by distillation of a pre-trained DM or trained from scratch, allow few or even single-step generation [59]. CMs are defined by using a so-called *consistency function*, a pushforward map defined as follows:

Definition 1 (Consistency function)**.** *Given a small $\eta > 0$ and a trajectory $\{\boldsymbol{x}_t\}_{t\in[\eta,T]}$ of the PF-ODE (2), we define the consistency function as $f : (\boldsymbol{x}_t, t) \to \boldsymbol{x}_\eta$.*

Given such a function, it is straightforward to sample $\boldsymbol{x}_\eta = f_\theta(\boldsymbol{x}_T, T)$ with $\boldsymbol{x}_T \sim \mathcal{N}(0, \mathrm{Id})$ in a single step. This function is indeed self-consistent in the sense that $f(\boldsymbol{x}_t, t) = f(\boldsymbol{x}_{t'}, t') \ \forall\, t, t' \in [\eta, T]$. Given timesteps $t_1 > t_2 > \cdots > t_{N-1} > \eta$, the multistep consistency sampling process is

$$
\begin{aligned}
&\hat{\boldsymbol{x}}_T \sim \mathcal{N}(0, \mathrm{Id}), \quad \boldsymbol{x} = f_\theta(\hat{\boldsymbol{x}}_T, T) \\
&\text{For } n = 1 \text{ to } N-1: \\
&\quad \hat{\boldsymbol{x}}_{t_n} = \boldsymbol{x} + \sqrt{(1-\alpha_{t_n}) - (1-\alpha_\eta)}\epsilon \ \text{ with } \ \epsilon \sim \mathcal{N}(0, \mathrm{Id}) \\
&\quad \boldsymbol{x} = f_\theta(\hat{\boldsymbol{x}}_{t_n}, t_n),
\end{aligned}
$$

where $\alpha_t := \prod_{s=1}^{t}(1-\beta_s)$. As done with DMs, it is possible to define a CM in the latent space [30] and train it by distilling a LDM [31] to obtain a LCM, e.g. by leveraging a LoRA [20] fine-tuning as done in LCM-LoRA [31].

A recent paradigm to distill a LDM into a one-step generator is the Distribution Matching Distillation (DMD) [63] in which a generator G_θ with the same architecture of the DM denoiser is trained. DMD distills many-step diffusion models into a one-step generator G by minimizing the expectation over t of approximate Kullback-Liebler (KL) divergences between the diffused latent target distribution $p_{\text{real},t}$ and the diffused latent generator output distribution $p_{\text{fake},t}$. Since DMD trains G by gradient descent, it only requires the gradient of this loss, which can be computed as the difference of two score functions. DMD uses a frozen pre-trained DM μ_{real} (the teacher), and dynamically updates a fake DM μ_{fake} while training G_θ, using a denoising score-matching loss on samples from the one-step generator, i.e., fake data.

DMD's improved version, DMD2 [64], adds during training a GAN-based loss term, which increases the stability and quality, allowing the obtained distilled model to beat his teacher one. Furthermore, the generator G_θ can be

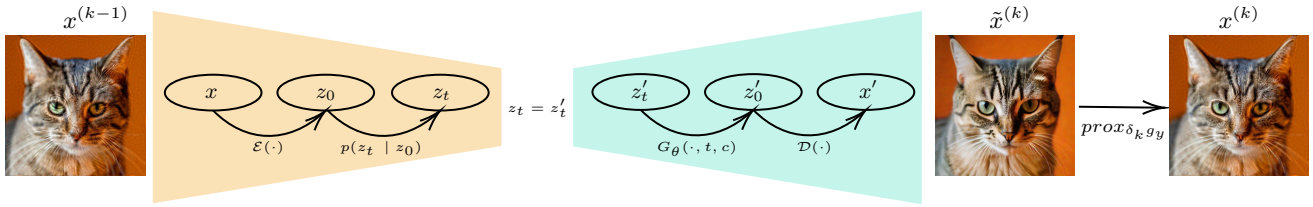

Figure 2. One step of the LATINO solver, a discretization of the Langevin SDE (3) which targets the posterior $p(\boldsymbol{x}|\boldsymbol{y}, c)$. The current iterate $\boldsymbol{x}_k$ is encoded by the VAE encoder and propagated forward via a noising diffusion kernel $p(\boldsymbol{z}_t|\boldsymbol{z}_0)$. This process is then reversed via the latent consistency model and the VAE decoder, followed by the proximal operator to involve the likelihood $p(\boldsymbol{y}|\boldsymbol{x})$.

conditioned on t leading to $G_\theta(\boldsymbol{z}_{t_i}, t_i)$ that predicts the final $\boldsymbol{z}_0$ given a noisy input $\boldsymbol{z}_{t_i}$. By deploying the generator following the CM framework $\hat{\boldsymbol{z}}_{0|t_i} = G_\theta(\boldsymbol{z}_{t_i}, t_i)$ and $\boldsymbol{z}_{t_{i+1}} = \sqrt{\alpha_{t_{i+1}}}\hat{\boldsymbol{z}}_{0|t_i} + \sqrt{1-\alpha_{t_{i+1}}}\epsilon$, one obtains a 4-step sampler. In practice, what is done is to consider G_θ as the pre-trained SDXL U-Net [42], that is fine-tuned for the times $t_i \in \{999, 749, 499, 249\}$, the 4 used by the CM. The final $\hat{\boldsymbol{z}}$ is then passed through the decoder to get $\hat{\boldsymbol{x}} = \mathcal{D}(\hat{\boldsymbol{z}})$.

