

MODE-SEEKING FOR INVERSE PROBLEMS WITH DIFFUSION MODELS

Anonymous authors

Paper under double-blind review

ABSTRACT

A pre-trained unconditional diffusion model, combined with posterior sampling or maximum a posteriori (MAP) estimation techniques, can solve arbitrary inverse problems without task-specific training or fine-tuning. However, existing posterior sampling and MAP estimation methods often rely on modeling approximations and can be computationally demanding. In this work, we propose the variational mode-seeking loss (VML), which, when minimized during each reverse diffusion step, guides the generated sample towards the MAP estimate. VML arises from a novel perspective of minimizing the Kullback-Leibler (KL) divergence between the diffusion posterior $p(\mathbf{x}_0|\mathbf{x}_t)$ and the measurement posterior $p(\mathbf{x}_0|\mathbf{y})$, where \mathbf{y} denotes the measurement. Importantly, for linear inverse problems, VML can be analytically derived and need not be approximated. Based on further theoretical insights, we propose VML-MAP, an empirically effective algorithm for solving inverse problems, and validate its efficacy over existing methods in both performance and computational time, through extensive experiments on diverse image-restoration tasks across multiple datasets.

1 INTRODUCTION

Solving an inverse problem essentially involves estimation of the original data sample \mathbf{x} based on a given partially degraded measurement \mathbf{y} . Formally, Equation (1) relates the degraded measurement with the original data sample, where \mathcal{A} denotes the degradation operator and η is a random variable denoting measurement noise, which is typically assumed to be Gaussian distributed with known standard deviation σ_y , i.e., $\eta \sim \mathcal{N}(0, \sigma_y^2 \mathbf{I})$. This implies that $p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathcal{A}(\mathbf{x}), \sigma_y^2 \mathbf{I})$. For linear inverse problems, \mathcal{A} is linear and can be denoted with a matrix instead, i.e., $\mathcal{A}(\mathbf{x}) = \mathbf{H}\mathbf{x}$, where we use the matrix \mathbf{H} to denote a linear degradation operator throughout the paper.

$$\mathbf{y} = \mathcal{A}(\mathbf{x}) + \eta. \quad (1)$$

Inverse problems are commonly ill-posed, where many plausible data samples could correspond to a given degraded measurement, rendering a probabilistic approach essential. In a Bayesian framework, solving an inverse problem amounts to estimating (or sampling from) the posterior distribution $p(\mathbf{x}|\mathbf{y})$. As diffusion models (Ho et al., 2020; Song et al., 2021; Rombach et al., 2022) gain prominence in generative modeling, leveraging pre-trained unconditional diffusion models to solve inverse problems in a plug-and-play fashion is becoming increasingly attractive (Lugmayr et al., 2022; Kawar et al., 2022; Chung et al., 2023; Song et al., 2023a; Zhu et al., 2023; Rout et al., 2023; Song et al., 2024; Mardani et al., 2024; Janati et al., 2024; Gutha et al., 2025; Zhang et al., 2025; Moufad et al., 2025; Zilberstein et al., 2025). In this work, we advance this line of research by proposing a mode-seeking loss based inference-time guidance strategy for solving inverse problems with pre-trained unconditional diffusion models.

In a diffusion process, noise is added progressively to the samples of the input data distribution, i.e., $\mathbf{x}_0 \sim p_{data}(\mathbf{x}_0)$ to convert these into noisy samples gradually over a time horizon $t \in (0, T]$. This process can be reversed stochastically with a Stochastic Differential Equation (SDE) or deterministically with Probability Flow Ordinary Differential Equation (PF ODE), both of which require the score function of the marginal distribution at time t , i.e., $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$ for all $t \in (0, T]$ (Song et al., 2021). The score functions are typically intractable, so a diffusion model $s_\theta(\mathbf{x}_t, t)$ is trained using score-matching loss (Vincent, 2011; Song et al., 2020) to approximate these.

Conditional generation and Related works. To sample from a posterior $p(\mathbf{x}_0|\mathbf{y})$ using diffusion models, where \mathbf{y} is a given condition or measurement, it suffices to replace the unconditional score function $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$ with the conditional score $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y})$ in the reverse diffusion process mentioned above. Since $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}) = \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t)$, the former term can be replaced with the unconditional diffusion model $s_\theta(\mathbf{x}_t, t)$ but the latter term remains intractable due to the intractability of $p(\mathbf{y}|\mathbf{x}_t)$ (Chung et al., 2023), so several works (Chung et al., 2023; Song et al., 2023a; Peng et al., 2024; Boys et al., 2024; Zhang et al., 2025) rely on Gaussian approximations. Other approaches for posterior sampling (Trippe et al., 2022; Cardoso et al., 2023; Achituve et al., 2025) use sequential monte-carlo sampling techniques, while some others (Wang et al., 2024; Gutha et al., 2025; Xu et al., 2025) circumvent the need to estimate the conditional score by instead aiming for the MAP estimate, however, relying on modeling approximations which typically require solving the PF ODE (Song et al., 2021) in reverse time. This task is computationally feasible with a pre-trained consistency model (Song et al., 2023b). However, with only having access to a diffusion model, it quickly becomes prohibitively expensive, since it requires several neural function evaluations. To avoid this, in practice, a few-step Euler discretization is used to solve the PF ODE for MAP estimation (Wang et al., 2024; Gutha et al., 2025) or for posterior sampling in the case of Zhang et al. (2025). We refer to the survey by Daras et al. (2024) for a more comprehensive categorization of these methods. Unlike previous methods, our approach is based on minimizing a mode-seeking loss *at each reverse diffusion step*, which aligns the diffusion posterior with the measurement posterior.

Contributions. Our main contributions in this work are summarized below.

- We introduce the variational mode-seeking loss (VML), which, when minimized during each reverse diffusion step, steers the intermediate sample \mathbf{x}_t towards the MAP estimate as $t \rightarrow 0$.
- For linear inverse problems, we derive a closed-form expression for VML without any approximations and also demonstrate the redundancy of certain terms by further theoretical analysis.
- Based on the previous insight, we propose a practically effective algorithm (VML-MAP) for solving inverse problems, and also a preconditioner for ill-conditioned linear operators.
- We demonstrate VML-MAP’s effectiveness over other approaches through extensive experiments on diverse image-restoration tasks across multiple datasets.

2 BACKGROUND ON DIFFUSION MODELS

The forward process in a diffusion model corrupts a clean sample of the input data distribution i.e., $\mathbf{x}_0 \sim P(\mathbf{x}_0)$ into intermediate noisy sample \mathbf{x}_t , $t \in (0, T]$, modeled by the forward SDE given by Equation (2), where $\mathbf{f}(\cdot, t) : \mathbb{R}^n \rightarrow \mathbb{R}^n$, and $g : \mathbb{R} \rightarrow \mathbb{R}$ are the drift and diffusion coefficients, respectively, and \mathbf{w}_t denotes a standard Wiener process (Song et al., 2021). The drift, diffusion coefficients, and T are chosen such that the distribution of \mathbf{x}_T is tractable to sample from and is typically independent of the input data distribution.

$$d\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t, t)dt + g(t)d\mathbf{w}_t. \quad (2)$$

The reverse SDE in Equation (3) converts a noisy sample \mathbf{x}_t into a clean data sample \mathbf{x}_0 , where $\bar{\mathbf{w}}_t$ denotes a standard Wiener process in reverse time. $p(\mathbf{x}_t)$ denotes the marginal distribution at time t . Its score function, i.e., $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$, is usually intractable, so a neural network $s_\theta(\mathbf{x}_t, t)$ is trained using score-matching loss (Vincent, 2011; Song et al., 2020) to approximate this for all t .

$$d\mathbf{x}_t = \{\mathbf{f}(\mathbf{x}_t, t) - g^2(t)\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)\}dt + g(t)d\bar{\mathbf{w}}_t. \quad (3)$$

For a given choice of \mathbf{f} , g , the PF ODE in Equation (4) describes a deterministic process where, an intermediate sample \mathbf{x}_t generated by the ODE share the same marginal probability $p(\mathbf{x}_t)$ as that simulated by the forward SDE for all $t \in (0, T]$.

$$d\mathbf{x}_t = \{\mathbf{f}(\mathbf{x}_t, t) - \frac{1}{2}g^2(t)\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)\}dt. \quad (4)$$

To sample $\mathbf{x}_0 \sim p(\mathbf{x}_0)$, we first sample $\mathbf{x}_T \sim p(\mathbf{x}_T)$ and solve either the reverse SDE with SDE solvers or the PF ODE with ODE solvers using the learned score function $s_\theta(\mathbf{x}_t, t)$. Corresponding to the Variance Exploding (VE) SDE formulation from Song et al. (2021), throughout this paper, we fix $\mathbf{f}(\mathbf{x}_t, t) = \mathbf{0}$, $g(t) = \sqrt{d\sigma^2(t)/dt}$, where $\sigma(t)$ denotes the noise schedule for $t \in [0, T]$. We use $\sigma(t)$ and σ_t interchangeably to denote the noise level at time t . With the above choice of \mathbf{f} , and g , the corresponding perturbation kernel is given by $p(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_0, \sigma_t^2\mathbf{I})$, and $p(\mathbf{x}_T) \approx \mathcal{N}(0, \sigma_T^2\mathbf{I})$. Note that setting $\sigma(t) = t$ recovers the case of EDM preconditioning (Karras et al., 2022).

3 VARIATIONAL MODE-SEEKING LOSS

Motivation. Given a measurement y , and an unconditional diffusion model that generates samples from $p(\mathbf{x}_0)$, we are primarily interested in finding the MAP estimate, i.e., $\arg \max_{\mathbf{x}_0} \log p(\mathbf{x}_0|y)$. The main motivation for VML stems from the following observation. Starting the reverse diffusion process from a fixed noisy sample \mathbf{x}_t at time t results in a distribution over \mathbf{x}_0 , i.e., $p(\mathbf{x}_0|\mathbf{x}_t)$. If we find an optimal \mathbf{x}_t^* such that $p(\mathbf{x}_0|\mathbf{x}_t^*)$ shares modes, i.e., high-density regions, with the posterior $p(\mathbf{x}_0|y)$, then, by starting the reverse diffusion process from \mathbf{x}_t^* at time t , one may expect to generate a probable sample of the posterior. If we repeat the task of finding such \mathbf{x}_t^* at each diffusion time step t , then, as $t \rightarrow 0$, \mathbf{x}_t^* converges¹ to the MAP estimate as explained in the rest of this section.

For a fixed value of \mathbf{x}_t , say $\mathbf{x}_t = \gamma$, the behavior of $p(\mathbf{x}_0|\mathbf{x}_t = \gamma)$ along various time steps of a diffusion process is shown in Figure 1. As $p(\mathbf{x}_0|\mathbf{x}_t = \gamma) \propto p(\mathbf{x}_t = \gamma|\mathbf{x}_0)p(\mathbf{x}_0)$, and with $p(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_0, \sigma_t^2 \mathbf{I})$, the distribution $p(\mathbf{x}_0|\mathbf{x}_t = \gamma)$ is essentially proportional to the product of $p(\mathbf{x}_0)$ and a Gaussian kernel with variance σ_t^2 , centered at γ (due to the symmetrical form of $p(\mathbf{x}_t|\mathbf{x}_0)$). Since $\sigma_t \rightarrow 0$ as $t \rightarrow 0$, the dependence of $p(\mathbf{x}_0|\mathbf{x}_t = \gamma)$ on γ grows stronger as t decreases, with $p(\mathbf{x}_0|\mathbf{x}_t = \gamma)$ converging to the Dirac delta function $\delta(\mathbf{x}_0 - \gamma)$ as $t \rightarrow 0$, for any γ .

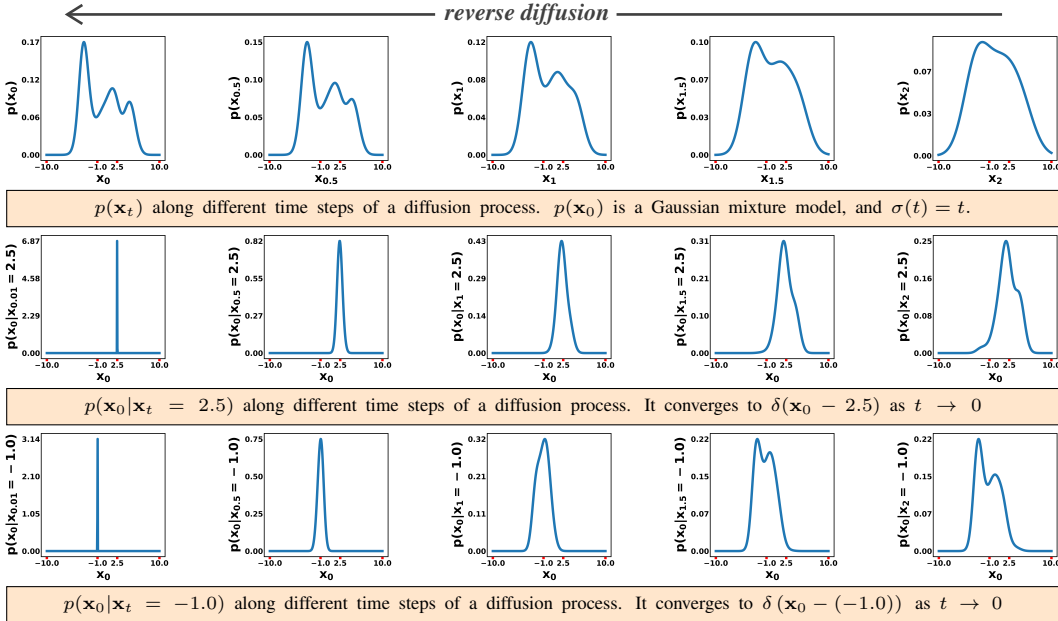


Figure 1: The figure depicts how the functional form of $p(\mathbf{x}_0|\mathbf{x}_t)$ gets peaky around \mathbf{x}_t as $t \rightarrow 0$.

Suppose, at each time step t , we find an optimal γ_t^* such that $p(\mathbf{x}_0|\mathbf{x}_t = \gamma_t^*)$ shares modes i.e., high-density regions with $p(\mathbf{x}_0|y)$. Note that for t arbitrarily close to 0, $p(\mathbf{x}_0|\mathbf{x}_t = \gamma_t^*)$ is an extremely peaky distribution around γ_t^* . For this distribution to share modes with the posterior $p(\mathbf{x}_0|y)$, it is ideal for γ_t^* to be closer, and converging to the MAP estimate as $t \rightarrow 0$, since the MAP estimate is the highest posterior mode, i.e., the sample with the highest posterior probability density.

Formalism. From a variational perspective, at a diffusion time step t , $p(\mathbf{x}_0|\mathbf{x}_t)$ is a parameterized distribution of \mathbf{x}_t . During each reverse diffusion time step t , we aim to find a specific distribution $p(\mathbf{x}_0|\mathbf{x}_t^*)$ from the class of parameterized distributions $\{p(\mathbf{x}_0|\mathbf{x}_t)\}_{\mathbf{x}_t}$ such that $p(\mathbf{x}_0|\mathbf{x}_t^*)$ shares modes i.e., high-density regions with the posterior. The functional form of $p(\mathbf{x}_0|\mathbf{x}_t)$ getting arbitrarily peaky as $t \rightarrow 0$ implies that the optimal \mathbf{x}_t^* ideally converges to the MAP estimate as $t \rightarrow 0$. The reverse KL divergence is known to promote this mode-matching behavior of distributions, so we choose $D_{\text{KL}}(p(\mathbf{x}_0|\mathbf{x}_t)||p(\mathbf{x}_0|y))$ as the minimization objective at each time step, which we refer to as the variational mode-seeking loss (VML). In practice, however, finding the exact MAP estimate is extremely challenging, as the VML can be highly non-convex, rendering optimization approaches ineffective. Instead, we settle for the modes of the posterior found by the VML optimizer in practice.

¹May require assumptions on $p(\mathbf{x}_t)$ and the convergence of $p(\mathbf{x}_0|\mathbf{x}_t)$. Note that $p(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(0, \sigma_t^2 \mathbf{I})$

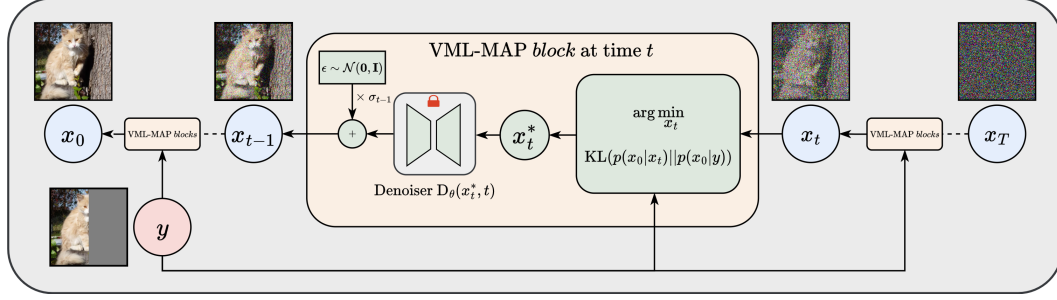


Figure 2: VML-MAP overview. Initially, \mathbf{x}_T is sampled from Gaussian noise and passed to a VML-MAP block, optimizing for \mathbf{x}_T^* that minimizes $D_{\text{KL}}(p(\mathbf{x}_0|\mathbf{x}_T)||p(\mathbf{x}_0|\mathbf{y}))$ followed by a standard reverse diffusion step to output \mathbf{x}_{T-1} . The process is repeated for each diffusion step until $t = 0$.