4. LATINO

4.1. A new Plug & Play Langevin sampling method

We propose a LAtent consisTency INverse sOlver (LATINO) to approximately draw samples from the posterior distribution $p(\boldsymbol{x}|\boldsymbol{y}, c)$, parametrized by the degraded observation $\boldsymbol{y}$ and a text prompt c. Following a PnP philosophy, LATINO is constructed by combining an analytical likelihood function $p(\boldsymbol{y}|\boldsymbol{x})$ with a prior distribution $p(\boldsymbol{x}|c)$ that is implicitly encoded by an text-to-image CM, obtained by distilling a foundational SD model. However, unlike the conventional PnP approach that exploits a denoising operator in lieu of a gradient or proximal mapping of a clean image prior $p(\boldsymbol{x})$, LATINO pioneers a new PnP approach suitable for leveraging generative models such as CMs. In this paradigm, the CM is first auto-encoded to define a Markov kernel that admits $p(\boldsymbol{x}|c)$ as invariant distribution. The kernel is subsequently used in place of an SDE targeting $p(\boldsymbol{x}|c)$ within a stochastic sampler for $p(\boldsymbol{x}|\boldsymbol{y}, c)$.

More precisely, consider an overdamped Langevin diffusion process to sample from $p(\boldsymbol{x}|\boldsymbol{y}, c)$, given by the SDE

$$\mathrm{d}\boldsymbol{x}_s = \nabla \log p(\boldsymbol{y}|\boldsymbol{x}_s)\mathrm{d}s + \nabla \log p(\boldsymbol{x}_s|c)\mathrm{d}s + \sqrt{2}\mathrm{d}\boldsymbol{w}_s\,, \quad (3)$$

where $\boldsymbol{w}_s$ denotes a n-dimensional Brownian motion. Under mild regularity assumptions on $p(\boldsymbol{x}|\boldsymbol{y}, c)$ [13], starting from an initial condition $\boldsymbol{x}_0$, the process $\boldsymbol{x}_s$ converges to $p(\boldsymbol{x}|\boldsymbol{y}, c)$ exponentially fast as s increases [13]. While solving (3) exactly is not generally possible, it provides a powerful computational engine to derive samplers that approximately sample $p(\boldsymbol{x}|\boldsymbol{y}, c)$ by mimicking $\boldsymbol{x}_s$. For example, an Euler-Maruyama approximation of (3) leads to the widely used unadjusted Langevin algorithm (ULA) [13].

LATINO stems from the following approximation of (3)

$$\boldsymbol{u} = \boldsymbol{x}_k + \int_0^\delta \nabla \log p(\tilde{\boldsymbol{x}}_s|c)\mathrm{d}s + \sqrt{2}\mathrm{d}\boldsymbol{w}_s\,,\ \tilde{\boldsymbol{x}}_0 = \boldsymbol{x}_k\,,$$
$$\boldsymbol{x}_{k+1} = \boldsymbol{u} + \delta\nabla \log p(\boldsymbol{y}|\boldsymbol{x}_{k+1})\,, \quad (4)$$

where we note the first half of the splitting, encoded in $\boldsymbol{u}$, solves (3) exactly with likelihood term removed, and the second step involves the likelihood via an implicit Euler step. Note that (4) has two key advantages over ULA: it is potentially very accurate, as the first step introduces no discretisation bias; and it is numerically stable for all $\delta > 0$, so it can be made to converge arbitrarily quickly by increasing δ, at the expense of some bias. Conversely, ULA only integrates the Brownian term $\boldsymbol{w}_s$ exactly, it involves the drift via an explicit Euler step, and is explosive unless δ is sufficiently small. Also note that the implicit Euler step in (4) can be reformulated as an explicit proximal point step, i.e., $\tilde{\boldsymbol{x}}_{k+1} = \mathrm{prox}_{\delta g_y}(\tilde{\boldsymbol{x}}_{k+1})$, where $g_y : \boldsymbol{x} \mapsto -\log p(\boldsymbol{y}|\boldsymbol{x})$[40], which is computationally tractable for common inverse problems such as deblurring or super-resolution[1].

The computation of the intermediate step $\boldsymbol{u}$ from $\boldsymbol{x}_k$ in (4) is usually intractable. LATINO circumvents this difficulty by noting that this step defines a contractive Markov kernel with $p(\boldsymbol{x}|c)$ as unique invariant distribution. Our proposed PnP approach replaces this kernel in (4) with a Stochastic Auto-Encoder (SAE) derived from a CM, which we design such that it also contracts random variables towards $p(\boldsymbol{x}|c)$.

4.2. Auto-encoding stable diffusion

Consider a distilled SD model which samples from $p(\boldsymbol{x}|c)$ by using a CM, G_θ, operating on the latent space of a deterministic auto-encoder $(\mathcal{E}, \mathcal{D})$. Assume G_θ solves the flow

$$d\boldsymbol{z}_t = \left[-\frac{\beta_t}{2}\boldsymbol{z}_t - \frac{\beta_t}{2}\nabla_{\boldsymbol{z}_t}\log p_t(\boldsymbol{z}_t)\right]dt\,, \quad (5)$$

which generates a set of distributions $p_0(\boldsymbol{z}_0|c)$ by pushing forward a latent normal random variable through $G_\theta(\cdot, T, c)$

[1]Note that $\mathrm{prox}_{\delta g_y}(\boldsymbol{x}) = \left(\delta\mathcal{A}^t\mathcal{A} + \sigma_n^2\mathrm{Id}\right)^{-1}\left(\delta^1 A^t\boldsymbol{y} + \sigma_n^2\boldsymbol{x}\right)$, which can be computed cheaply when we have access to the singular value decomposition of $\mathcal{A}$ (see, e.g., [65]), or by using a specialised subiteration.

for a sufficiently large T, which in turn produces the target distribution $p(\boldsymbol{x}|c)$ through the action of the decoder $\mathcal{D}$.

Our aim is to use G_θ and $(\mathcal{E}, \mathcal{D})$ to construct a SAE $(\mathfrak{E}_t, \mathfrak{D}_{t,c})$ such that it contracts random variables on $\mathbb{R}^n$ towards $p(\boldsymbol{x}|c)$, its fixed point. In order to achieve this, we define the stochastic encoder parametrized by $t \in (0, T]$

$$\mathfrak{E}_t : \quad \boldsymbol{z}_t|\boldsymbol{x} \sim \mathcal{N}(\sqrt{\alpha_t}\mathcal{E}(\boldsymbol{x}), (1-\alpha_t)\mathrm{Id}_d)\,.$$

$\mathfrak{E}$ maps $\boldsymbol{x}$ to the output $\boldsymbol{z}_t$ by first applying the deterministic encoder to compute $\boldsymbol{z}_0 = \mathcal{E}(\boldsymbol{x})$ and subsequently drawing $\boldsymbol{z}_t$ conditionally to $\boldsymbol{z}_0$, as given by the SDE $d\boldsymbol{z}_t = -\frac{\beta_t}{2}\boldsymbol{z}_t dt + \sqrt{\beta_t} d\boldsymbol{w}'$, which inverts the probability flow (5) [59].

This stochastic encoder $\mathfrak{E}_t$ is paired with the decoder $\mathfrak{D}_{t,c}$, which takes as input $\boldsymbol{z}'_t = \boldsymbol{z}_t$ and pushes forward to the ambient space through the deterministic mapping

$$\mathfrak{D}_{t,c} : \quad \boldsymbol{x}' = \mathcal{D}(G_\theta(\boldsymbol{z}'_t, t, c))\,.$$

The proposed AE architecture is summarized in Figure 2.

Assume that $\mathcal{E}$ inverts $\mathcal{D}$ exactly. If $\boldsymbol{x}$ is distributed according to $p(\boldsymbol{x}|c)$, then, for any $t > 0$, $\boldsymbol{z}_t = \mathfrak{E}_t(\boldsymbol{x})$ has distribution p_t as described by the flow (5). Equally, if $\boldsymbol{z}'_t$ follows p_t, then $\boldsymbol{z}'_0 = G_\theta(\boldsymbol{z}'_t, t, c)$ follows p_0 and hence $\boldsymbol{x}' = \mathfrak{D}(\boldsymbol{z}'_t, t)$ follows $p(\boldsymbol{x}|c)$, implying that $p(\boldsymbol{x}|c)$ is a fixed point as required (in praxis, for numerical stability, G_θ targets $p_\tau(\boldsymbol{z}_\tau|c) \approx p_0(\boldsymbol{z}_0|c)$ for $\tau > 0$ small and $\mathcal{E}$ does not perfectly invert $\mathcal{D}$, leading to some small bias).

Alternatively, if $\boldsymbol{x}$ is not distributed according to $p(\boldsymbol{x}|c)$, $(\mathfrak{E}_t, \mathfrak{D}_{t,c})$ will transport it towards $p(\boldsymbol{x}|c)$. In this situation, the parameter t plays a key role, analogous to a regularization parameter. Suppose that t is very large, such that the encoded random variable $\boldsymbol{z}_t \approx \mathcal{N}(0, \mathrm{Id})$ regardless of the distribution of $\boldsymbol{x}$. Then $\mathfrak{D}$ behaves as a standard SD generative model and output $\boldsymbol{x}'$ distributed according to $p(\boldsymbol{x}|c)$. Conversely, now suppose that t is very small. In that case, $(\mathfrak{E}_t, \mathfrak{D}_{t,c})$ reduce to $(\mathcal{E}, \mathcal{D})$. Because of the consistency property of G_θ, we have $\boldsymbol{x}' \approx \mathcal{D}[\mathcal{E}(\boldsymbol{x})]$. This implies that $\boldsymbol{x}' \approx \boldsymbol{x}$ if the distribution of $\boldsymbol{x}$ is concentrated on the range of $\mathcal{D}$. For intermediate values of t, not infinitely large or infinitesimally small, the stochastic autoencoder $\boldsymbol{x}$ is transported some extent towards $p(\boldsymbol{x}|c)$. We conclude that $(\mathfrak{E}, \mathfrak{D})$ has the desired fixed point and contraction properties, with t controlling the contraction strength analogously to the step-size or integration period δ in (4). This is summarized in Figure 3 and by the recursion

$$\begin{aligned} \tilde{\boldsymbol{x}}_{k+1} &= \mathfrak{D}_{t,c} \circ \mathfrak{E}_t(\boldsymbol{x}_k)\,, \\ \boldsymbol{x}_{k+1} &= \mathrm{prox}_{\delta g_y}(\tilde{\boldsymbol{x}}_{k+1})\,. \end{aligned} \tag{6}$$

Note that, because (6) approximates (4), t and δ must be in agreement, similarly to conventional PnP methods that need to balance the step-size and the denoiser's parameters [26].