Proposition 1. The variational mode-seeking loss (VML) at diffusion time t , for a degradation operator \mathcal{A} , measurement \mathbf{y} , and measurement noise variance σ_y^2 is given by

$$\begin{aligned} \text{VML} = D_{\text{KL}}(p(\mathbf{x}_0|\mathbf{x}_t)||p(\mathbf{x}_0|\mathbf{y})) &= -\log p(\mathbf{x}_t) - \frac{\|D(\mathbf{x}_t, t) - \mathbf{x}_t\|^2}{2\sigma_t^2} - \frac{1}{2\sigma_t^2} \text{Tr} \{ \text{Cov}[\mathbf{x}_0|\mathbf{x}_t] \} \\ &+ \frac{1}{2\sigma_y^2} \left(-2\mathbf{y}^\top \int_{\mathbf{x}_0} \mathcal{A}(\mathbf{x}_0)p(\mathbf{x}_0|\mathbf{x}_t)d\mathbf{x}_0 + \int_{\mathbf{x}_0} \|\mathcal{A}(\mathbf{x}_0)\|^2 p(\mathbf{x}_0|\mathbf{x}_t)d\mathbf{x}_0 \right) + C \end{aligned}$$

where C is a constant, independent of \mathbf{x}_t . Tr denotes the matrix trace, Cov denotes the covariance matrix, and $D(\cdot, \cdot)$ denotes the true denoiser (see Appendix A.1).

Proposition 2. The variational mode-seeking loss (VML) at diffusion time t , for a **linear** degradation matrix \mathbf{H} , measurement \mathbf{y} , and measurement noise variance σ_y^2 is given by

$$\begin{aligned} \text{VML} = D_{\text{KL}}(p(\mathbf{x}_0|\mathbf{x}_t)||p(\mathbf{x}_0|\mathbf{y})) &= -\log p(\mathbf{x}_t) - \frac{\|D(\mathbf{x}_t, t) - \mathbf{x}_t\|^2}{2\sigma_t^2} - \frac{1}{2\sigma_t^2} \text{Tr} \{ \text{Cov}[\mathbf{x}_0|\mathbf{x}_t] \} \\ &+ \underbrace{\frac{\|\mathbf{y} - \mathbf{H}D(\mathbf{x}_t, t)\|^2}{2\sigma_y^2}}_{\text{measurement consistency}} + \frac{1}{2\sigma_y^2} \text{Tr} \{ \mathbf{H} \text{Cov}[\mathbf{x}_0|\mathbf{x}_t] \mathbf{H}^\top \} + C \end{aligned}$$

where C is a constant, independent of \mathbf{x}_t . Tr denotes the matrix trace, Cov denotes the covariance matrix, and $D(\cdot, \cdot)$ denotes the true denoiser (see Appendix A.1).

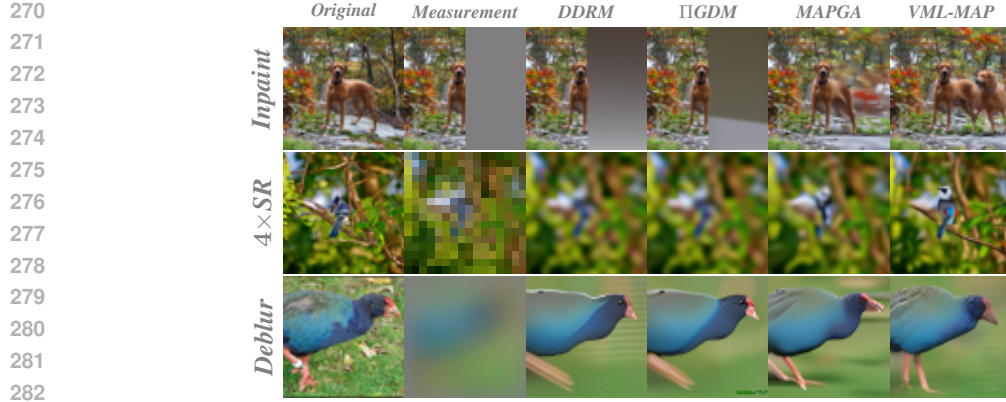
Proofs are provided in Appendix A.4. In the case of a linear degradation operator, VML has a measurement consistency term (see Proposition 2) which resembles the widely used approximation of the guidance term $\log p(\mathbf{y}|\mathbf{x}_t)$ in the literature (Chung et al., 2023; Song et al., 2023a), typically in the context of posterior sampling. However, VML arrives at this term without any modeling approximation, and in the context of MAP estimation. The remaining terms within VML are referred to as the prior terms, as they do not involve the measurement \mathbf{y} .

Note that the higher-order terms of VML involving $\text{Cov}[\mathbf{x}_0|\mathbf{x}_t]$ are computationally demanding, especially when VML has to be differentiable, to use gradient-based optimization for minimization. Based on further theoretical insights from Appendix B, we hypothesize that these higher-order terms may not be crucial in practice, and propose the simplified VML (see Equation 5) for linear inverse problems denoted as VML_S , where the higher-order terms are excluded.

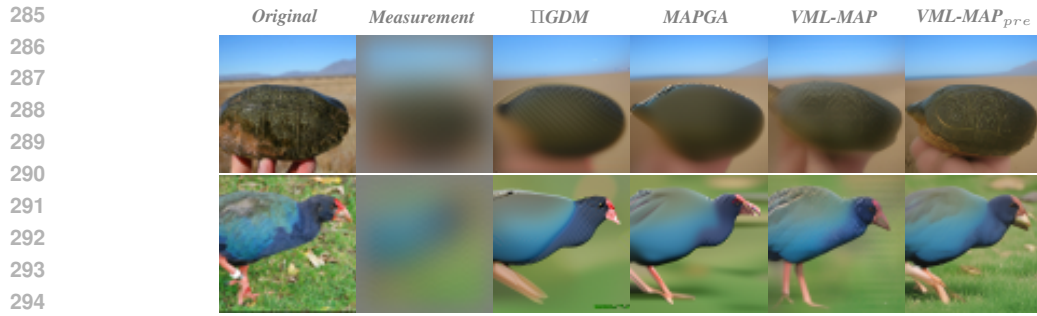
$$\text{VML}_S = \frac{\|\mathbf{y} - \mathbf{H}D(\mathbf{x}_t, t)\|^2}{2\sigma_y^2} - \log p(\mathbf{x}_t) - \frac{\|D(\mathbf{x}_t, t) - \mathbf{x}_t\|^2}{2\sigma_t^2} + C \quad (5)$$

$$\nabla_{\mathbf{x}_t} \text{VML}_S = \underbrace{-\frac{\partial D^\top(\mathbf{x}_t, t)}{\partial \mathbf{x}_t} \mathbf{H}^\top (\mathbf{y} - \mathbf{H}D(\mathbf{x}_t, t))}{\text{measurement consistency gradient}} \underbrace{- \frac{\partial D^\top(\mathbf{x}_t, t)}{\partial \mathbf{x}_t} (D(\mathbf{x}_t, t) - \mathbf{x}_t)}_{\text{prior gradient}} \quad (6)$$

Equation (6) shows the gradient of VML_S , which we use in practice during optimization. Figure 2 provides an overview, and Algorithm 1 shows the exact implementation details of our proposed



283 Figure 3: Half-mask inpainting, 4× super-resolution, and deblurring tasks from Table 1.



298 Figure 4: Deblurring task from Table 1. Zoom in for the best view.

299 Decomposition (SVD) of the linear degradation matrix H to find the pseudoinverse of terms involved
300 therein, VML-MAP only requires the forward operation of H . However, when H is ill-conditioned,
301 VML-MAP can struggle with the optimization, as seems to be the case with deblurring. In the next
302 section, we introduce a preconditioner to alleviate this problem, which also requires the SVD of H .

303 4.2 PRECONDITIONER

304 Preconditioners help accelerate convergence and thereby improve the effectiveness of the optimizer,
305 especially when dealing with ill-conditioned loss objectives. Assuming a linear degradation ma-
306 trix, H with SVD given by $H = U\Sigma V^T$ (U, V denote the left and right singular orthogonal ma-
307 trices respectively, with Σ denoting the singular values matrix), we use the preconditioner P in
308 Equation (7) to essentially replace the gradient $\nabla_{\mathbf{x}_t} \text{VML}_S$ in VML-MAP with the preconditioned
309 gradient $P\nabla_{\mathbf{x}_t} \text{VML}_S$. We refer to this method as VML-MAP_{pre}, also presented in Algorithm 1.

$$310 \quad P = \left(\frac{\partial D^T(\mathbf{x}_t, t)}{\partial \mathbf{x}_t} \right) M^{-1} \left(\frac{\partial D^T(\mathbf{x}_t, t)}{\partial \mathbf{x}_t} \right)^{-1} \quad (7)$$

$$311 \quad M = (\mathbf{I} - \Sigma^+ \Sigma) + H^T H \quad (8)$$

$$312 \quad \nabla_{\mathbf{x}_t} \text{VML}_{S_{pre}} = -\frac{\partial D^T(\mathbf{x}_t, t)}{\partial \mathbf{x}_t} M^{-1} \frac{H^T (\mathbf{y} - H D(\mathbf{x}_t, t))}{\sigma_y^2} - \frac{\partial D^T(\mathbf{x}_t, t)}{\partial \mathbf{x}_t} M^{-1} \frac{(D(\mathbf{x}_t, t) - \mathbf{x}_t)}{\sigma_t^2} \quad (9)$$

313 Σ^+ above denotes the pseudoinverse of Σ . Equation (9) further expands the preconditioned gradient
314 $P\nabla_{\mathbf{x}_t} \text{VML}_S$ (that we hereon denote with $\nabla_{\mathbf{x}_t} \text{VML}_{S_{pre}}$). Note that $\frac{\partial D^T(\mathbf{x}_t, t)}{\partial \mathbf{x}_t} = \sigma_t^2 \text{Cov}[\mathbf{x}_0 | \mathbf{x}_t]$ (see
315 Appendix A.2). Assuming $\text{Cov}[\mathbf{x}_0 | \mathbf{x}_t] \succ 0$ (i.e., positive definite), implies the positive definiteness
316 and hence the invertibility of $\frac{\partial D^T(\mathbf{x}_t, t)}{\partial \mathbf{x}_t}$, which further implies the invertibility of P .

317 We use the SVD of H to compute M^{-1} efficiently, which in turn is used to compute $\nabla_{\mathbf{x}_t} \text{VML}_{S_{pre}}$
318 for VML-MAP_{pre} (see Equation 9). Note that DDRM, PIGDM, and MAPGA also require SVD of

H, which makes it a fair comparison against VML-MAP_{pre}. Results from Table 1 also indicate the effectiveness of the preconditioner on 4× super-resolution and deblurring tasks as VML-MAP_{pre} shows significant improvements in LPIPS and FID over VML-MAP and other baselines, denoting higher perceptual quality of the restored images. For inpainting, VML-MAP and VML-MAP_{pre} are essentially equivalent since $M = P = I$. Figure 4 presents a qualitative comparison of the restored samples with different baselines, VML-MAP, and VML-MAP_{pre} for the deblurring task.

5 MAIN EXPERIMENTS AND RESULTS

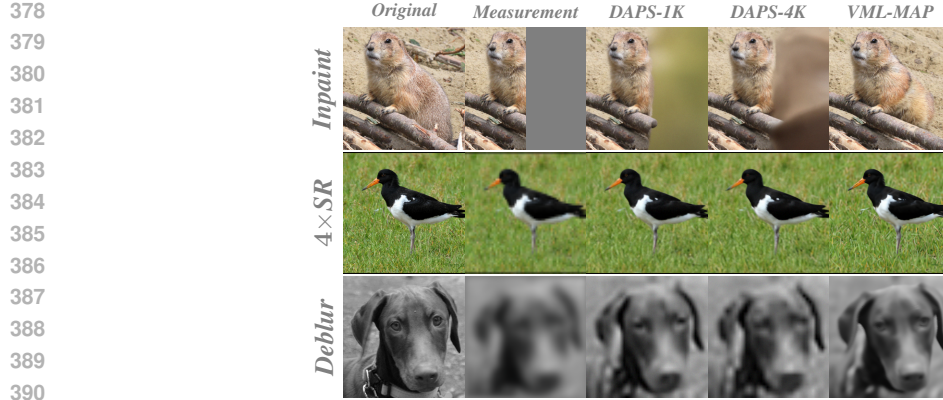
Our experiments in Table 2 include half-mask inpainting, 4× super-resolution, and the deblurring tasks previously mentioned in Section 4.1. In Table 3, we focus on the image inpainting task with several masks. In all our experiments, we evaluate on 100 validation images of ImageNet (Deng et al., 2009) with a resolution of 256×256 , using the unconditional ImageNet256 pre-trained diffusion model from Ho et al. (2020), and on 100 images of FFHQ (Karras et al., 2019), with a resolution of 256×256 , using the FFHQ256 pre-trained diffusion model from Chung et al. (2023). With FID and LPIPS as evaluation metrics, we compare VML-MAP and VML-MAP_{pre} against several baselines such as DDRM (Kawar et al., 2022), IIGDM (Song et al., 2023a), MAPGA (Gutha et al., 2025), and DAPS (Zhang et al., 2025). In all our experiments, we fix a budget of approximately 1000 neural function evaluations of the diffusion model for both VML-MAP and VML-MAP_{pre}. For DAPS, we consider two configurations, with 1000, 4000 neural function evaluations denoted DAPS-1K, DAPS-4K, respectively, see Zhang et al. (2025). We refer to Appendix C for further details regarding the experiment setup, hyperparameters, and runs across different seeds.

Table 2: Evaluation of several image restoration methods on noiseless inverse problems of half-mask inpainting, 4× super-resolution, and deblurring on 100 validation images of ImageNet256, and on 100 images of FFHQ256. Best values in **bold**, second best values underlined.

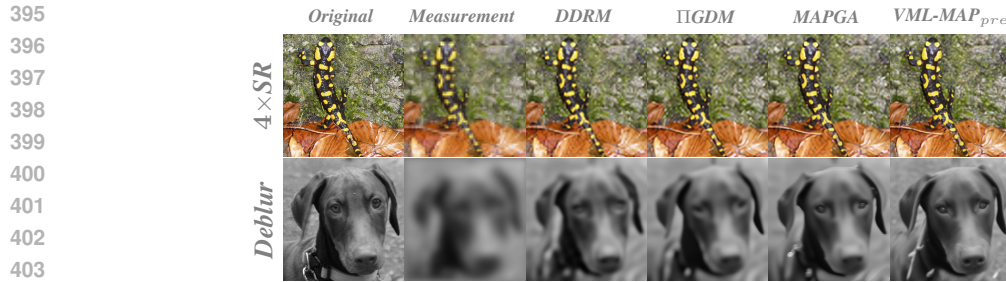
| Dataset | Method | Inpainting | | 4× Super-res | | Deblurring | |
|----------|------------------------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | | LPIPS↓ | FID↓ | LPIPS↓ | FID↓ | LPIPS↓ | FID↓ |
| ImageNet | DDRM | 0.391 | 102.8 | 0.289 | 90.50 | 0.618 | 234.6 |
| | IIGDM | 0.373 | 103.8 | 0.292 | 83.80 | 0.560 | 231.3 |
| | MAPGA | <u>0.289</u> | <u>83.13</u> | 0.274 | 76.13 | <u>0.459</u> | <u>194.7</u> |
| | DAPS-1K | 0.385 | 98.72 | 0.256 | 73.33 | 0.607 | 220.3 |
| | DAPS-4K | 0.371 | 94.98 | 0.243 | 69.98 | 0.593 | 224.4 |
| | VML-MAP | 0.265 | 69.21 | 0.192 | <u>59.91</u> | 0.511 | 205.2 |
| | VML-MAP_{pre} | 0.265 | 69.21 | <u>0.194</u> | 56.60 | 0.371 | 166.7 |
| FFHQ | DDRM | 0.243 | 71.21 | 0.154 | 70.47 | 0.307 | 117.5 |
| | IIGDM | 0.234 | 71.02 | 0.147 | 68.65 | 0.293 | 114.3 |
| | MAPGA | <u>0.206</u> | 65.29 | 0.132 | 64.45 | 0.235 | 120.9 |
| | DAPS-1K | 0.232 | <u>59.93</u> | 0.113 | 61.59 | 0.259 | 100.2 |
| | DAPS-4K | 0.223 | 60.20 | <u>0.100</u> | 58.38 | <u>0.230</u> | <u>95.05</u> |
| | VML-MAP | 0.180 | 51.94 | 0.099 | <u>56.58</u> | 0.247 | 99.84 |
| | VML-MAP_{pre} | 0.180 | 51.94 | 0.099 | 52.04 | 0.183 | 93.99 |

The quantitative results from Table 2 and the corresponding qualitative comparisons from Figures 5 and 6 highlight the effectiveness of VML-MAP and VML-MAP_{pre} over existing methods. Also from Figure 7, which shows the tradeoff between runtime and perceptual quality of reconstructed images for several baselines, VML-MAP achieves better perceptual quality with lower runtime than other methods, highlighting its computational efficiency. Also, VML-MAP_{pre} has an almost similar runtime as VML-MAP, as we compute M^{-1} in Equation (9) with negligible overhead using SVD.