4.3. The LATINO algorithm

Algorithm 1 below summarizes the proposed LATINO sampling scheme, which is based on recursion (6) (see also Figure 2). We warm-start the algorithm by using the observation-informed initialization $\boldsymbol{x}^{(0)} = \mathcal{A}^\dagger \boldsymbol{y}$. For computational efficiency, we use an annealing strategy that involves iteration-dependent parameters t_k, δ_k. We now introduce the choice of t_k and leave to Appendix B the details on δ_k. In our experiments, we find that $N = 8$ suffice to deliver remarkably accurate samples when G_θ is given by the DMD2 [64].

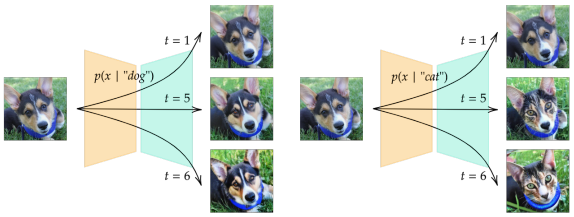

Figure 3. SAE applied to images in and out of distribution for different values of t, illustrating contraction towards $p(\boldsymbol{x}|c)$.

Algorithm 1 LATINO

given $\boldsymbol{x}^{(0)} = \mathcal{A}^\dagger \boldsymbol{y}$, text prompt c, number of step $N = 4$ or 8, latent consistency model G_θ, latent space decoder $\mathcal{D}$, latent space encoder $\mathcal{E}$, sequences $\{t_k, \delta_k\}_{k=1}^N$.
for $k = 1, \dots, N$ **do**
  $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathrm{Id})$
  $\boldsymbol{z}_{t_k}^{(k)} \leftarrow \sqrt{\alpha_{t_k}}\mathcal{E}(\boldsymbol{x}^{(k-1)}) + \sqrt{1-\alpha_{t_k}}\boldsymbol{\epsilon}$ ▷ Encode
  $\boldsymbol{u}^{(k)} \leftarrow \mathcal{D}(G_\theta(\boldsymbol{z}_{t_k}^{(k)}, t_k, c))$ ▷ Decode
  $\boldsymbol{x}^{(k)} \leftarrow \mathrm{prox}_{\delta_k g_{\boldsymbol{y}}}(\boldsymbol{u}^{(k)})$ ▷ $g_{\boldsymbol{y}} : \boldsymbol{x} \mapsto -\log p(\boldsymbol{y}|\boldsymbol{x})$
end for
return $\boldsymbol{x}^{(N)}$

The LCM prior choice. We use DMD2 [64], a pretrained model based on the SDXL architecture [42]. That said, our method is agnostic to the choice of LCM; e.g., see appendix G for a comparison of our method using DMD2 and SD1.5-LoRA. We can either consider the original DMD2 4-step prior or an 8-step variant of DMD2, where the timesteps are $t_i \in \{999, 874, 749, 624, 499, 374, 249, 124\}$. We mix the 4 distilled steps $t_i \in \{999, 749, 499, 249\}$ with the 4 steps from the original SDXL model. This choice has been adopted to increase the quality of reconstructions by doubling the number of steps. Still, 4 distilled time steps gives excellent results, and we make use of this version in our LATINO-PRO scheme that self-calibrates the prompt c.

Interestingly, if instead of an LCM we construct $(\mathfrak{E}_t, \mathfrak{D}_{t,k})$ with a DM we recover the DM-based PnP-ULA scheme of [32] as a special case, whereas if we use the stochastic denoiser [45] we recover a PnP-ULA [26].

5. LATINO-PRO: Prompt optimization via maximum marginal likelihood estimation

State-of-the-art generative models such as DMD2 provide a highly informative prior $p(\boldsymbol{x}|c)$ that concentrates its probability mass in very narrow regions of the solution space. Conversely, in inverse problems that are ill-conditioned or ill-posed, the likelihood function $p(\boldsymbol{y}|\boldsymbol{x})$ has identifiability issues and is only weakly informative as a result. Hence, the prior $p(\boldsymbol{x}|c)$ dominates key aspects of the posterior $p(\boldsymbol{x}|\boldsymbol{y}, c)$. Consequently, it is essential to calibrate $p(\boldsymbol{x}|c)$ by identifying a suitable text prompt c, as otherwise $p(\boldsymbol{x}|\boldsymbol{y}, c)$ exhibits strong bias. Unfortunately, setting c a priori is difficult.

Adopting an empirical Bayesian approach, we propose to set c by maximum marginal likelihood estimation (MMLE),

$$\hat{c}(\boldsymbol{y}) = \arg\max_{c \in \mathbb{R}^k} p(\boldsymbol{y} \mid c)\,, \tag{7}$$

followed by inference with the empirical Bayesian posterior distribution $p(\boldsymbol{x} \mid \boldsymbol{y}, \hat{c}(\boldsymbol{y}))$. The marginal likelihood

$$p(\boldsymbol{y} \mid c) = \mathrm{E}_{\boldsymbol{x}|c}[p(\boldsymbol{y}|\boldsymbol{x})]$$

measures the model's goodness-of-fit to the data $\boldsymbol{y}$. A main challenge in solving (7) is that the likelihood $p(\boldsymbol{y} \mid c)$ is computationally intractable, as it requires integration over the solution space. We address this difficulty by using LATINO to implement a stochastic approximation proximal gradient (SAPG) scheme which seeks to simultaneously solve (7) and draw samples from $p(\boldsymbol{x} \mid \boldsymbol{y}, \hat{c}(\boldsymbol{y}))$.

More precisely, in a manner akin to [3], we consider a stochastic optimization scheme that mimics the following projected gradient algorithm to solve (7)

$$c_{m+1} = \Pi_C[c_m + \gamma_m \nabla_c \log p(\boldsymbol{y} \mid c_m)] \tag{8}$$

where γ_m is a sequence of decreasing positive step-sizes and $C \subset \mathbb{R}^k$ is a convex set of admissible values for c. While (8) is also intractable, from Fisher's identity, the gradient

$$\begin{aligned}\nabla_c \log p(\boldsymbol{y} \mid c) &= \mathrm{E}_{\boldsymbol{x}|\boldsymbol{y},c}[\nabla_c \log p(\boldsymbol{y}, \boldsymbol{x} \mid c)]\,, \\ &= \mathrm{E}_{\boldsymbol{x}|\boldsymbol{y},c}[\nabla_c \log p(\boldsymbol{x} \mid c)]\,,\end{aligned}$$

which suggests implementing a stochastic variant of (8) with

$$\nabla_c \log p(\boldsymbol{y} \mid c_m) \approx \nabla_c \log p(\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)} \mid c_m),$$

where $\{\boldsymbol{x}^{(k)}\}_{k=1}^N$ is a Markov chain targeting $p(\boldsymbol{x}|\boldsymbol{y}, c_m)$, as generated by LATINO (Alg. 1). The resulting scheme alternates between using LATINO to generate a batch $\{\boldsymbol{x}^{(k)}\}_{k=1}^N$ of samples, and then updating c_m through the recursion

$$c_{m+1} = \Pi_C\left[c_m + \gamma_m \nabla_c \log p(\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}|c_m)\right]. \tag{9}$$

In praxis, the computation of (9) is tractable by automatic differentiation on the latent space (see Appendix D for details). LATINO with prompt optimization (LATINO-PRO), is summarized in Algorithm 2 below. Under some technical assumptions, [3, Theorem 5] guarantees that the iterates c_m generated by SAPG schemes driven by Langevin samplers converge to $\hat{c}(\boldsymbol{y})$ as m increases. LATINO-PRO automatically verifies the conditions of [3, Theorem 5] under simplifying assumptions on target $p(\boldsymbol{x}|\boldsymbol{y}, c)$ and $(\mathfrak{E}_t, \mathfrak{D}_{t,c})$ (e.g., smoothness and log-concavity of $p(\boldsymbol{x}|\boldsymbol{y}, c)$). Extending [3, Theorem 5] to derive convergence guarantees for LATINO-PRO under realistic assumptions for CM-based priors is a main perspective for future work.

We recommend warm-starting LATINO-PRO by specifying an initial value for c_0 that describes the expected solution (e.g., "a sharp photo of a dog"). We find that the quality of the samples generated by LATINO-PRO increases rapidly as we start to update c_m, and that stopping iterations early introduces some regularization, which is beneficial. In our experiments, we perform $M = 15$ iterations of (9), with $N = 4$ sub-iterations of Algorithm 1, except in the final stage when we use $N = 8$ iterations of Algorithm 1 to obtain a higher quality sample from $p(\boldsymbol{x}|\boldsymbol{y}, \hat{c}(\boldsymbol{y}))$.

Algorithm 2 LATINO-PRO

given $\boldsymbol{x}^{(0)} = \mathcal{A}^\dagger \boldsymbol{y}$, text prompt c_0 and admissible set C, number of step SAPG steps M, sub-iteration parameters $\{N_m, \gamma_m\}_{m=1}^M, \{t_k, \delta_k\}_{k=1}^{N_m}$, latent consistency model G_θ, latent space decoder $\mathcal{D}$ and encoder $\mathcal{E}$.
for $m = 1, \ldots, M$ **do**
  for $k = 1, \ldots, N_m$ **do** ▷ LATINO
    $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathrm{Id})$
    $\boldsymbol{z}_{t_k}^{(k)} \leftarrow \sqrt{\alpha_{t_k}}\mathcal{E}(\boldsymbol{x}^{(k-1)}) + \sqrt{1-\alpha_{t_k}}\boldsymbol{\epsilon}$
    $\boldsymbol{u}^{(k)} \leftarrow \mathcal{D}(G_\theta(\boldsymbol{z}_{t_k}^{(k)}, t_k, c_m))$
    $\boldsymbol{x}^{(k)} \leftarrow \mathrm{prox}_{\delta_k g_y}(\boldsymbol{u}^{(k)})$
  end for
  $h(c_m) \leftarrow \nabla_c \log p(\boldsymbol{z}_{t_1}^{(1)}, \ldots, \boldsymbol{z}_{t_{N_m}}^{(N_m)}|c_m)$
  $c_{m+1} = \Pi_C\left[c_m + \gamma_m h(c_m)\right]$ ▷ SAPG
  $\boldsymbol{x}^{(0)} \leftarrow \boldsymbol{x}^{(N_m)}$ ▷ Carry state forward
end for
return $\boldsymbol{x}^{(N_M)}$

6. Experiments

Datasets and Models. We consider two high-quality (HQ) datasets to test our models: FFHQ [21] and AFHQ. For FFHQ, we consider both the 1024×1024 and 512×512 versions, depending on the objective of the tests conducted, and we take the first 1k test images, as done by [8]. For AFHQ, we consider the 512×512 version, and we take the validation sets of *dogs* and *cats*, as done in [23]. The model used for our LATINO algorithm is the DMD2 [64] based on SDXL [42]. SD1.5 [47] is instead the prior for all the other LDIS, like P2L, PSLD, LDIR, and LDPS. In Appendix G we discuss why it is fair to compare SD1.5 with DMD2 in

Figure 4. Comparison of image restorations. Samples taken from AFHQ-512. Prompts: `a sharp photo of a dog` (resp. `a cat`).