As mentioned in Section 4.2, DDRM, IIGDM, MAPGA, and VML-MAP_{pre} require SVD of H, while DAPS and VML-MAP only require the forward operation of H. To ensure a fair comparison



391 Figure 5: Half-mask inpainting, 4 \times super-resolution, and deblurring tasks from Table 2. Note that
392 DAPS-1K, DAPS-4K, and VML-MAP only require the forward operation of the linear degradation
393 operator and not its SVD. Zoom in for the best view.
394



405 Figure 6: 4 \times Super-resolution, and Deblurring tasks from Table 2. Note that DDRM, PIGDM,
406 MAPGA, and VML-MAP_{pre} require the SVD of the linear degradation operator.
407

408 Table 3: Evaluation of several image restoration methods on noiseless inpainting with expand mask,
409 box mask, super-resolution mask, and random mask on 100 validation images of ImageNet256, and
410 on 100 images of FFHQ256. Best values in **bold**, second best values underlined.
411

| Dataset | Method | Expand mask | | Box mask | | Sup-res mask | | Random mask | |
|----------|----------------|--------------------|------------------|--------------------|------------------|--------------------|------------------|--------------------|------------------|
| | | LPIPS \downarrow | FID \downarrow | LPIPS \downarrow | FID \downarrow | LPIPS \downarrow | FID \downarrow | LPIPS \downarrow | FID \downarrow |
| ImageNet | DDRM | 0.548 | 152.0 | 0.211 | 133.1 | 0.089 | 29.24 | 0.052 | 21.81 |
| | PIGDM | 0.523 | 155.7 | 0.203 | 123.9 | <u>0.080</u> | 26.54 | 0.054 | 24.48 |
| | MAPGA | <u>0.466</u> | <u>126.9</u> | <u>0.150</u> | <u>91.14</u> | 0.087 | <u>25.02</u> | <u>0.051</u> | <u>20.31</u> |
| | DAPS-1K | 0.550 | 167.2 | 0.201 | 112.6 | 0.132 | 53.46 | 0.092 | 35.93 |
| | DAPS-4K | 0.521 | 154.0 | 0.187 | 102.8 | 0.114 | 45.65 | 0.082 | 31.67 |
| | VML-MAP | 0.434 | 116.2 | 0.138 | 75.80 | 0.068 | 19.96 | 0.044 | 16.14 |
| FFHQ | DDRM | 0.426 | 151.1 | 0.087 | 50.60 | <u>0.031</u> | 20.75 | <u>0.027</u> | <u>18.79</u> |
| | PIGDM | 0.415 | 146.3 | 0.084 | 46.95 | 0.033 | 23.98 | <u>0.027</u> | 19.93 |
| | MAPGA | <u>0.393</u> | 129.3 | <u>0.070</u> | 41.34 | 0.036 | 27.01 | <u>0.027</u> | 20.88 |
| | DAPS-1K | 0.423 | <u>126.7</u> | 0.076 | <u>33.17</u> | 0.057 | 47.21 | 0.059 | 45.55 |
| | DAPS-4K | 0.398 | 117.5 | 0.077 | 33.82 | 0.051 | 42.32 | 0.051 | 40.06 |
| | VML-MAP | 0.365 | 112.9 | 0.057 | 28.38 | 0.027 | <u>21.40</u> | 0.022 | 16.30 |

428
429
430
431

in this regard, we evaluate all the methods on the image inpainting task with several masks, where the SVD of H is trivial. The quantitative results from Table 3 and the qualitative comparison in Appendix C.3 reveal the superior performance of VML-MAP in all the inpainting tasks.

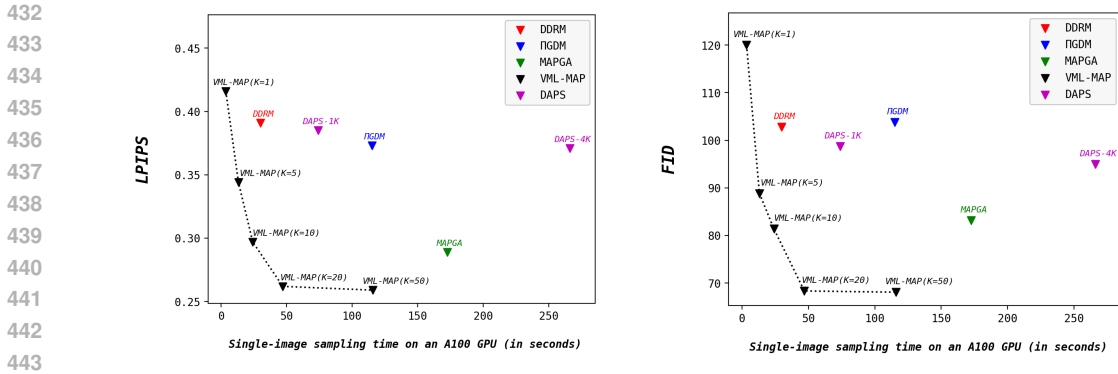


Figure 7: Runtime vs Perceptual quality for the half-mask inpainting experiment in Table 2. DDRM and IIGDM use 500 and 1000 reverse diffusion steps, respectively, to achieve their best results. For VML-MAP, we fix the reverse diffusion steps (N) to 20, and vary the number of gradient-descent iterations per step (K) across $\{1, 5, 10, 20, 50\}$. See Appendix C.1 for more details. VML-MAP achieves better perceptual quality with lower compute than other methods.

Note that the effectiveness of the preconditioner for $4\times$ super-resolution and deblurring tasks reinforces the need for efficient optimizers in boosting the performance of VML-MAP in practice. In Appendix D, we extend VML-MAP to Latent Diffusion Models (LDM), and compare with other baselines. In practice, even for linear inverse problems, we observe that the non-linearity of the LDM decoder makes the optimization highly challenging, which again highlights the need for better optimizers. By treating the LDM encoder and the decoder as identity mappings, this can also serve as an extension of VML-MAP to non-linear inverse problems using pixel diffusion models.

6 CONCLUSION

In this work, we proposed a training-free guidance method that steers the intermediate sample \mathbf{x}_t of an unconditional diffusion model towards the MAP estimate, i.e., $\arg \max_{\mathbf{x}_0} \log p(\mathbf{x}_0|\mathbf{y})$, for a given measurement \mathbf{y} , thereby enabling the solution of downstream inverse problems. The core of our approach is a novel formulation based on minimizing the KL divergence between $p(\mathbf{x}_0|\mathbf{x}_t)$ and $p(\mathbf{x}_0|\mathbf{y})$, which we define as the variational mode-seeking loss (VML). We derived VML in a closed form for linear inverse problems without any modeling approximations and use it within our proposed algorithm (VML-MAP), which optimizes VML at each reverse diffusion step. To address the optimization difficulties arising from ill-conditioned linear degradation operators, we proposed a preconditioned variant (VML-MAP_{pre}) that offers a simple yet effective remedy. Finally, we demonstrated the effectiveness of our approach through extensive experiments on several image restoration inverse problems across multiple real-world datasets.

7 LIMITATIONS

Although this paper primarily focuses on developing a principled framework and establishing the theoretical foundations of VML, the availability of a practically effective optimizer for minimizing the VML objective is equally critical. In our experiments, we found that gradient descent performs sufficiently well to validate the proposed framework empirically. Nonetheless, approximate higher-order methods and advanced optimization strategies have the potential to improve performance further, as illustrated by our proposed preconditioner for ill-conditioned linear inverse problems. We also observed that the VML objective exhibits notable sensitivity to measurement noise (σ_y) in practice. For instance, in inpainting, while increasing σ_y from 0.001 to 0.01 still preserves most of the perceptual content of \mathbf{y} , the measurement consistency term in VML is downweighted by a factor of 100, which in turn introduces blurry artifacts in the reconstructed images. In the case of LDMs, the nonlinearity of the decoder exacerbates these optimization challenges, even when addressing linear inverse problems. Importantly, while advanced optimization techniques may enhance performance, they must not come at the expense of prohibitive computational costs. Designing optimizers that are both efficient and practically feasible remains an essential direction for future work.

REFERENCES

- 486
487
488 Idan Achituve, Hai Victor Habi, Amir Rosenfeld, Arnon Netzer, Idit Diamant, and Ethan Fe-
489 taya. Inverse problem sampling in latent space using sequential monte carlo. *arXiv preprint*
490 *arXiv:2502.05908*, 2025.
- 491 Benjamin Boys, Mark Girolami, Jakiw Pidstrigach, Sebastian Reich, Alan Mosca, and Omer Deniz
492 Akyildiz. Tweedie moment projected diffusions for inverse problems. *Transactions on Machine*
493 *Learning Research*, 2024. ISSN 2835-8856. URL [https://openreview.net/forum?](https://openreview.net/forum?id=4unJi0qrTE)
494 [id=4unJi0qrTE](https://openreview.net/forum?id=4unJi0qrTE). Featured Certification.
- 495 Gabriel Cardoso, Yazid Janati El Idrissi, Sylvain Le Corff, and Eric Moulines. Monte carlo guided
496 diffusion for bayesian linear inverse problems. *arXiv preprint arXiv:2308.07983*, 2023.
- 497
498 Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul
499 Ye. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh Interna-*
500 *tional Conference on Learning Representations*, 2023. URL [https://openreview.net/](https://openreview.net/forum?id=OnD9zGAGT0k)
501 [forum?id=OnD9zGAGT0k](https://openreview.net/forum?id=OnD9zGAGT0k).
- 502 Giannis Daras, Hyungjin Chung, Chieh-Hsin Lai, Yuki Mitsufuji, Jong Chul Ye, Peyman Milan-
503 far, Alexandros G Dimakis, and Mauricio Delbracio. A survey on diffusion models for inverse
504 problems. *arXiv preprint arXiv:2410.00083*, 2024.
- 505
506 Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hi-
507 erarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*,
508 pp. 248–255. Ieee, 2009.
- 509 Sai Bharath Chandra Gutha, Ricardo Vinuesa, and Hossein Azizpour. Inverse problems with diffu-
510 sion models: A map estimation perspective. In *2025 IEEE/CVF Winter Conference on Applica-*
511 *tions of Computer Vision (WACV)*, pp. 4153–4162. IEEE, 2025.
- 512 Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in*
513 *neural information processing systems*, 33:6840–6851, 2020.
- 514
515 Yazid Janati, Badr Moufad, Alain Durmus, Eric Moulines, and Jimmy Olsson. Divide-and-conquer
516 posterior sampling for denoising diffusion priors. *Advances in Neural Information Processing*
517 *Systems*, 37:97408–97444, 2024.
- 518 Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative
519 adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern*
520 *recognition*, pp. 4401–4410, 2019.
- 521
522 Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-
523 based generative models. *Advances in neural information processing systems*, 35:26565–26577,
524 2022.
- 525 Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration
526 models. *Advances in neural information processing systems*, 35:23593–23606, 2022.
- 527
528 Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild.
529 In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.
- 530 Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool.
531 Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the*
532 *IEEE/CVF conference on computer vision and pattern recognition*, pp. 11461–11471, 2022.
- 533
534 Morteza Mardani, Jiaming Song, Jan Kautz, and Arash Vahdat. A variational perspective on solving
535 inverse problems with diffusion models. In *The Twelfth International Conference on Learning*
536 *Representations*, 2024. URL <https://openreview.net/forum?id=1Y04EE3SPB>.
- 537 Badr Moufad, Yazid Janati, Lisa Bedin, Alain Oliviero Durmus, randal douc, Eric Moulines,
538 and Jimmy Olsson. Variational diffusion posterior sampling with midpoint guidance. In *The*
539 *Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=6EUtjXAvmj>.

- 540 Xinyu Peng, Ziyang Zheng, Wenrui Dai, Nuoqian Xiao, Chenglin Li, Junni Zou, and Hongkai
541 Xiong. Improving diffusion models for inverse problems using optimal posterior covariance. In
542 *International Conference on Machine Learning*, pp. 40347–40370. PMLR, 2024.
- 543 Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-
544 resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF confer-
545 ence on computer vision and pattern recognition*, pp. 10684–10695, 2022.
- 547 Litu Rout, Negin Raoof, Giannis Daras, Constantine Caramanis, Alex Dimakis, and Sanjay Shakkot-
548 tai. Solving linear inverse problems provably via posterior sampling with latent diffusion mod-
549 els. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL
550 <https://openreview.net/forum?id=XKBFdYwfRo>.
- 551 Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng
552 Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual
553 recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.
- 554 Bowen Song, Soo Min Kwon, Zecheng Zhang, Xinyu Hu, Qing Qu, and Liyue Shen. Solving inverse
555 problems with latent diffusion models via hard data consistency. In *The Twelfth International
556 Conference on Learning Representations*, 2024. URL [https://openreview.net/forum?
557 id=j8hdRqOUhN](https://openreview.net/forum?id=j8hdRqOUhN).
- 559 Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion
560 models for inverse problems. In *International Conference on Learning Representations*, 2023a.
- 561 Yang Song, Sahaj Garg, Jiaxin Shi, and Stefano Ermon. Sliced score matching: A scalable approach
562 to density and score estimation. In *Uncertainty in artificial intelligence*, pp. 574–584. PMLR,
563 2020.
- 564 Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben
565 Poole. Score-based generative modeling through stochastic differential equations. In *Internat-
566 ional Conference on Learning Representations*, 2021. URL [https://openreview.net/
567 forum?id=PXTIG12RRHS](https://openreview.net/forum?id=PXTIG12RRHS).
- 569 Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International
570 Conference on Machine Learning*, pp. 32211–32252. PMLR, 2023b.
- 571 Brian L Trippe, Jason Yim, Doug Tischer, David Baker, Tamara Broderick, Regina Barzilay, and
572 Tommi Jaakkola. Diffusion probabilistic modeling of protein backbones in 3d for the motif-
573 scaffolding problem. *arXiv preprint arXiv:2206.04119*, 2022.
- 574 Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural compu-
575 tation*, 23(7):1661–1674, 2011.
- 577 Hengkang Wang, Xu Zhang, Taihui Li, Yuxiang Wan, Tiancong Chen, and Ju Sun. Dmplug: A plug-
578 in method for solving inverse problems with diffusion models. *Advances in Neural Information
579 Processing Systems*, 37:117881–117916, 2024.
- 580 Tongda Xu, Xiyan Cai, Xinjie Zhang, Xingtong Ge, Dailan He, Ming Sun, Jingjing Liu, Ya-Qin
581 Zhang, Jian Li, and Yan Wang. Rethinking diffusion posterior sampling: From conditional score
582 estimator to maximizing a posterior. In *The Thirteenth International Conference on Learning
583 Representations*, 2025. URL <https://openreview.net/forum?id=GcvLoqOoXL>.
- 584 Bingliang Zhang, Wenda Chu, Julius Berner, Chenlin Meng, Anima Anandkumar, and Yang Song.
585 Improving diffusion inverse problem solving with decoupled noise annealing. In *Proceedings of
586 the Computer Vision and Pattern Recognition Conference*, pp. 20895–20905, 2025.
- 588 Yuanzhi Zhu, Kai Zhang, Jingyun Liang, Jiezhong Cao, Bihan Wen, Radu Timofte, and Luc
589 Van Gool. Denoising diffusion models for plug-and-play image restoration. In *Proceedings of the
590 IEEE/CVF conference on computer vision and pattern recognition*, pp. 1219–1229, 2023.
- 591 Nicolas Zilberstein, Morteza Mardani, and Santiago Segarra. Repulsive latent score distillation for
592 solving inverse problems. In *The Thirteenth International Conference on Learning Representa-
593 tions*, 2025. URL <https://openreview.net/forum?id=bwJxUB0y46>.

A APPENDIX

A.1 TWEEDIE'S FORMULA

$$D(\mathbf{x}_t, t) = \mathbb{E}[\mathbf{x}_0|\mathbf{x}_t] = \int_{\mathbf{x}_0} \mathbf{x}_0 p(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 = \mathbf{x}_t + \sigma_t^2 \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) \approx \mathbf{x}_t + \sigma_t^2 S_\theta(\mathbf{x}_t, t)$$

where, $D(\mathbf{x}_t, t)$ denotes the true denoiser, and $S_\theta(\cdot, \cdot)$ denotes the learned score function.