		Deblur (Gaussian)		SR×16	
Method	**NFE↓**	**FID↓**	**PSNR↑**	**FID↓**	**PSNR↑**
LATINO-PRO	68	**18.37**	**26.82**	**30.40**	**21.52**
LATINO	**8**	20.03	26.25	42.14	20.05
P2L [8]	2000	85.80	20.96	121.7	19.99
TReg [23]	200	35.47	21.13	37.13	19.60
LDPS	1000	64.88	22.60	101.13	17.34
PSLD [48]	1000	125.5	20.52	113.4	16.48

Table 1. Results for Gaussian Deblurring with $\sigma = 5.0$, and $\times 16$ super-resolution, both with noise $\sigma_y = 0.01$ on the AFHQ-512 val dataset. Our LATINO and LATINO-PRO models are compared to recent state-of-the-art methods. Prompts: `a sharp photo of a dog` (resp. `a cat`) **Bold**: best, underline: second best.

terms of prior quality, and we also explore an SD1.5-based CM, to show the universal applicability of our method.

Problems considered. The degradations considered are the same of [8]: Gaussian deblurring with kernel of size 61×61 with $\sigma = 3.0$ for the tests on FFHQ and $\sigma = 5.0$ for those on AFHQ. Motion deblurring with kernel of size 61×61 randomly sampled with intensity 0.5[2]. We also consider various types of Super-Resolution tests, ranging from $\times 8$ upscaling with the average pooling kernel to $\times 16$ upscaling with the bicubic interpolation kernel. We add a white noise of intensity $\sigma_n = 0.01$ to all our tests. We also add in appendix N harder problem settings, providing visual results to show the effectiveness of our algorithms. Additional experiments are reported in Appendix M and E.

Evaluation on inverse problems tasks. Since the current SOTA methods that employ LDMs to solve inverse problems work only with 512×512 resolution, whereas the pretrained model we use works at 1024×1024 resolution, we must adapt our model for fair comparison purposes. Indeed, when comparing with P2L, PSLD, LDIR and LDPS we adapt the inverse problems as follows:

1. Super-resolution $\times 8$ becomes $\times 16$ ($\times 16$ becomes $\times 32$), so the image fed to the algorithm always has the same size, i.e. 64×64. The output of our model is then downsampled to 512×512. More details in Appendix H.2.
2. The Deblurring task is converted into a simultaneous super-resolution and Deblurring problem since we first have to downsample the clean image $\boldsymbol{x}$ to 512×512 and then apply the actual blur operator $\mathcal{A}$. The formal details of this specific case are illustrated in Appendix H.1.

In Appendix I we also provide visual results and FID, PSNR, and LPIPS metrics for the FFHQ1024 case, allowing future work that will handle this resolution to be compared with our results. We can see from Table 1 and Table 2 how both our LATINO and LATINO-PRO models have similar performances compared to current SOTA, if not even better in terms of PSNR and LPIPS. Especially on the AFHQ dataset, we can beat all the SOTA methods in all the metrics considered. On the FFHQ dataset we can beat SOTA in most cases. Importantly, we can see a huge gain in terms of NFEs. Furthermore, LATINO requires around only 13 Gb of GPU memory to run, thanks to the absence of any gradient computation in the steps (more details on memory and time consumption in Appendix J). Considering that the DMD2 prior takes around 10.7Gb, the overhead is minimum. Figures 4 and 11 show a qualitative comparison of the methods proposed against common latent DIS like LDPS, PSLD and P2L. We refer to the TReg original work [23] to compare on the AFHQ dataset with both algorithms, see Appendix K.

Prompt Optimization. As discussed in Section 5, we can use the LATINO-PRO SAPG scheme to optimize the text prompt c. This greatly improves accuracy, both when the prompt is already partially aligned with the image and when it is misleading, as shown in Table 9 in the Appendix L.

[2] https://github.com/LeviBorodenko/motionblur

		Deblur (Gaussian)			Deblur (Motion)			SR×8		
Method	**NFE↓**	**FID↓**	**PSNR↑**	**LPIPS↓**	**FID↓**	**PSNR↑**	**LPIPS↓**	**FID↓**	**PSNR↑**	**LPIPS↓**
LATINO-PRO	68	31.98	**29.11**	**0.292**	27.80	27.14	**0.301**	40.95	26.58	0.355
LATINO	**8**	33.94	28.95	0.296	29.17	26.88	0.318	37.13	26.22	0.356
P2L [8]	2000	**30.62**	26.97	0.299	28.34	**27.23**	0.302	**31.23**	28.55	**0.290**
LDPS	1000	45.89	27.82	0.334	58.66	26.19	0.382	36.81	**28.78**	0.292
PSLD [48]	1000	41.04	28.47	0.320	47.71	27.05	0.348	36.93	26.62	0.335
LDIR [16]	1000	35.61	25.75	0.341	**24.40**	24.40	0.376	36.04	25.79	0.345

Table 2. Results for Gaussian deblurring with $\sigma = 3.0$, motion deblurring, and $\times 8$ super-resolution, all with noise $\sigma_y = 0.01$ on the FFHQ-512 val dataset. Our LATINO and LATINO-PRO models are compared to recent state-of-the-art methods. Prompt: `a sharp photo of a face`. **Bold**: best, underline: second best.

Figure 5. Qualitative comparison of image restoration results. Samples taken from FFHQ-512. Prompt: `a sharp photo of a face`.

We initialize the prompt c by concatenating two strings, a fixed string `a sharp photo of` whose tokens will remain frozen, plus a descriptive string, e.g. `a dog`, which is updated via SAPG updates. The hyperparameters used are $M = 15$, $C = B(c_0, 15)$, and $\gamma_m = 0.1 \cdot 0.9^{\max(0, m-10)}$.

Tables 1 and 2 show that LATINO-PRO can significantly outperform LATINO, with a computational cost that remains $15\times$ lower than competing SOTA methods. Although LATINO-PRO is not gradient-free due to the terms ∇_c, these gradients are computed in the latent space and do not require backpropagation through $\mathcal{D}$ or $\mathcal{E}$. We provide in Table 7 in Appendix J an extensive comparison of the memory consumption and running times of different IR algorithms compared to ours. In Appendix M, we provide inpainting experiments highlighting the capacity of LATINO-PRO to produce diverse image restorations with different prompts.

7. Conclusion

We introduced LATINO, a fast, light, zero-shot, prompt-guided IS that can leverage SOTA latent consistency models such as DMD2 [64]. Its effectiveness has been compared to current SOTA methods, which achieve comparable quality but require two to three orders of magnitude more NFEs. We also presented LATINO-PRO, which self-calibrates the text prompt by maximum likelihood estimation. LATINO-PRO outperforms SOTA on AFHQ512 and FFHQ512 datasets on multiple inverse problems.

Future work will seek to analyze the theoretical properties of LATINO and LATINO-PRO, with special attention to non-asymptotic convergence results. The development of strategies to automatically adjust the parameters of LATINO and LATINO-PRO is another main perspective for future work. Future work will also explore strategies for decoding the prompt embedding to reveal the optimized text prompt.

Acknowledgments and Disclosure of Funding

MP acknowledges support by UKRI Engineering and Physical Sciences Research Council (EPSRC) (EP/V006134/1, EP/Z534481/1). AS, AA, and NP acknowledge support from the France 2030 research program on artificial intelligence via the PEPR PDE-AI grant (ANR-23-PEIA-0004). JP acknowledges support by the "Chaire IA Sherlock" held by P. Chainais, which includes the ANR project ANR-20-CHIA-0031-01, the national support within the *programme d'investissements d'avenir* ANR-16-IDEX-0004 ULNE and Région HdF. HPC resources provided by GENCI-IDRIS Jean-Zay (Grant 2024-AD011014557).

References

[1] Sergio Blanes, Fernando Casas, and Ander Murua. Splitting methods for differential equations. *Acta Numerica*, 33:1–161, 2024. 2

[2] Ashish Bora, Ajil Jalal, Eric Price, and Alexandros G Dimakis. Compressed sensing using generative models. In *International conference on machine learning*, pages 537–546. PMLR, 2017. 2

[3] Valentin De Bortoli, Alain Durmus, Marcelo Pereyra, and Ana Fernandez Vidal. Efficient stochastic optimisation by unadjusted langevin monte carlo. *Statistics and Computing*, 31, 2019. 6

[4] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 - mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014. 7

[5] Stephen P. Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3:1–122, 2011. 2, 1

[6] Ziruo Cai, Junqi Tang, Subhadip Mukherjee, Jinglai Li, Carola Bibiane Schönlieb, and Xiaoqun Zhang. Nf-ula: Langevin monte carlo with normalizing flow prior for imaging inverse problems. *preprint arXiv:2304.08342*, 2023. 2

[7] 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. 2, 1

[8] Hyungjin Chung, Jong Chul Ye, Peyman Milanfar, and Mauricio Delbracio. Prompt-tuning latent diffusion models for inverse problems. In *Forty-first International Conference on Machine Learning*, 2024. 2, 6, 7, 8, 1, 15

[9] Florentin Coeurdoux, Nicolas Dobigeon, and Pierre Chainais. Normalizing flow sampling with langevin dynamics in the latent space. *Machine Learning*, 113(11):8301–8326, 2024. 3

[10] 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*, 2024. 3

[11] Giannis Daras, Joseph Dean, Ajil Jalal, and Alex Dimakis. Intermediate layer optimization for inverse problems using deep generative models. In *International Conference on Machine Learning*, pages 2421–2432. PMLR, 2021. 2

[12] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *(NeurIPS) Advances in Neural Information Processing Systems*, pages 8780–8794. Curran Associates, Inc., 2021. 2, 3

[13] Alain Durmus and Éric Moulines. Nonasymptotic convergence analysis for the unadjusted Langevin algorithm. *The Annals of Applied Probability*, 27(3):1551 – 1587, 2017. 4

[14] 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)*, pages 2398–2407, 2025. 2