Proof.

$$\begin{aligned} \nabla_{\mathbf{x}_t} p(\mathbf{x}_t) &= \nabla_{\mathbf{x}_t} \int_{\mathbf{x}_0} p(\mathbf{x}_t|\mathbf{x}_0) p(\mathbf{x}_0) d\mathbf{x}_0 = \int_{\mathbf{x}_0} p(\mathbf{x}_0) \nabla_{\mathbf{x}_t} p(\mathbf{x}_t|\mathbf{x}_0) d\mathbf{x}_0 \\ \nabla_{\mathbf{x}_t} p(\mathbf{x}_t) &= \int_{\mathbf{x}_0} p(\mathbf{x}_0) p(\mathbf{x}_t|\mathbf{x}_0) \frac{\mathbf{x}_0 - \mathbf{x}_t}{\sigma_t^2} d\mathbf{x}_0 \quad \{\text{Note that } p(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_0, \sigma_t^2 \mathbf{I})\} \\ \sigma_t^2 \nabla_{\mathbf{x}_t} p(\mathbf{x}_t) &= \left(\int_{\mathbf{x}_0} \mathbf{x}_0 p(\mathbf{x}_0) p(\mathbf{x}_t|\mathbf{x}_0) d\mathbf{x}_0 \right) - \mathbf{x}_t p(\mathbf{x}_t) \\ \mathbf{x}_t p(\mathbf{x}_t) + \sigma_t^2 \nabla_{\mathbf{x}_t} p(\mathbf{x}_t) &= \int_{\mathbf{x}_0} \mathbf{x}_0 p(\mathbf{x}_0) p(\mathbf{x}_t|\mathbf{x}_0) d\mathbf{x}_0 = p(\mathbf{x}_t) \mathbb{E}[\mathbf{x}_0|\mathbf{x}_t] \\ \mathbf{x}_t + \sigma_t^2 \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) &= \mathbb{E}[\mathbf{x}_0|\mathbf{x}_t] \approx \mathbf{x}_t + \sigma_t^2 S_\theta(\mathbf{x}_t, t) \end{aligned}$$

□

A.2 COVARIANCE FORMULA

$$\text{Cov}[\mathbf{x}_0|\mathbf{x}_t] = \int_{\mathbf{x}_0} (\mathbf{x}_0 - \mathbb{E}[\mathbf{x}_0|\mathbf{x}_t])(\mathbf{x}_0 - \mathbb{E}[\mathbf{x}_0|\mathbf{x}_t])^\top p(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 = \sigma_t^2 \frac{\partial D(\mathbf{x}_t, t)}{\partial \mathbf{x}_t} \approx \sigma_t^2 \frac{\partial D_\theta(\mathbf{x}_t, t)}{\partial \mathbf{x}_t}$$

where, $D(\mathbf{x}_t, t) = \mathbb{E}[\mathbf{x}_0|\mathbf{x}_t]$ is the true denoiser, and $D_\theta(\mathbf{x}_t, t)$ is the learned denoiser.

Proof.

$$\begin{aligned} D(\mathbf{x}_t, t) &= \mathbb{E}[\mathbf{x}_0|\mathbf{x}_t] = \int_{\mathbf{x}_0} p(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 \\ \frac{\partial D(\mathbf{x}_t, t)}{\partial \mathbf{x}_t} &= \frac{\partial}{\partial \mathbf{x}_t} \int_{\mathbf{x}_0} \mathbf{x}_0 p(\mathbf{x}_0|\mathbf{x}_t) = \int_{\mathbf{x}_0} \mathbf{x}_0 p(\mathbf{x}_0) \frac{\partial}{\partial \mathbf{x}_t} \left(\frac{p(\mathbf{x}_t|\mathbf{x}_0)}{p(\mathbf{x}_t)} \right) d\mathbf{x}_0 \\ &\quad \{\text{Note that } p(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_0, \sigma_t^2 \mathbf{I})\} \\ \frac{\partial D(\mathbf{x}_t, t)}{\partial \mathbf{x}_t} &= \int_{\mathbf{x}_0} \mathbf{x}_0 p(\mathbf{x}_0) \left(\frac{p(\mathbf{x}_t|\mathbf{x}_0) p(\mathbf{x}_t|\mathbf{x}_0) \frac{(\mathbf{x}_0 - \mathbf{x}_t)^\top}{\sigma_t^2} - p(\mathbf{x}_t|\mathbf{x}_0) \frac{\partial p(\mathbf{x}_t)}{\partial \mathbf{x}_t}}{p(\mathbf{x}_t)^2} \right) d\mathbf{x}_0 \\ \frac{\partial D(\mathbf{x}_t, t)}{\partial \mathbf{x}_t} &= \left(\int_{\mathbf{x}_0} \mathbf{x}_0 p(\mathbf{x}_0) \frac{p(\mathbf{x}_t|\mathbf{x}_0) \frac{(\mathbf{x}_0 - \mathbf{x}_t)^\top}{\sigma_t^2}}{p(\mathbf{x}_t)} d\mathbf{x}_0 \right) - \left(\int_{\mathbf{x}_0} \mathbf{x}_0 p(\mathbf{x}_0) \frac{p(\mathbf{x}_t|\mathbf{x}_0) \frac{\partial p(\mathbf{x}_t)}{\partial \mathbf{x}_t}}{p(\mathbf{x}_t)^2} d\mathbf{x}_0 \right) \\ \frac{\partial D(\mathbf{x}_t, t)}{\partial \mathbf{x}_t} &= \left(\int_{\mathbf{x}_0} \mathbf{x}_0 \frac{(\mathbf{x}_0 - \mathbf{x}_t)^\top}{\sigma_t^2} p(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 \right) - \left(\int_{\mathbf{x}_0} \mathbf{x}_0 \frac{\partial \log p(\mathbf{x}_t)}{\partial \mathbf{x}_t} p(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 \right) \\ \frac{\partial D(\mathbf{x}_t, t)}{\partial \mathbf{x}_t} &= \int_{\mathbf{x}_0} \mathbf{x}_0 \left(\frac{(\mathbf{x}_0 - \mathbf{x}_t)^\top}{\sigma_t^2} - \frac{\partial \log p(\mathbf{x}_t)}{\partial \mathbf{x}_t} \right) p(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 \\ \frac{\partial D(\mathbf{x}_t, t)}{\partial \mathbf{x}_t} &= \int_{\mathbf{x}_0} \mathbf{x}_0 \frac{(\mathbf{x}_0 - D(\mathbf{x}_t, t))^\top}{\sigma_t^2} p(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 \\ \sigma_t^2 \frac{\partial D(\mathbf{x}_t, t)}{\partial \mathbf{x}_t} &= \int_{\mathbf{x}_0} \mathbf{x}_0 (\mathbf{x}_0 - D(\mathbf{x}_t, t))^\top p(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 \\ \sigma_t^2 \frac{\partial D(\mathbf{x}_t, t)}{\partial \mathbf{x}_t} &= \int_{\mathbf{x}_0} (\mathbf{x}_0 - D(\mathbf{x}_t, t)) (\mathbf{x}_0 - D(\mathbf{x}_t, t))^\top p(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 \\ \sigma_t^2 \frac{\partial D(\mathbf{x}_t, t)}{\partial \mathbf{x}_t} &= \text{Cov}[\mathbf{x}_0|\mathbf{x}_t] \approx \sigma_t^2 \frac{\partial D_\theta(\mathbf{x}_t, t)}{\partial \mathbf{x}_t} \end{aligned}$$

□

648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701

A.3 LEMMAS

Lemma 1. $\int_{\mathbf{x}_0} \|\mathbf{x}_0\|^2 p(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 = \text{Tr} \{ \text{Cov}[\mathbf{x}_0|\mathbf{x}_t] \} + \|\mathbf{D}(\mathbf{x}_t, t)\|^2$

Proof.

$$\begin{aligned} \int_{\mathbf{x}_0} \|\mathbf{x}_0\|^2 p(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 &= \int_{\mathbf{x}_0} \text{Tr} \{ \mathbf{x}_0 \mathbf{x}_0^\top \} p(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 = \text{Tr} \left\{ \int_{\mathbf{x}_0} \mathbf{x}_0 \mathbf{x}_0^\top p(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 \right\} \\ \int_{\mathbf{x}_0} \|\mathbf{x}_0\|^2 p(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 &= \text{Tr} \left\{ \text{Cov}[\mathbf{x}_0|\mathbf{x}_t] + \mathbf{D}(\mathbf{x}_t, t) \mathbf{D}(\mathbf{x}_t, t)^\top \right\} \\ \int_{\mathbf{x}_0} \|\mathbf{x}_0\|^2 p(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 &= \text{Tr} \{ \text{Cov}[\mathbf{x}_0|\mathbf{x}_t] \} + \|\mathbf{D}(\mathbf{x}_t, t)\|^2 \end{aligned}$$

□

Lemma 2. $\int_{\mathbf{x}_0} \|\mathbf{H}\mathbf{x}_0\|^2 p(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 = \text{Tr} \{ \mathbf{H} \text{Cov}[\mathbf{x}_0|\mathbf{x}_t] \mathbf{H}^\top \} + \|\mathbf{H}\mathbf{D}(\mathbf{x}_t, t)\|^2$

Proof.

$$\begin{aligned} \int_{\mathbf{x}_0} \|\mathbf{H}\mathbf{x}_0\|^2 p(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 &= \int_{\mathbf{x}_0} \text{Tr} \{ \mathbf{H}\mathbf{x}_0 \mathbf{x}_0^\top \mathbf{H}^\top \} p(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 \\ \int_{\mathbf{x}_0} \|\mathbf{H}\mathbf{x}_0\|^2 p(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 &= \text{Tr} \left\{ \mathbf{H} \left(\int_{\mathbf{x}_0} \mathbf{x}_0 \mathbf{x}_0^\top p(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 \right) \mathbf{H}^\top \right\} \\ \int_{\mathbf{x}_0} \|\mathbf{H}\mathbf{x}_0\|^2 p(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 &= \text{Tr} \left\{ \mathbf{H} \text{Cov}[\mathbf{x}_0|\mathbf{x}_t] \mathbf{H}^\top + \mathbf{H} \mathbf{D}(\mathbf{x}_t, t) \mathbf{D}(\mathbf{x}_t, t)^\top \mathbf{H}^\top \right\} \\ \int_{\mathbf{x}_0} \|\mathbf{H}\mathbf{x}_0\|^2 p(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 &= \text{Tr} \left\{ \mathbf{H} \text{Cov}[\mathbf{x}_0|\mathbf{x}_t] \mathbf{H}^\top \right\} + \|\mathbf{H}\mathbf{D}(\mathbf{x}_t, t)\|^2 \end{aligned}$$

□

A.4 PROOFS

Proposition 1. *The variational mode-seeking-loss (VML) at diffusion time t , for a non-linear degradation operator \mathcal{A} , measurement \mathbf{y} , and measurement noise variance σ_y^2 is given by*

$$\begin{aligned} \text{VML} = D_{\text{KL}}(p(\mathbf{x}_0|\mathbf{x}_t)||p(\mathbf{x}_0|\mathbf{y})) &= -\log p(\mathbf{x}_t) - \frac{\|\mathbf{D}(\mathbf{x}_t, t) - \mathbf{x}_t\|^2}{2\sigma_t^2} - \frac{1}{2\sigma_t^2} \text{Tr} \{ \text{Cov}[\mathbf{x}_0|\mathbf{x}_t] \} \\ &+ \frac{1}{2\sigma_y^2} \left(-2\mathbf{y}^\top \int_{\mathbf{x}_0} \mathcal{A}(\mathbf{x}_0) p(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 + \int_{\mathbf{x}_0} \|\mathcal{A}(\mathbf{x}_0)\|^2 p(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 \right) + \text{C} \end{aligned}$$

where C is a constant, independent of \mathbf{x}_t . Tr denotes the matrix trace, Cov denotes the covariance matrix, and $\mathbf{D}(\cdot, \cdot)$ denotes the denoiser.

Proof.

$$\begin{aligned} D_{\text{KL}}(p(\mathbf{x}_0|\mathbf{x}_t)||p(\mathbf{x}_0|\mathbf{y})) &= \int_{\mathbf{x}_0} p(\mathbf{x}_0|\mathbf{x}_t) \log \frac{p(\mathbf{x}_0|\mathbf{x}_t)}{p(\mathbf{x}_0|\mathbf{y})} d\mathbf{x}_0 \\ D_{\text{KL}}(p(\mathbf{x}_0|\mathbf{x}_t)||p(\mathbf{x}_0|\mathbf{y})) &= \int_{\mathbf{x}_0} p(\mathbf{x}_0|\mathbf{x}_t) \log \frac{p(\mathbf{x}_t|\mathbf{x}_0) p(\mathbf{x}_0) p(\mathbf{y})}{p(\mathbf{x}_t) p(\mathbf{y}|\mathbf{x}_0) p(\mathbf{x}_0)} d\mathbf{x}_0 \\ D_{\text{KL}}(p(\mathbf{x}_0|\mathbf{x}_t)||p(\mathbf{x}_0|\mathbf{y})) &= \log p(\mathbf{y}) - \log p(\mathbf{x}_t) + \int_{\mathbf{x}_0} p(\mathbf{x}_0|\mathbf{x}_t) \log \frac{p(\mathbf{x}_t|\mathbf{x}_0)}{p(\mathbf{y}|\mathbf{x}_0)} d\mathbf{x}_0 \\ D_{\text{KL}}(p(\mathbf{x}_0|\mathbf{x}_t)||p(\mathbf{x}_0|\mathbf{y})) &= \log p(\mathbf{y}) - \log p(\mathbf{x}_t) + \left(\int_{\mathbf{x}_0} p(\mathbf{x}_0|\mathbf{x}_t) \log p(\mathbf{x}_t|\mathbf{x}_0) d\mathbf{x}_0 \right) \\ &- \left(\int_{\mathbf{x}_0} p(\mathbf{x}_0|\mathbf{x}_t) \log p(\mathbf{y}|\mathbf{x}_0) d\mathbf{x}_0 \right) \end{aligned}$$

{Note that $p(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_0, \sigma_t^2 \mathbf{I})$ and $p(\mathbf{y}|\mathbf{x}_0) = \mathcal{N}(\mathcal{A}(\mathbf{x}_0), \sigma_y^2 \mathbf{I})$. Also, let $\mathbf{x}_0 \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^m$ }

$$\begin{aligned}
D_{\text{KL}}(p(\mathbf{x}_0|\mathbf{x}_t)||p(\mathbf{x}_0|\mathbf{y})) &= -\log p(\mathbf{x}_t) - \frac{1}{2} \left(\int_{\mathbf{x}_0} p(\mathbf{x}_0|\mathbf{x}_t) \frac{\|\mathbf{x}_t - \mathbf{x}_0\|^2}{\sigma_t^2} d\mathbf{x}_0 \right) \\
&\quad + \frac{1}{2} \left(\int_{\mathbf{x}_0} p(\mathbf{x}_0|\mathbf{x}_t) \frac{\|\mathbf{y} - \mathcal{A}(\mathbf{x}_0)\|^2}{\sigma_y^2} d\mathbf{x}_0 \right) + \underbrace{\log p(\mathbf{y}) - \log \frac{\sigma_t^n}{\sigma_y^m} - \frac{n-m}{2} \log 2\pi}_{\text{C}} \\
D_{\text{KL}}(p(\mathbf{x}_0|\mathbf{x}_t)||p(\mathbf{x}_0|\mathbf{y})) &= -\log p(\mathbf{x}_t) - \frac{1}{2\sigma_t^2} \left(\|\mathbf{x}_t\|^2 - 2\mathbf{x}_t^\top \text{D}(\mathbf{x}_t, t) + \int_{\mathbf{x}_0} \|\mathbf{x}_0\|^2 p(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 \right) \\
&\quad + \frac{1}{2\sigma_y^2} \left(-2\mathbf{y}^\top \int_{\mathbf{x}_0} \mathcal{A}(\mathbf{x}_0) p(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 + \int_{\mathbf{x}_0} \|\mathcal{A}(\mathbf{x}_0)\|^2 p(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 \right) + \text{C} \\
&\quad \{\text{By Lemma 1}\} \\
D_{\text{KL}}(p(\mathbf{x}_0|\mathbf{x}_t)||p(\mathbf{x}_0|\mathbf{y})) &= -\log p(\mathbf{x}_t) - \frac{1}{2\sigma_t^2} \left(\|\mathbf{x}_t\|^2 - 2\mathbf{x}_t^\top \text{D}(\mathbf{x}_t, t) + \text{Tr}\{\text{Cov}[\mathbf{x}_0|\mathbf{x}_t]\} \right. \\
&\quad \left. + \|\text{D}(\mathbf{x}_t, t)\|^2 \right) + \frac{1}{2\sigma_y^2} \left(-2\mathbf{y}^\top \int_{\mathbf{x}_0} \mathcal{A}(\mathbf{x}_0) p(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 + \int_{\mathbf{x}_0} \|\mathcal{A}(\mathbf{x}_0)\|^2 p(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 \right) + \text{C} \\
D_{\text{KL}}(p(\mathbf{x}_0|\mathbf{x}_t)||p(\mathbf{x}_0|\mathbf{y})) &= -\log p(\mathbf{x}_t) - \frac{\|\text{D}(\mathbf{x}_t, t) - \mathbf{x}_t\|^2}{2\sigma_t^2} - \frac{1}{2\sigma_t^2} \text{Tr}\{\text{Cov}[\mathbf{x}_0|\mathbf{x}_t]\} \\
&\quad + \frac{1}{2\sigma_y^2} \left(-2\mathbf{y}^\top \int_{\mathbf{x}_0} \mathcal{A}(\mathbf{x}_0) p(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 + \int_{\mathbf{x}_0} \|\mathcal{A}(\mathbf{x}_0)\|^2 p(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 \right) + \text{C}
\end{aligned}$$

□

Proposition 2. *The variational mode-seeking-loss (VML) at diffusion time t , for a linear degradation matrix \mathbf{H} , measurement \mathbf{y} , and measurement noise variance σ_y^2 is given by*

$$\begin{aligned}
\text{VML} = D_{\text{KL}}(p(\mathbf{x}_0|\mathbf{x}_t)||p(\mathbf{x}_0|\mathbf{y})) &= -\log p(\mathbf{x}_t) - \frac{\|\text{D}(\mathbf{x}_t, t) - \mathbf{x}_t\|^2}{2\sigma_t^2} - \frac{1}{2\sigma_t^2} \text{Tr}\{\text{Cov}[\mathbf{x}_0|\mathbf{x}_t]\} \\
&\quad + \underbrace{\frac{\|\mathbf{y} - \mathbf{H}\text{D}(\mathbf{x}_t, t)\|^2}{2\sigma_y^2}}_{\text{measurement consistency}} + \frac{1}{2\sigma_y^2} \text{Tr}\{\mathbf{H}\text{Cov}[\mathbf{x}_0|\mathbf{x}_t]\mathbf{H}^\top\} + \text{C}
\end{aligned}$$

where C is a constant, independent of \mathbf{x}_t . Tr denotes the matrix trace, Cov denotes the covariance matrix, and $\text{D}(\cdot, \cdot)$ denotes the denoiser.

Proof.

Substituting \mathcal{A} with \mathbf{H} in Proposition 1

$$\begin{aligned}
D_{\text{KL}}(p(\mathbf{x}_0|\mathbf{x}_t)||p(\mathbf{x}_0|\mathbf{y})) &= -\log p(\mathbf{x}_t) - \frac{\|\text{D}(\mathbf{x}_t, t) - \mathbf{x}_t\|^2}{2\sigma_t^2} - \frac{1}{2\sigma_t^2} \text{Tr}\{\text{Cov}[\mathbf{x}_0|\mathbf{x}_t]\} \\
&\quad + \frac{1}{2\sigma_y^2} \left(-2\mathbf{y}^\top \int_{\mathbf{x}_0} \mathbf{H}\mathbf{x}_0 p(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 + \int_{\mathbf{x}_0} \|\mathbf{H}\mathbf{x}_0\|^2 p(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 \right) + \text{C} \\
&\quad \{\text{By Lemma 2}\}
\end{aligned}$$

$$\begin{aligned}
D_{\text{KL}}(p(\mathbf{x}_0|\mathbf{x}_t)||p(\mathbf{x}_0|\mathbf{y})) &= -\log p(\mathbf{x}_t) - \frac{\|\text{D}(\mathbf{x}_t, t) - \mathbf{x}_t\|^2}{2\sigma_t^2} - \frac{1}{2\sigma_t^2} \text{Tr}\{\text{Cov}[\mathbf{x}_0|\mathbf{x}_t]\} \\
&\quad + \frac{1}{2\sigma_y^2} \left(-2\mathbf{y}^\top \mathbf{H}\text{D}(\mathbf{x}_t, t) + \text{Tr}\{\mathbf{H}\text{Cov}[\mathbf{x}_0|\mathbf{x}_t]\mathbf{H}^\top\} + \|\mathbf{H}\text{D}(\mathbf{x}_t, t)\|^2 \right) + \text{C}
\end{aligned}$$

$$\begin{aligned}
D_{\text{KL}}(p(\mathbf{x}_0|\mathbf{x}_t)||p(\mathbf{x}_0|\mathbf{y})) &= -\log p(\mathbf{x}_t) - \frac{\|\text{D}(\mathbf{x}_t, t) - \mathbf{x}_t\|^2}{2\sigma_t^2} - \frac{1}{2\sigma_t^2} \text{Tr}\{\text{Cov}[\mathbf{x}_0|\mathbf{x}_t]\} \\
&\quad + \frac{\|\mathbf{y} - \mathbf{H}\text{D}(\mathbf{x}_t, t)\|^2}{2\sigma_y^2} + \frac{1}{2\sigma_y^2} \text{Tr}\{\mathbf{H}\text{Cov}[\mathbf{x}_0|\mathbf{x}_t]\mathbf{H}^\top\} + \text{C}
\end{aligned}$$

{Note that $\text{C} = \log p(\mathbf{y}) - \log \frac{\sigma_t^n}{\sigma_y^m} - \frac{n-m}{2} \log 2\pi$, see the proof of Proposition 1}

□

Simplified VML gradient. The gradient of the simplified VML (i.e., VML_S) for a linear degradation matrix \mathbf{H} is given by

$$\begin{aligned} \text{VML}_S &= -\log p(\mathbf{x}_t) - \frac{\|\mathbf{D}(\mathbf{x}_t, t) - \mathbf{x}_t\|^2}{2\sigma_t^2} + \frac{\|\mathbf{y} - \mathbf{H}\mathbf{D}(\mathbf{x}_t, t)\|^2}{2\sigma_y^2} \\ \nabla_{\mathbf{x}_t} \text{VML}_S &= \underbrace{-\frac{\partial \mathbf{D}^\top(\mathbf{x}_t, t)}{\partial \mathbf{x}_t} \frac{\mathbf{H}^\top (\mathbf{y} - \mathbf{H}\mathbf{D}(\mathbf{x}_t, t))}{\sigma_y^2}}_{\text{measurement consistency gradient}} - \underbrace{\frac{\partial \mathbf{D}^\top(\mathbf{x}_t, t)}{\partial \mathbf{x}_t} \frac{(\mathbf{D}(\mathbf{x}_t, t) - \mathbf{x}_t)}{\sigma_t^2}}_{\text{prior gradient}} \end{aligned}$$

Proof.

$$\begin{aligned} \nabla_{\mathbf{x}_t} \text{VML}_S &= \{-\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)\} - \left\{ \nabla_{\mathbf{x}_t} \frac{\|\mathbf{D}(\mathbf{x}_t, t) - \mathbf{x}_t\|^2}{2\sigma_t^2} \right\} + \left\{ \nabla_{\mathbf{x}_t} \frac{\|\mathbf{y} - \mathbf{H}\mathbf{D}(\mathbf{x}_t, t)\|^2}{2\sigma_y^2} \right\} \\ \nabla_{\mathbf{x}_t} \text{VML}_S &= \left\{ -\frac{\mathbf{D}(\mathbf{x}_t, t) - \mathbf{x}_t}{\sigma_t^2} \right\} - \left\{ \left(\frac{\partial \mathbf{D}^\top(\mathbf{x}_t, t)}{\partial \mathbf{x}_t} \frac{(\mathbf{D}(\mathbf{x}_t, t) - \mathbf{x}_t)}{\sigma_t^2} \right) - \frac{\mathbf{D}(\mathbf{x}_t, t) - \mathbf{x}_t}{\sigma_t^2} \right\} \\ &\quad + \left\{ -\frac{\partial \mathbf{D}^\top(\mathbf{x}_t, t)}{\partial \mathbf{x}_t} \frac{\mathbf{H}^\top (\mathbf{y} - \mathbf{H}\mathbf{D}(\mathbf{x}_t, t))}{\sigma_y^2} \right\} \\ \nabla_{\mathbf{x}_t} \text{VML}_S &= -\frac{\partial \mathbf{D}^\top(\mathbf{x}_t, t)}{\partial \mathbf{x}_t} \frac{\mathbf{H}^\top (\mathbf{y} - \mathbf{H}\mathbf{D}(\mathbf{x}_t, t))}{\sigma_y^2} - \frac{\partial \mathbf{D}^\top(\mathbf{x}_t, t)}{\partial \mathbf{x}_t} \frac{(\mathbf{D}(\mathbf{x}_t, t) - \mathbf{x}_t)}{\sigma_t^2} \end{aligned}$$

□

B EXCLUDING HIGHER-ORDER TERMS IN THE VML

The true denoiser $\mathbf{D}(\mathbf{x}_t, t) = \mathbb{E}[\mathbf{x}_0 | \mathbf{x}_t]$ is related to the true score function $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$ by Tweedie's formula $\mathbf{D}(\mathbf{x}_t, t) = \mathbf{x}_t + \sigma_t^2 \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$ (see Appendix A.1). Applying the derivative to this equation gives $\frac{\partial \mathbf{D}(\mathbf{x}_t, t)}{\partial \mathbf{x}_t} = \mathbf{I} + \sigma_t^2 \nabla_{\mathbf{x}_t}^2 \log p(\mathbf{x}_t) = \frac{1}{\sigma_t^2} \text{Cov}[\mathbf{x}_0 | \mathbf{x}_t]$ (see Appendix A.2). With these reformulations, the higher-order terms of VML can be equivalently expressed in terms of $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$ and $\nabla_{\mathbf{x}_t}^2 \log p(\mathbf{x}_t)$ as follows.

Reformulating the higher-order terms of VML: From Proposition 2, the higher-order terms of the VML (involving $\text{Cov}[\mathbf{x}_0 | \mathbf{x}_t]$), denoted by $(\text{VML}_{\text{High}})$, for a linear degradation matrix \mathbf{H} is

$$\text{VML}_{\text{High}} = -\frac{1}{2\sigma_t^2} \text{Tr} \{ \text{Cov}[\mathbf{x}_0 | \mathbf{x}_t] \} + \frac{1}{2\sigma_y^2} \text{Tr} \{ \mathbf{H} \text{Cov}[\mathbf{x}_0 | \mathbf{x}_t] \mathbf{H}^\top \}$$

where, $\mathbf{x}_t \in \mathbb{R}^n \forall t \geq 0$, $\mathbf{y} \in \mathbb{R}^m$ and $\mathbf{H} \in \mathbb{R}^{m \times n}$. Reformulating VML_{High} in terms of $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$, and $\nabla_{\mathbf{x}_t}^2 \log p(\mathbf{x}_t)$ gives

$$\text{VML}_{\text{High}} = -\frac{\sigma_t^2}{2} \text{Tr} \{ \nabla_{\mathbf{x}_t}^2 \log p(\mathbf{x}_t) \} + \frac{\sigma_t^4}{2\sigma_y^2} \text{Tr} \{ \mathbf{H} \nabla_{\mathbf{x}_t}^2 \log p(\mathbf{x}_t) \mathbf{H}^\top \} + \frac{\sigma_t^2}{2\sigma_y^2} \text{Tr} \{ \mathbf{H} \mathbf{H}^\top \} + C_{\text{VML}_{\text{High}}}$$

where, $C_{\text{VML}_{\text{High}}} = -\frac{n}{2}$.

Proposition 3. Let $p_0(\cdot)$ denote the input data distribution and $p_t(\cdot)$ denote the intermediate marginal distributions of a diffusion process for $t > 0$. Let $\exists \tau > 0, d > 0$ such that $p_t \in C^2$ (twice continuously differentiable) and $\|\mathbf{x}\| \leq d \forall t < \tau$ (i.e., $\forall t < \tau$, \mathbf{x} lies in a compact ball, and $p_t \in C^2$). The function $\text{VML}_{\text{High}_t}(\mathbf{x})$ denoting the higher-order terms (involving $\text{Cov}[\mathbf{x}_0 | \mathbf{x}_t]$) of VML, for a linear degradation operator matrix \mathbf{H} , measurement \mathbf{y} , and measurement noise variance σ_y^2 converges uniformly to $C_{\text{VML}_{\text{High}}}$ in the limit as $t \rightarrow 0$. (Note that $\mathbf{x}_t \in \mathbb{R}^n \forall t \geq 0$, $\mathbf{y} \in \mathbb{R}^m$ and $C_{\text{VML}_{\text{High}}} = -\frac{n}{2}$ as previously mentioned)

$$\begin{aligned} \text{VML}_{\text{High}_t}(\mathbf{x}) &= -\frac{\sigma_t^2}{2} \text{Tr} \{ \nabla_{\mathbf{x}}^2 \log p_t(\mathbf{x}) \} + \frac{\sigma_t^4}{2\sigma_y^2} \text{Tr} \{ \mathbf{H} \nabla_{\mathbf{x}}^2 \log p_t(\mathbf{x}) \mathbf{H}^\top \} \\ &\quad + \frac{\sigma_t^2}{2\sigma_y^2} \text{Tr} \{ \mathbf{H} \mathbf{H}^\top \} + C_{\text{VML}_{\text{High}}} \end{aligned}$$

and, $\text{unif} \lim_{t \rightarrow 0} \text{VML}_{\text{High}_t}(\mathbf{x}) = C_{\text{VML}_{\text{High}}} \forall \mathbf{x} \text{ s.t. } \|\mathbf{x}\| \leq d$

proof sketch. For $t < \tau$, $p_t \in C^2$ implies $\nabla_{\mathbf{x}}^2 \log p_t(\mathbf{x})$ is continuous, which further implies continuity of its component functions, i.e., $\frac{\partial^2 \log p_t(\mathbf{x})}{\partial x^{(i)} \partial x^{(j)}} \forall i, j \in [1, 2, \dots, n]$. Since \mathbf{x} lies in a compact ball (i.e., $\|\mathbf{x}\| \leq d$), it implies that the component functions are bounded $\forall t < \tau$, which further implies boundedness of $\text{Tr} \{ \nabla_{\mathbf{x}}^2 \log p_t(\mathbf{x}) \}$, and $\text{Tr} \{ \mathbf{H} \nabla_{\mathbf{x}}^2 \log p_t(\mathbf{x}) \mathbf{H}^\top \}$. Note that $\text{Tr} \{ \mathbf{H} \mathbf{H}^\top \}$ is also bounded. With $\sigma_t \rightarrow 0$, as $t \rightarrow 0$, it is apparent that $\text{VML}_{\text{High}_t}$ converges uniformly (since the bounds are global and hold for all \mathbf{x} s.t. $\|\mathbf{x}\| \leq d$) to $\text{C}_{\text{VML}_{\text{High}}}$. \square

Reformulating the VML: From Proposition 2, the VML for a linear degradation matrix \mathbf{H} is

$$\begin{aligned} \text{VML} = D_{\text{KL}}(p(\mathbf{x}_0|\mathbf{x}_t)||p(\mathbf{x}_0|\mathbf{y})) &= -\log p(\mathbf{x}_t) - \frac{\|\mathbf{D}(\mathbf{x}_t, t) - \mathbf{x}_t\|^2}{2\sigma_t^2} - \frac{1}{2\sigma_t^2} \text{Tr} \{ \text{Cov}[\mathbf{x}_0|\mathbf{x}_t] \} \\ &+ \underbrace{\frac{\|\mathbf{y} - \text{HD}(\mathbf{x}_t, t)\|^2}{2\sigma_{\mathbf{y}}^2}}_{\text{measurement consistency}} + \frac{1}{2\sigma_{\mathbf{y}}^2} \text{Tr} \{ \mathbf{H} \text{Cov}[\mathbf{x}_0|\mathbf{x}_t] \mathbf{H}^\top \} + \text{C} \end{aligned}$$

where, $\mathbf{x}_t \in \mathbb{R}^n \forall t \geq 0$, $\mathbf{y} \in \mathbb{R}^m$ and $\text{C} = \log p(\mathbf{y}) - \log \frac{\sigma_t^n}{\sigma_{\mathbf{y}}^m} - \frac{n-m}{2} \log 2\pi$ (see the proof of Proposition 2). Reformulating VML in terms of $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$, and $\nabla_{\mathbf{x}_t}^2 \log p(\mathbf{x}_t)$ gives

$$\begin{aligned} \text{VML} = D_{\text{KL}}(p(\mathbf{x}_0|\mathbf{x}_t)||p(\mathbf{x}_0|\mathbf{y})) &= -\log p(\mathbf{x}_t) - \frac{\sigma_t^2}{2} \|\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)\|^2 - \frac{\sigma_t^2}{2} \text{Tr} \{ \nabla_{\mathbf{x}_t}^2 \log p(\mathbf{x}_t) \} \\ &+ \underbrace{\frac{\|\mathbf{y} - \text{HD}(\mathbf{x}_t, t)\|^2}{2\sigma_{\mathbf{y}}^2}}_{\text{measurement consistency}} + \frac{\sigma_t^4}{2\sigma_{\mathbf{y}}^2} \text{Tr} \{ \mathbf{H} \nabla_{\mathbf{x}_t}^2 \log p(\mathbf{x}_t) \mathbf{H}^\top \} + \frac{\sigma_t^2}{2\sigma_{\mathbf{y}}^2} \text{Tr} \{ \mathbf{H} \mathbf{H}^\top \} + \log p(\mathbf{y}) - n \log \sigma_t + \text{C}_{\text{VML}} \end{aligned}$$

where, $\text{C}_{\text{VML}} = -\frac{n}{2} + m \log \sigma_{\mathbf{y}} - \frac{n-m}{2} \log 2\pi$.