[15] Mario González, Andrés Almansa, and Pauline Tan. Solving inverse problems by joint posterior maximization with autoencoding prior. *SIAM Journal on Imaging Sciences*, 15 (2):822–859, 2022. 2

[16] Linchao He, Hongyu Yan, Mengting Luo, Hongjie Wu, Kunming Luo, Wang Wang, Wenchao Du, Hu Chen, Hongyu Yang, Yi Zhang, and Jiancheng Lv. Fast and stable diffusion inverse solver with history gradient update, 2024. 2, 8, 1

[17] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*. 2, 3

[18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 3

[19] 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. 3

[20] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. 3

[21] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4396–4405, 2018. 6

[22] Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. *Advances in Neural Information Processing Systems*, 35:23593–23606, 2022. 2

[23] Jeongsol Kim, Geon Yeong Park, Hyungjin Chung, and Jong Chul Ye. Regularization by texts for latent diffusion inverse solvers. In *The Thirteenth International Conference on Learning Representations*, 2025. 2, 6, 7, 1, 3, 15

[24] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *preprint arXiv:1412.6980*, 2014. 1

[25] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013. 2, 3

[26] 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. 3, 5

[27] 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. 2

[28] Romain Lopez, Pierre Boyeau, Nir Yosef, Michael I. Jordan, and Jeffrey Regier. Auto-encoding variational bayes. In *Proceedings of the Neural Information Processing Systems Conference*, 2020. 2

[29] Eric Luhman and Troy Luhman. Knowledge distillation in iterative generative models for improved sampling speed. *ArXiv*, abs/2101.02388, 2021. 2

[30] 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, 2023. 2, 3

[31] 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. 3, 4

[32] Charlesquin Kemajou Mbakam, Jean-Francois Giovannelli, and Marcelo Pereyra. Empirical bayesian image restoration by langevin sampling with a denoising diffusion implicit prior, 2024. 3, 5

[33] Tim Meinhardt, Michael Moller, Caner Hazirbas, and Daniel Cremers. Learning proximal operators: Using denoising networks for regularizing inverse imaging problems. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1781–1790, 2017. 2

[34] 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. 3

[35] Chenlin Meng, Ruiqi Gao, Diederik P. Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14297–14306, 2022. 2

[36] Sachit Menon, Alexandru Damian, Shijia Hu, Nikhil Ravi, and Cynthia Rudin. Pulse: Self-supervised photo upsampling via latent space exploration of generative models. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 2437–2445, 2020. 2

[37] 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. 2

[38] Xingang Pan, Xiaohang Zhan, Bo Dai, Dahua Lin, Chen Change Loy, and Ping Luo. Exploiting deep generative prior for versatile image restoration and manipulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):7474–7489, 2021. 2

[39] Neal Parikh and Stephen P. Boyd. Proximal algorithms. *Found. Trends Optim.*, 1:127–239, 2013. 2

[40] Marcelo Pereyra. Proximal markov chain monte carlo algorithms. *Statistics and Computing*, 26:745–760, 2016. 4

[41] 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. 3

[42] 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*. 2, 4, 5, 6

[43] Jean Prost, Antoine Houdard, Andrés Almansa, and Nicolas Papadakis. Inverse problem regularization with hierarchical variational autoencoders. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22894–22905, 2023. 2

[44] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021. 3

[45] 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*. JMLR.org, 2024. 5

[46] Yaniv Romano, Michael Elad, and Peyman Milanfar. The little engine that could: Regularization by denoising (red). *SIAM Journal on Imaging Sciences*, 10(4):1804–1844, 2017. 2

[47] 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)*, pages 10674–10685, 2021. 2, 3, 6, 4

[48] Litu Rout, Negin Raoof, Giannis Daras, Constantine Caramanis, Alex Dimakis, and Sanjay Shakkottai. Solving linear inverse problems provably via posterior sampling with latent diffusion models. *Advances in Neural Information Processing Systems*, 36:49960–49990, 2023. 2, 7, 8, 1, 13, 15

[49] L Rout, Y Chen, A Kumar, C Caramanis, S Shakkottai, and W Chu. Beyond first-order tweedie: Solving inverse problems using latent diffusion, 2024. 2

[50] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models, 2022. 2

[51] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294, 2022. 3

[52] Jascha Narain Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. *ArXiv*, abs/1503.03585, 2015. 3

[53] Bowen Song, Soo Min Kwon, Zecheng Zhang, Xinyu Hu, Qing Qu, and Liyue Shen. Solving inverse problems with la-

tent diffusion models via hard data consistency. In *The Twelfth International Conference on Learning Representations*, . 2

[54] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *ArXiv*, abs/2010.02502, 2020. 1

[55] Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *International Conference on Learning Representations*, 2023. 2

[56] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Neural Information Processing Systems*, 2019. 2

[57] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33:12438–12448, 2020. 3

[58] 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*, . 2, 3

[59] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning*, 2023. 2, 3, 5

[60] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23:1661–1674, 2011. 3

[61] 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. 3

[62] 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. 2

[63] 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)*, pages 6613–6623, 2023. 3

[64] 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, 2024. 3, 5, 6, 8, 4

[65] 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. 2, 4

[66] Jiankun Zhao, Bowen Song, and Liyue Shen. Cosign: Few-step guidance of consistency model to solve general inverse problems. In *European Conference on Computer Vision*, pages 108–126. Springer, 2024. 2

[67] 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)*, pages 1219–1229, 2023. 2