Proposition 4. Let $p_0(\cdot)$ denote the input data distribution and $p_t(\cdot)$ denote the intermediate marginal distributions of a diffusion process for $t > 0$. Let $\exists \tau > 0, d > 0$, such that $p_t \in C^2$ (twice continuously differentiable) and $\|\mathbf{x}\| \leq d \forall t < \tau$ (In other words, $\exists \tau > 0, d > 0$, such that $p_t \in C^2$, and $\|\mathbf{x}_t\| \leq d \forall t < \tau$ where $\mathbf{x}_t \in \mathcal{M}_t$ i.e., the intermediate diffusion manifold at time t). Assuming sufficient conditions for $\lim_{t \rightarrow 0} \log p_t(\mathbf{x}) = \log p_0(\mathbf{x})$, the function $\text{VML}_t(\mathbf{x}) + n \log \sigma_t$, for a linear degradation operator matrix \mathbf{H} , measurement \mathbf{y} , and measurement noise variance $\sigma_{\mathbf{y}}^2$ converges pointwise to $-\log p_0(\mathbf{x}|\mathbf{y}) + \hat{\text{C}}_{\text{VML}}$ in the limit as $t \rightarrow 0$. (Note that $\mathbf{x}_t \in \mathbb{R}^n \forall t \geq 0$, $\mathbf{y} \in \mathbb{R}^m$ and $\hat{\text{C}}_{\text{VML}} = -\frac{n}{2} - \frac{n}{2} \log 2\pi$)

$$\begin{aligned} \text{VML}_t(\mathbf{x}) &= -\log p_t(\mathbf{x}) - \frac{\sigma_t^2}{2} \|\nabla_{\mathbf{x}} \log p_t(\mathbf{x})\|^2 - \frac{\sigma_t^2}{2} \text{Tr} \{ \nabla_{\mathbf{x}}^2 \log p_t(\mathbf{x}) \} + \frac{\|\mathbf{y} - \text{HD}(\mathbf{x}, t)\|^2}{2\sigma_{\mathbf{y}}^2} \\ &+ \frac{\sigma_t^4}{2\sigma_{\mathbf{y}}^2} \text{Tr} \{ \mathbf{H} \nabla_{\mathbf{x}}^2 \log p_t(\mathbf{x}) \mathbf{H}^\top \} + \frac{\sigma_t^2}{2\sigma_{\mathbf{y}}^2} \text{Tr} \{ \mathbf{H} \mathbf{H}^\top \} + \log p(\mathbf{y}) - n \log \sigma_t + \text{C}_{\text{VML}} \end{aligned}$$

$$\text{and, } \lim_{t \rightarrow 0} \text{VML}_t(\mathbf{x}) + n \log \sigma_t = -\log p_0(\mathbf{x}|\mathbf{y}) + \hat{\text{C}}_{\text{VML}} \forall \mathbf{x} \text{ s.t. } \|\mathbf{x}\| \leq d$$

proof sketch. It suffices to show that $\lim_{t \rightarrow 0} \{ \text{VML}_t(\mathbf{x}) + n \log \sigma_t + \log p_0(\mathbf{x}|\mathbf{y}) - \hat{\text{C}}_{\text{VML}} \} = 0 \forall \mathbf{x}$ s.t. $\|\mathbf{x}\| \leq d$.

$$\begin{aligned} &\text{VML}_t(\mathbf{x}) + n \log \sigma_t + \log p_0(\mathbf{x}|\mathbf{y}) - \hat{\text{C}}_{\text{VML}} \\ &= \text{VML}_t(\mathbf{x}) + n \log \sigma_t + \log p_0(\mathbf{y}|\mathbf{x}) + \log p_0(\mathbf{x}) - \log p(\mathbf{y}) - \text{C}_{\text{VML}} + \left\{ m \log \sigma_{\mathbf{y}} + \frac{m}{2} \log 2\pi \right\} \\ &= \underbrace{\left\{ -\frac{\sigma_t^2}{2} \|\nabla_{\mathbf{x}} \log p_t(\mathbf{x})\|^2 - \frac{\sigma_t^2}{2} \text{Tr} \{ \nabla_{\mathbf{x}}^2 \log p_t(\mathbf{x}) \} + \frac{\sigma_t^4}{2\sigma_{\mathbf{y}}^2} \text{Tr} \{ \mathbf{H} \nabla_{\mathbf{x}}^2 \log p_t(\mathbf{x}) \mathbf{H}^\top \} + \frac{\sigma_t^2}{2\sigma_{\mathbf{y}}^2} \text{Tr} \{ \mathbf{H} \mathbf{H}^\top \} \right\}}_{T_A} \\ &+ \underbrace{\left\{ -\log p_t(\mathbf{x}) + \log p_0(\mathbf{x}) \right\}}_{T_B} + \underbrace{\left\{ m \log \sigma_{\mathbf{y}} + \frac{m}{2} \log 2\pi + \frac{\|\mathbf{y} - \text{HD}(\mathbf{x}, t)\|^2}{2\sigma_{\mathbf{y}}^2} + \log p_0(\mathbf{y}|\mathbf{x}) \right\}}_{T_C} \end{aligned}$$

To show that $\lim_{t \rightarrow 0} \{ \text{VML}_t(\mathbf{x}) + n \log \sigma_t + \log p_0(\mathbf{x}|\mathbf{y}) - \hat{\text{C}}_{\text{VML}} \} = 0 \forall \mathbf{x}$ s.t. $\|\mathbf{x}\| \leq d$, we need to show that $\lim_{t \rightarrow 0} T_A = \lim_{t \rightarrow 0} T_B = \lim_{t \rightarrow 0} T_C = 0$. Under sufficient conditions assumed for

864 $\lim_{t \rightarrow 0} \log p_t = \log p_0$, it implies that $\lim_{t \rightarrow 0} T_B = 0$. Considering T_A : as $p_t(\mathbf{x}) \in C^2$ and \mathbf{x}
 865 lies in a compact set, it implies that $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ and $\nabla_{\mathbf{x}}^2 \log p_t(\mathbf{x})$ are bounded for all $t < \tau$. With
 866 $\sigma_t \rightarrow 0$ as $t \rightarrow 0$, $\lim_{t \rightarrow 0} T_A = 0$. Considering T_C : Note that $p_0(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{H}\mathbf{x}, \sigma_{\mathbf{y}}^2 \mathbf{I})$ (see Equation 1)
 867 and $\log p_0(\mathbf{y}|\mathbf{x}) = -m \log \sigma_{\mathbf{y}} - \frac{m}{2} \log 2\pi - \frac{\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2}{2\sigma_{\mathbf{y}}^2}$. It can be seen that, as $t \rightarrow 0$, $D(\mathbf{x}, t) \rightarrow \mathbf{x}$, since
 868 $D(\mathbf{x}, t) = \mathbf{x} + \sigma_t^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ (Appendix A.1). This further implies $\lim_{t \rightarrow 0} T_C = 0$. \square
 869

870 **Reformulating the Simplified-VML:** From Equation (5), the Simplified-VML (VML_S) for a linear
 871 degradation matrix \mathbf{H} is
 872

$$873 \text{ VML}_S = -\log p(\mathbf{x}_t) - \frac{\|D(\mathbf{x}_t, t) - \mathbf{x}_t\|^2}{2\sigma_t^2} + \underbrace{\frac{\|\mathbf{y} - \mathbf{H}D(\mathbf{x}_t, t)\|^2}{2\sigma_{\mathbf{y}}^2}}_{\text{measurement consistency}} + C$$

877 where, $\mathbf{x}_t \in \mathbb{R}^n \forall t \geq 0$, $\mathbf{y} \in \mathbb{R}^m$ and $C = \log p(\mathbf{y}) - \log \frac{\sigma_t^n}{\sigma_{\mathbf{y}}^m} - \frac{n-m}{2} \log 2\pi$. Reformulating
 878 VML_S in terms of $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$, and $\nabla_{\mathbf{x}_t}^2 \log p(\mathbf{x}_t)$ gives
 879

$$880 \text{ VML}_S = -\log p(\mathbf{x}_t) - \frac{\sigma_t^2}{2} \|\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)\|^2 + \underbrace{\frac{\|\mathbf{y} - \mathbf{H}D(\mathbf{x}_t, t)\|^2}{2\sigma_{\mathbf{y}}^2}}_{\text{measurement consistency}} + \log p(\mathbf{y}) - n \log \sigma_t + C_{\text{VML}_S}$$

884 where, $C_{\text{VML}_S} = m \log \sigma_{\mathbf{y}} - \frac{n-m}{2} \log 2\pi$.

885 **Proposition 5.** Let $p_0(\cdot)$ denote the input data distribution and $p_t(\cdot)$ denote the intermediate
 886 marginal distributions of a diffusion process for $t > 0$. Let $\exists \tau > 0, d > 0$ such that $p_t \in C^1$
 887 (once continuously differentiable) and $\|\mathbf{x}\| \leq d \forall t < \tau$ (i.e., $\forall t < \tau$, \mathbf{x} lies in a compact ball, and
 888 $p_t \in C^1$). Assuming sufficient conditions for $\lim_{t \rightarrow 0} \log p_t(\mathbf{x}) = \log p_0(\mathbf{x}) \forall \mathbf{x}$, the function given
 889 by $\text{VML}_{S_t}(\mathbf{x}) + n \log \sigma_t$, (where $\text{VML}_{S_t}(\mathbf{x})$ denotes the Simplified-VML) for a linear degrada-
 890 tion operator matrix \mathbf{H} , measurement \mathbf{y} , and measurement noise variance $\sigma_{\mathbf{y}}^2$ converges pointwise
 891 to $-\log p_0(\mathbf{x}|\mathbf{y}) + \hat{C}_{\text{VML}_S}$ in the limit as $t \rightarrow 0$. (Note that $\mathbf{x}_t \in \mathbb{R}^n \forall t \geq 0$, $\mathbf{y} \in \mathbb{R}^m$ and
 892 $\hat{C}_{\text{VML}_S} = -\frac{n}{2} \log 2\pi$)
 893

$$894 \text{ VML}_{S_t}(\mathbf{x}) = -\log p_t(\mathbf{x}) - \frac{\sigma_t^2}{2} \|\nabla_{\mathbf{x}} \log p_t(\mathbf{x})\|^2 + \frac{\|\mathbf{y} - \mathbf{H}D(\mathbf{x}, t)\|^2}{2\sigma_{\mathbf{y}}^2} + \log p(\mathbf{y}) - n \log \sigma_t + C_{\text{VML}_S}$$

$$897 \text{ and, } \lim_{t \rightarrow 0} \text{VML}_{S_t}(\mathbf{x}) + n \log \sigma_t = -\log p_0(\mathbf{x}|\mathbf{y}) + \hat{C}_{\text{VML}_S} \forall \mathbf{x} \text{ s.t. } \|\mathbf{x}\| \leq d$$

899 *proof sketch.* By arguments similar to those in the proof of Proposition 4 \square
 900

901 **Remark 1.** Note that the limit of $\text{VML}_t(\mathbf{x})$ as $t \rightarrow 0$ doesn't exist. However, for a given arbitrary
 902 t , a global minimizer of $\text{VML}_t(\mathbf{x})$ is also a global minimizer of $\text{VML}_t(\mathbf{x}) + n \log \sigma_t$ (for $n \log \sigma_t$ is
 903 a constant given t) and vice-versa. From Proposition 4, $\text{VML}_t(\mathbf{x}) + n \log \sigma_t$ converges pointwise
 904 to $-\log p_0(\mathbf{x}|\mathbf{y}) + \hat{C}_{\text{VML}}$ in the limit as $t \rightarrow 0$.
 905

906 **Remark 2.** Note that the limit of $\text{VML}_{S_t}(\mathbf{x})$ as $t \rightarrow 0$ doesn't exist. However, for a given arbitrary
 907 t , a global minimizer of $\text{VML}_{S_t}(\mathbf{x})$ is also a global minimizer of $\text{VML}_{S_t}(\mathbf{x}) + n \log \sigma_t + C_{\text{VML}_{\text{High}}}$
 908 (for $n \log \sigma_t + C_{\text{VML}_{\text{High}}}$ is a constant given t) and vice-versa. From Proposition 5, $\text{VML}_{S_t}(\mathbf{x}) +$
 909 $n \log \sigma_t + C_{\text{VML}_{\text{High}}}$ converges pointwise to $-\log p_0(\mathbf{x}|\mathbf{y}) + \hat{C}_{\text{VML}_S} + C_{\text{VML}_{\text{High}}} = -\log p_0(\mathbf{x}|\mathbf{y}) +$
 910 \hat{C}_{VML} in the limit as $t \rightarrow 0$.
 911

912 **Remark 3.** From Proposition 3, the function $(\text{VML}_{\text{High}_t} - C_{\text{VML}_{\text{High}}})$ converges uniformly to the
 913 zero function in the limit as $t \rightarrow 0$. Note that $\text{VML}_{\text{High}_t} - C_{\text{VML}_{\text{High}}} = (\text{VML}_t + n \log \sigma_t) -$
 914 $(\text{VML}_{S_t} + n \log \sigma_t + C_{\text{VML}_{\text{High}}})$, i.e., the difference of essentially equivalent (in terms of global
 915 minimizers) functions of VML and Simplified-VML respectively (see Remarks 1 and 2). It implies that
 916 the difference of these functions becomes arbitrarily small as $t \rightarrow 0$. In practice, this approximation
 917 of VML with Simplified-VML may not be critical, as the errors arising due to the imperfect optimizer
 and numerical errors from discretizing the reverse SDE or PF ODE typically dominate early in the
 reverse diffusion process.

C EXPERIMENTAL SETUP, IMPLEMENTATION DETAILS, QUALITATIVE VISUALIZATIONS, AND MORE

C.1 EXPERIMENTS IN TABLE 1 AND TABLE 2

In these experiments, we considered image restoration inverse problems with severe enough degradations to make it more challenging. However, we do not resort to extreme degradations, as the corresponding measurements typically do not provide strong guidance for recovering the ground truth image, since extreme degradations make the posterior highly multimodal to an extent that the restored image is perceptually dissimilar to the ground truth image, which makes it challenging to assess the performance using the usual LIPS/FID metrics. Our experiments included half-mask inpainting, 4× super-resolution, and uniform deblurring with a 16×16 kernel. We utilize the SVD-based super-resolution and uniform deblurring operators from Kawar et al. (2022) to ensure that the preconditioner can be computed efficiently. For uniform deblurring, we observed that the degradations are not severe enough, as the pseudoinverse solution already gives an almost perfect reconstruction. To make it more challenging, we zero out the singular values below a high enough threshold (0.2) as opposed to the default threshold (0.03) used in Kawar et al. (2022).

Table 4: Best learning rate configuration for experiments in Table 1 and Table 2. Note that the learning rate is $\gamma_0 \cdot \sigma_y^2$, with γ_0 as reported in the table.

| Dataset | Method | Inpainting | 4× Super-res | Deblurring |
|-------------|------------------------|-------------------|-------------------|-------------------|
| ImageNet64 | VML-MAP | $\gamma_0 = 1.5$ | $\gamma_0 = 30.0$ | $\gamma_0 = 2.25$ |
| | VML-MAP _{pre} | $\gamma_0 = 1.5$ | $\gamma_0 = 1.75$ | $\gamma_0 = 2.0$ |
| ImageNet256 | VML-MAP | $\gamma_0 = 1.25$ | $\gamma_0 = 25.0$ | $\gamma_0 = 2.0$ |
| | VML-MAP _{pre} | $\gamma_0 = 1.25$ | $\gamma_0 = 1.5$ | $\gamma_0 = 1.5$ |
| FFHQ256 | VML-MAP | $\gamma_0 = 1.25$ | $\gamma_0 = 30.0$ | $\gamma_0 = 2.0$ |
| | VML-MAP _{pre} | $\gamma_0 = 1.25$ | $\gamma_0 = 1.5$ | $\gamma_0 = 1.5$ |

For DDRM, IIGDM, MAPGA, VML-MAP, and VML-MAP_{pre}, we use the EDM noise schedule from Karras et al. (2022), with $\sigma_{min} = 0.002$, $\sigma_{max} = 140$, and $\rho = 7$. Note that MAPGA requires a consistency model by default, so throughout this paper, we use the variant MAPGA(D) from Gutha et al. (2025), which replaces the consistency model with a single-step denoiser approximation. We use the EDM schedule for DDRM and IIGDM as it performs the best compared to the default schedules used in their original repositories. For DAPS, we use the default DAPS-1K and DAPS-4K configurations mentioned in the original paper, and observed that the default hyperparameters used in the paper for box inpainting, super-resolution, and Gaussian deblurring also perform the best for our half-mask inpainting, super-resolution, and uniform deblurring tasks.

For each experiment with DDRM and IIGDM, we search for N (i.e., the number of reverse diffusion steps) over $\{20.50, 100, 200, 500, 1000\}$ and report the best result. For each experiment using MAPGA, VML-MAP, and VML-MAP_{pre}, we search for (N, K) (N denotes the total number of diffusion time steps, and K denotes the number of gradient ascent/descent iterations per step) over $\{(20, 50), (50, 20), (100, 10), (200, 5), (500, 2), (1000, 1)\}$ and report the best performance (this keeps the total budget for MAP-GA, VML-MAP and VML-MAP_{pre} within 1000 optimization steps in total). In every case, we find the best configuration to be $(N, K) = (20, 50)$. We set $\sigma_y = 0$ for DDRM, IIGDM, and MAPGA, while for DAPS, VML-MAP and VML-MAP_{pre}, we set $\sigma_y = 1e-9$. For MAPGA, the default learning rate from the original repository was used, while for VML-MAP and VML-MAP_{pre}, we report the best learning rate configuration for each task as $\gamma_0 \cdot \sigma_y^2$, with γ_0 shown in Table 4. Our implementation of DDRM, IIGDM, and MAPGA is based on the following original repositories ddrm, pggm, mapga, respectively.

C.2 EXPERIMENTS IN TABLE 3

In these experiments, we focused on the image inpainting task with different types of masks. As mentioned in Sections 4.2 and 5, DDRM, Π GDM, MAPGA, and VML-MAP_{pre} require the SVD of H , while DAPS and VML-MAP only require the forward operation of H . For inpainting, H is a diagonal matrix with zeros for indices corresponding to masked pixels, and ones for indices corresponding to observed pixels. The SVD of H in this case is trivial, since H itself is the singular value matrix, with the left and the right singular matrices being identity. This ensures a fair comparison among all the methods, irrespective of whether a method requires the SVD of H or not.

Different types of inpainting masks that were considered in these experiments include

- **Expand mask:** Pixels outside the 128×128 square center-crop are masked
- **Box mask:** Pixels within the 128×128 square center-crop are masked
- **Super-resolution mask:** Alternative pixels are masked
- **Random mask:** 70% of the pixels are randomly masked

We use $\gamma_0 = 1.0$ (i.e., a learning rate of $1.0 \cdot \sigma_y^2$) for all four tasks on both ImageNet256 and FFHQ256, and follow the same settings in Appendix C.1 for other hyperparameter configurations.

C.3 QUALITATIVE VISUALIZATIONS

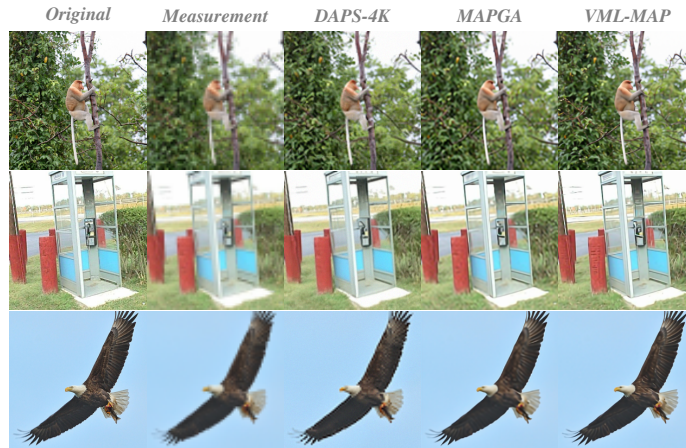


Figure 8: $4\times$ Super-resolution task from Table 2. Zoom in for the best view.

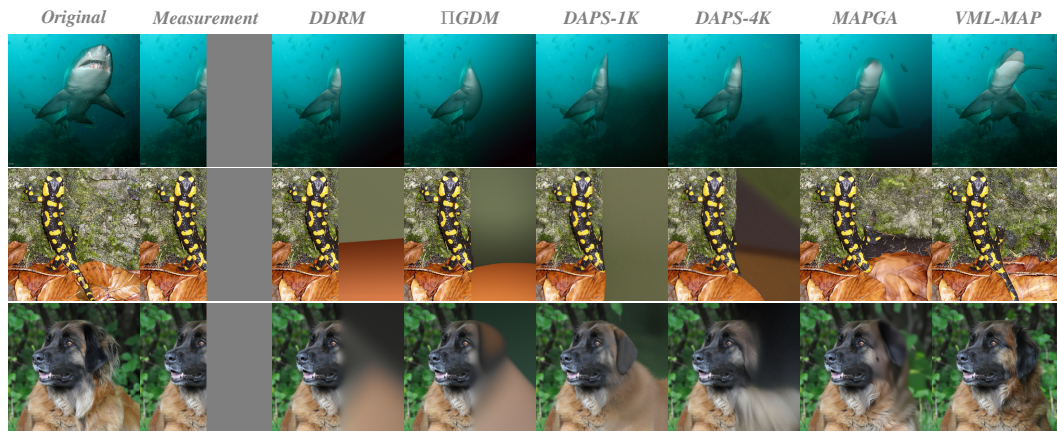


Figure 9: Half-mask inpainting task from Table 2. Zoom in for the best view.

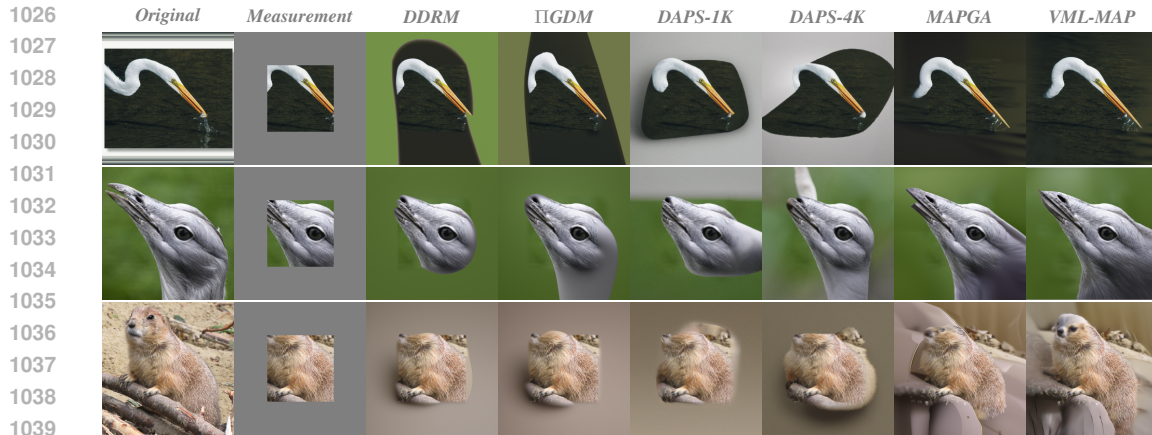


Figure 10: Expand-mask inpainting task from Table 3. Zoom in for the best view.

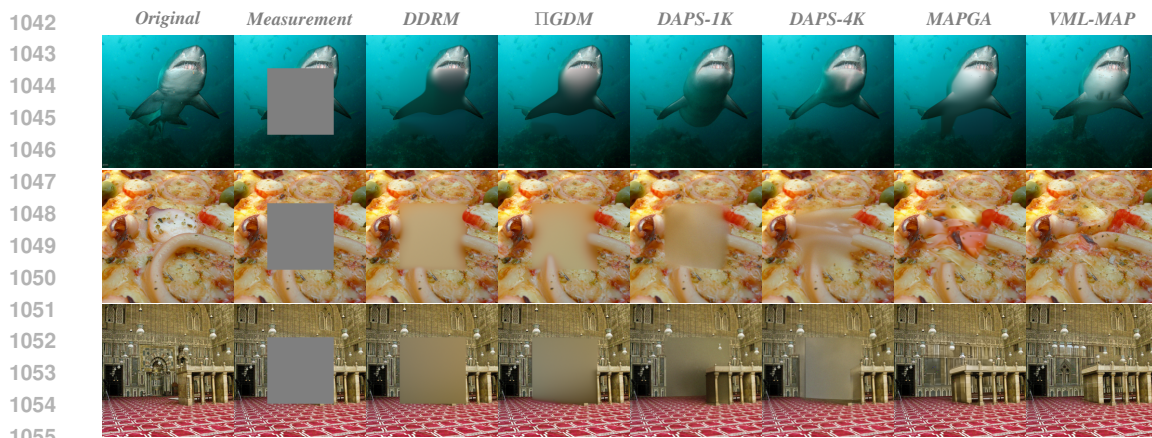


Figure 11: Box-mask inpainting task from Table 3. Zoom in for the best view.



Figure 12: Super-res-mask inpainting task from Table 3. Zoom in for the best view.

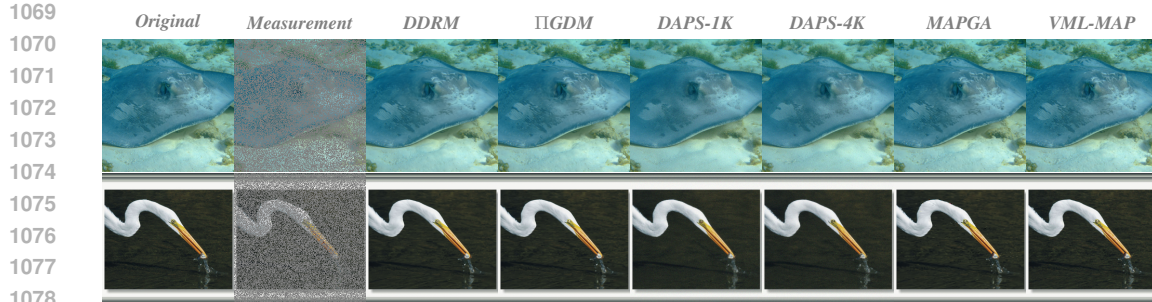


Figure 13: Random-mask inpainting task from Table 3. Zoom in for the best view.

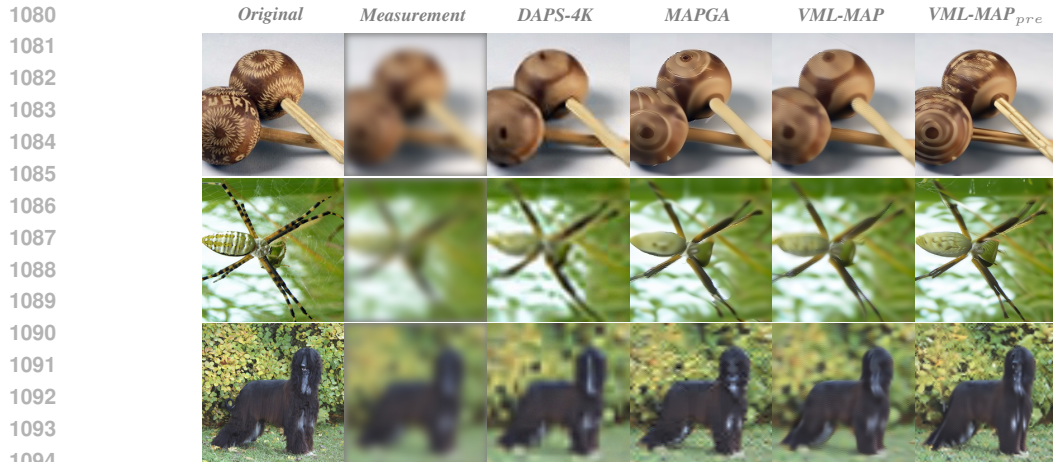


Figure 14: Deblurring task from Table 2. Zoom in for the best view.

1095 C.4 EXPERIMENTS ACROSS DIFFERENT SEEDS

1096

1097

1098

1099

1100 Table 5: Experiments from Table 1 (i.e., Half-mask inpainting, $4\times$ Super-resolution, and Uniform

1101 deblurring on 1000 images of ImageNet64 validation set) are repeated across 4 different seeds. The

1102 mean and standard deviation across these runs are reported.

1103

1104

1105

1106

| Method | Inpainting | | $4\times$ Super-res | | Deblurring | |
|------------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| | LPIPS \downarrow | FID \downarrow | LPIPS \downarrow | FID \downarrow | LPIPS \downarrow | FID \downarrow |
| DDRM | 0.263 \pm 0.000 | 57.15 \pm 0.241 | 0.234 \pm 0.000 | 77.82 \pm 0.526 | 0.467 \pm 0.001 | 197.7 \pm 0.600 |
| PIGDM | 0.242 \pm 0.000 | 54.69 \pm 0.334 | 0.241 \pm 0.000 | 88.63 \pm 0.472 | 0.439 \pm 0.001 | 164.7 \pm 1.098 |
| MAPGA | 0.172 \pm 0.000 | 46.43 \pm 0.150 | 0.204 \pm 0.000 | 84.72 \pm 0.779 | <u>0.322</u> \pm 0.001 | 113.9 \pm 0.753 |
| VML-MAP | 0.146 \pm 0.000 | 38.84 \pm 0.233 | <u>0.136</u> \pm 0.001 | <u>62.19</u> \pm 0.298 | 0.356 \pm 0.001 | <u>106.6</u> \pm 0.682 |
| VML-MAP_{pre} | 0.146 \pm 0.000 | 38.84 \pm 0.233 | 0.129 \pm 0.000 | 60.01 \pm 0.400 | 0.266 \pm 0.001 | 77.46 \pm 0.424 |

1107

1108

1109

1110

1111

1112

1113

1114 Table 6: Experiments from Table 2 (i.e., Half-mask inpainting, $4\times$ Super-resolution, and Uniform

1115 deblurring on 100 validation images of ImageNet256 and 100 images of FFHQ256) are repeated

1116 across 4 different seeds. The mean and standard deviation across these runs are reported.

1117

1118

1119

1120

1121

1122

1123

1124

1125

1126

1127

1128

1129

1130

1131

1132

1133

| Dataset | Method | Inpainting | | $4\times$ Super-res | | Deblurring | |
|------------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| | | LPIPS \downarrow | FID \downarrow | LPIPS \downarrow | FID \downarrow | LPIPS \downarrow | FID \downarrow |
| ImageNet | DDRM | 0.393 \pm 0.001 | 103.5 \pm 0.657 | 0.289 \pm 0.000 | 89.40 \pm 0.962 | 0.618 \pm 0.000 | 233.4 \pm 2.033 |
| | PIGDM | 0.373 \pm 0.000 | 103.7 \pm 0.694 | 0.292 \pm 0.001 | 83.88 \pm 0.226 | 0.562 \pm 0.000 | 231.8 \pm 0.631 |
| | MAPGA | <u>0.290</u> \pm 0.001 | <u>80.76</u> \pm 2.013 | 0.273 \pm 0.001 | 76.12 \pm 0.587 | <u>0.459</u> \pm 0.002 | <u>194.9</u> \pm 1.548 |
| | DAPS-1K | 0.384 \pm 0.001 | 98.45 \pm 1.879 | 0.254 \pm 0.001 | 71.60 \pm 1.010 | 0.605 \pm 0.001 | 217.8 \pm 3.042 |
| | DAPS-4K | 0.365 \pm 0.004 | 93.62 \pm 0.856 | 0.244 \pm 0.000 | 69.71 \pm 1.554 | 0.593 \pm 0.001 | 227.9 \pm 2.991 |
| | VML-MAP | 0.262 \pm 0.002 | 74.21 \pm 3.209 | 0.194 \pm 0.001 | <u>60.08</u> \pm 0.687 | 0.509 \pm 0.003 | 200.4 \pm 2.812 |
| VML-MAP_{pre} | 0.262 \pm 0.002 | 74.21 \pm 3.209 | <u>0.196</u> \pm 0.003 | 58.60 \pm 2.076 | 0.367 \pm 0.002 | 165.2 \pm 2.037 | |
| FFHQ | DDRM | 0.246 \pm 0.001 | 71.23 \pm 0.521 | 0.154 \pm 0.000 | 70.07 \pm 0.495 | 0.307 \pm 0.000 | 117.9 \pm 0.427 |
| | PIGDM | 0.237 \pm 0.001 | 70.40 \pm 0.494 | 0.147 \pm 0.000 | 68.38 \pm 0.605 | 0.293 \pm 0.000 | 114.0 \pm 0.302 |
| | MAPGA | <u>0.206</u> \pm 0.000 | 64.15 \pm 0.660 | 0.132 \pm 0.000 | 63.82 \pm 0.455 | 0.235 \pm 0.000 | 119.2 \pm 1.045 |
| | DAPS-1K | 0.233 \pm 0.001 | 61.22 \pm 0.759 | 0.113 \pm 0.000 | 60.89 \pm 1.028 | 0.260 \pm 0.000 | 100.8 \pm 0.951 |
| | DAPS-4K | 0.224 \pm 0.000 | <u>60.22</u> \pm 0.738 | <u>0.100</u> \pm 0.000 | 58.29 \pm 0.398 | <u>0.230</u> \pm 0.001 | <u>93.71</u> \pm 1.217 |
| | VML-MAP | 0.180 \pm 0.001 | 52.76 \pm 0.526 | 0.100 \pm 0.000 | <u>57.55</u> \pm 1.414 | 0.247 \pm 0.000 | 99.40 \pm 1.799 |
| VML-MAP_{pre} | 0.180 \pm 0.001 | 52.76 \pm 0.526 | 0.100 \pm 0.000 | 52.20 \pm 0.814 | 0.182 \pm 0.000 | 93.48 \pm 0.681 | |

Here, we repeat the experiments from Table 1 and Table 2 across 4 different seeds and report the mean and standard deviation of LPIPS and FID. The variation of these quantitative metrics across different seeds is quite insignificant (see Table 5 and Table 6), which also reinforces the validity of the original conclusions drawn from Table 1 and Table 2 in Section 4 and Section 5, respectively.

While the quantitative metrics are not sensitive to different seeds, the qualitative results do show variations across seeds. Note that VML-MAP involves optimization of VML (using gradient descent) during each reverse diffusion step (see Figure 2, and Algorithm 1). Since gradient descent is not a perfect optimizer in practice, the results of the optimization can widely differ based on the initialization \mathbf{x}_T and also the stochastic components within VML-MAP (such as the sampling operation for renoising after each optimization block). Since the random seed directly influences this, the qualitative results differ across different seeds. While it's probably not equivalent to posterior sampling, this behaviour, however, can be leveraged to produce diverse samples (see Figure 15).

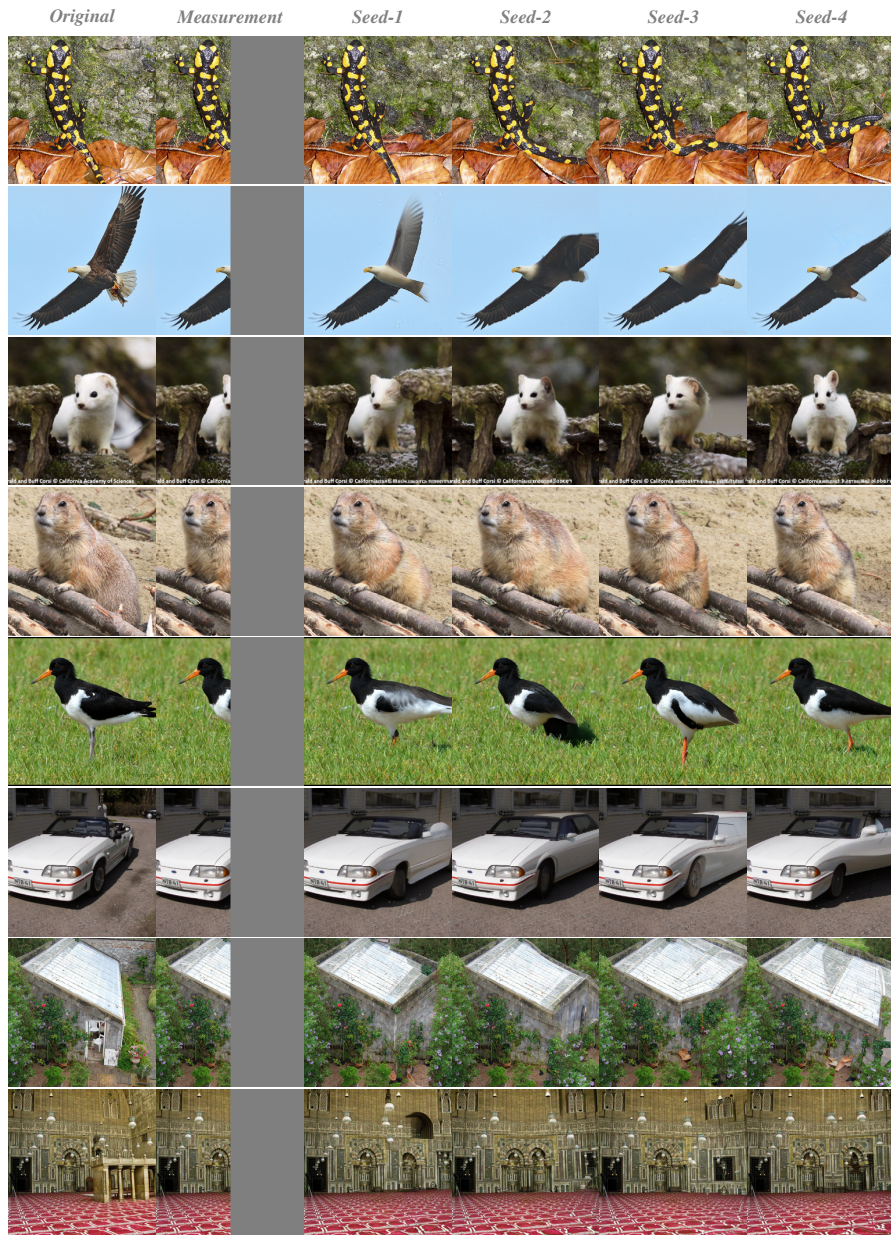


Figure 15: Restored images of VML-MAP across different seeds for the half-mask inpainting task. Left to right: original, measurement, and restored images of VML-MAP with 4 different seeds.

D EXTENSION TO LATENT DIFFUSION MODELS (LDM)

D.1 APPROXIMATING THE VML FOR LDMs

Here, we provide an extension of the VML objective to Latent Diffusion Models (LDM) for solving inverse problems. In LDMs, we treat both the encoder \mathcal{E} and the decoder \mathcal{D} as deterministic mappings from a clean image \mathbf{x}_0 to a clean latent variable \mathbf{z}_0 and vice-versa, respectively. To solve an inverse problem with a pre-trained LDM and a given measurement \mathbf{y} , we first aim to solve for the MAP estimate $\mathbf{z}_0^* = \arg \max_{\mathbf{z}_0} \log p(\mathbf{z}_0|\mathbf{y})$ with the VML objective extended to LDMs and later use the decoder to predict a clean image in a deterministic manner i.e., $\mathbf{x}_0^* = \mathcal{D}(\mathbf{z}_0^*)$. First, we define the VML objective in LDMs, which we term VML_{LDM} , as the KL divergence between $p(\mathbf{z}_0|\mathbf{z}_t)$ and $p(\mathbf{z}_0|\mathbf{y})$, and further approximate it as given below.

Proposition 6. *The variational mode-seeking-loss for an LDM (VML^{LDM}) at diffusion time t , for a degradation operator \mathcal{A} , measurement \mathbf{y} , and measurement noise variance σ_y^2 is given by*

$$\begin{aligned} \text{VML}^{\text{LDM}} = D_{\text{KL}}(p(\mathbf{z}_0|\mathbf{z}_t)||p(\mathbf{z}_0|\mathbf{y})) &\approx -\log p(\mathbf{z}_t) - \frac{\|\mathbf{D}(\mathbf{z}_t, t) - \mathbf{z}_t\|^2}{2\sigma_t^2} - \frac{1}{2\sigma_t^2} \text{Tr} \{ \text{Cov}[\mathbf{z}_0|\mathbf{z}_t] \} \\ &+ \frac{\|\mathbf{y} - \mathcal{A}(\mathcal{D}(\mathbf{D}(\mathbf{z}_t, t)))\|^2}{2\sigma_y^2} + \frac{1}{2\sigma_y^2} \left(\text{Tr} \left\{ \frac{\partial \mathcal{A}(\mathcal{D}(\mathbf{D}(\mathbf{z}_t, t)))}{\partial \mathbf{D}(\mathbf{z}_t, t)} \text{Cov}[\mathbf{z}_0|\mathbf{z}_t] \frac{\partial \mathcal{A}(\mathcal{D}(\mathbf{D}(\mathbf{z}_t, t)))^\top}{\partial \mathbf{D}(\mathbf{z}_t, t)} \right\} \right) + \text{C} \end{aligned}$$

where C is a constant, independent of \mathbf{x}_t . Tr denotes the matrix trace, Cov denotes the covariance matrix, $\mathbf{D}(\cdot, \cdot)$ denotes the denoiser, and $\mathcal{D}(\cdot)$ denotes the LDM decoder.

Proof.

$$\begin{aligned} D_{\text{KL}}(p(\mathbf{z}_0|\mathbf{z}_t)||p(\mathbf{z}_0|\mathbf{y})) &= \int_{\mathbf{z}_0} p(\mathbf{z}_0|\mathbf{z}_t) \log \frac{p(\mathbf{z}_0|\mathbf{z}_t)}{p(\mathbf{z}_0|\mathbf{y})} d\mathbf{z}_0 \\ D_{\text{KL}}(p(\mathbf{z}_0|\mathbf{z}_t)||p(\mathbf{z}_0|\mathbf{y})) &= \int_{\mathbf{z}_0} p(\mathbf{z}_0|\mathbf{z}_t) \log \frac{p(\mathbf{z}_t|\mathbf{z}_0)p(\mathbf{z}_0)p(\mathbf{y})}{p(\mathbf{z}_t)p(\mathbf{y}|\mathbf{z}_0)p(\mathbf{z}_0)} d\mathbf{z}_0 \\ D_{\text{KL}}(p(\mathbf{z}_0|\mathbf{z}_t)||p(\mathbf{z}_0|\mathbf{y})) &= \log p(\mathbf{y}) - \log p(\mathbf{z}_t) + \int_{\mathbf{z}_0} p(\mathbf{z}_0|\mathbf{z}_t) \log \frac{p(\mathbf{z}_t|\mathbf{z}_0)}{p(\mathbf{y}|\mathbf{z}_0)} d\mathbf{z}_0 \\ D_{\text{KL}}(p(\mathbf{z}_0|\mathbf{z}_t)||p(\mathbf{z}_0|\mathbf{y})) &= \log p(\mathbf{y}) - \log p(\mathbf{z}_t) + \left(\int_{\mathbf{z}_0} p(\mathbf{z}_0|\mathbf{z}_t) \log p(\mathbf{z}_t|\mathbf{z}_0) d\mathbf{z}_0 \right) \\ &\quad - \left(\int_{\mathbf{z}_0} p(\mathbf{z}_0|\mathbf{z}_t) \log p(\mathbf{y}|\mathbf{z}_0) d\mathbf{z}_0 \right) \end{aligned}$$

{Note that $p(\mathbf{z}_t|\mathbf{z}_0) = \mathcal{N}(\mathbf{z}_0, \sigma_t^2 \mathbf{I})$ and we approximate $p(\mathbf{y}|\mathbf{z}_0) \approx \mathcal{N}(\mathcal{A}(\mathcal{D}(\mathbf{z}_0)), \sigma_y^2 \mathbf{I})$ }

$$\begin{aligned} D_{\text{KL}}(p(\mathbf{z}_0|\mathbf{z}_t)||p(\mathbf{z}_0|\mathbf{y})) &\approx -\log p(\mathbf{z}_t) - \frac{1}{2} \left(\int_{\mathbf{z}_0} p(\mathbf{z}_0|\mathbf{z}_t) \frac{\|\mathbf{z}_t - \mathbf{z}_0\|^2}{\sigma_t^2} d\mathbf{z}_0 \right) \\ &\quad + \frac{1}{2} \left(\int_{\mathbf{z}_0} p(\mathbf{z}_0|\mathbf{z}_t) \frac{\|\mathbf{y} - \mathcal{A}(\mathcal{D}(\mathbf{z}_0))\|^2}{\sigma_y^2} d\mathbf{z}_0 \right) + \text{C} \\ D_{\text{KL}}(p(\mathbf{z}_0|\mathbf{z}_t)||p(\mathbf{z}_0|\mathbf{y})) &\approx -\log p(\mathbf{z}_t) - \frac{1}{2\sigma_t^2} \left(\|\mathbf{z}_t\|^2 - 2\mathbf{z}_t^\top \mathbf{D}(\mathbf{z}_t, t) + \int_{\mathbf{z}_0} \|\mathbf{z}_0\|^2 p(\mathbf{z}_0|\mathbf{z}_t) d\mathbf{z}_0 \right) \\ &\quad + \frac{1}{2\sigma_y^2} \left(-2\mathbf{y}^\top \int_{\mathbf{z}_0} \mathcal{A}(\mathcal{D}(\mathbf{z}_0)) p(\mathbf{z}_0|\mathbf{z}_t) d\mathbf{z}_0 + \int_{\mathbf{z}_0} \|\mathcal{A}(\mathcal{D}(\mathbf{z}_0))\|^2 p(\mathbf{z}_0|\mathbf{z}_t) d\mathbf{z}_0 \right) + \text{C} \end{aligned}$$

We make a linear approximation of $\mathcal{A}(\mathcal{D}(\mathbf{z}_0))$ around $\hat{\mathbf{z}}_t = \mathbf{D}(\mathbf{z}_t, t) = \int_{\mathbf{z}_0} \mathbf{z}_0 p(\mathbf{z}_0|\mathbf{z}_t)$ as follows

$$\mathcal{A}(\mathcal{D}(\mathbf{z}_0)) \approx \mathcal{A}(\mathcal{D}(\hat{\mathbf{z}}_t)) + \frac{\partial \mathcal{A}(\mathcal{D}(\hat{\mathbf{z}}_t))}{\partial \hat{\mathbf{z}}_t} (\mathbf{z}_0 - \hat{\mathbf{z}}_t)$$

$$\begin{aligned} D_{\text{KL}}(p(\mathbf{z}_0|\mathbf{z}_t)||p(\mathbf{z}_0|\mathbf{y})) &\approx -\log p(\mathbf{z}_t) - \frac{1}{2\sigma_t^2} \left(\|\mathbf{z}_t\|^2 - 2\mathbf{z}_t^\top \mathbf{D}(\mathbf{z}_t, t) + \text{Tr} \{ \text{Cov}[\mathbf{z}_0|\mathbf{z}_t] \} + \|\mathbf{D}(\mathbf{z}_t, t)\|^2 \right) \\ &+ \frac{1}{2\sigma_y^2} \left(-2\mathbf{y}^\top \left\{ \mathcal{A}(\mathcal{D}(\hat{\mathbf{z}}_t)) + \frac{\partial \mathcal{A}(\mathcal{D}(\hat{\mathbf{z}}_t))}{\partial \hat{\mathbf{z}}_t} \int_{\mathbf{z}_0} (\mathbf{z}_0 - \hat{\mathbf{z}}_t) p(\mathbf{z}_0|\mathbf{z}_t) d\mathbf{z}_0 \right\} + \int_{\mathbf{z}_0} \|\mathcal{A}(\mathcal{D}(\mathbf{z}_0))\|^2 p(\mathbf{z}_0|\mathbf{z}_t) d\mathbf{z}_0 \right) + \text{C} \\ D_{\text{KL}}(p(\mathbf{z}_0|\mathbf{z}_t)||p(\mathbf{z}_0|\mathbf{y})) &\approx -\log p(\mathbf{z}_t) - \frac{\|\mathbf{D}(\mathbf{z}_t, t) - \mathbf{z}_t\|^2}{2\sigma_t^2} - \frac{1}{2\sigma_t^2} \text{Tr} \{ \text{Cov}[\mathbf{z}_0|\mathbf{z}_t] \} \end{aligned}$$

$$\begin{aligned}
& + \frac{1}{2\sigma_y^2} \left(-2\mathbf{y}^\top \mathcal{A}(\mathcal{D}(\hat{\mathbf{z}}_t)) + \|\mathcal{A}(\mathcal{D}(\hat{\mathbf{z}}_t))\|^2 + 2\mathcal{A}(\mathcal{D}(\hat{\mathbf{z}}_t))^\top \frac{\partial \mathcal{A}(\mathcal{D}(\hat{\mathbf{z}}_t))}{\partial \hat{\mathbf{z}}_t} \int_{\mathbf{z}_0} (\mathbf{z}_0 - \hat{\mathbf{z}}_t) p(\mathbf{z}_0 | \mathbf{z}_t) d\mathbf{z}_0 \right) \\
& + \frac{1}{2\sigma_y^2} \left(\int_{\mathbf{z}_0} \left\| \frac{\partial \mathcal{A}(\mathcal{D}(\hat{\mathbf{z}}_t))}{\partial \hat{\mathbf{z}}_t} (\mathbf{z}_0 - \hat{\mathbf{z}}_t) \right\|^2 p(\mathbf{z}_0 | \mathbf{z}_t) d\mathbf{z}_0 \right) + C \\
D_{\text{KL}}(p(\mathbf{z}_0 | \mathbf{z}_t) | | p(\mathbf{z}_0 | \mathbf{y})) & \approx -\log p(\mathbf{z}_t) - \frac{\|\mathbf{D}(\mathbf{z}_t, t) - \mathbf{z}_t\|^2}{2\sigma_t^2} - \frac{1}{2\sigma_t^2} \text{Tr} \{ \text{Cov}[\mathbf{z}_0 | \mathbf{z}_t] \} + \frac{\|\mathbf{y} - \mathcal{A}(\mathcal{D}(\hat{\mathbf{z}}_t))\|^2}{2\sigma_y^2} \\
& + \frac{1}{2\sigma_y^2} \left(\text{Tr} \left\{ \frac{\partial \mathcal{A}(\mathcal{D}(\hat{\mathbf{z}}_t))}{\partial \hat{\mathbf{z}}_t} \left(\int_{\mathbf{z}_0} (\mathbf{z}_0 - \hat{\mathbf{z}}_t) (\mathbf{z}_0 - \hat{\mathbf{z}}_t)^\top p(\mathbf{z}_0 | \mathbf{z}_t) d\mathbf{z}_0 \right) \frac{\partial \mathcal{A}(\mathcal{D}(\hat{\mathbf{z}}_t))^\top}{\partial \hat{\mathbf{z}}_t} \right\} \right) + C \\
D_{\text{KL}}(p(\mathbf{z}_0 | \mathbf{z}_t) | | p(\mathbf{z}_0 | \mathbf{y})) & \approx -\log p(\mathbf{z}_t) - \frac{\|\mathbf{D}(\mathbf{z}_t, t) - \mathbf{z}_t\|^2}{2\sigma_t^2} - \frac{1}{2\sigma_t^2} \text{Tr} \{ \text{Cov}[\mathbf{z}_0 | \mathbf{z}_t] \} + \frac{\|\mathbf{y} - \mathcal{A}(\mathcal{D}(\hat{\mathbf{z}}_t))\|^2}{2\sigma_y^2} \\
& + \frac{1}{2\sigma_y^2} \left(\text{Tr} \left\{ \frac{\partial \mathcal{A}(\mathcal{D}(\hat{\mathbf{z}}_t))}{\partial \hat{\mathbf{z}}_t} \text{Cov}[\mathbf{z}_0 | \mathbf{z}_t] \frac{\partial \mathcal{A}(\mathcal{D}(\hat{\mathbf{z}}_t))^\top}{\partial \hat{\mathbf{z}}_t} \right\} \right) + C \\
D_{\text{KL}}(p(\mathbf{z}_0 | \mathbf{z}_t) | | p(\mathbf{z}_0 | \mathbf{y})) & \approx -\log p(\mathbf{z}_t) - \frac{\|\mathbf{D}(\mathbf{z}_t, t) - \mathbf{z}_t\|^2}{2\sigma_t^2} - \frac{1}{2\sigma_t^2} \text{Tr} \{ \text{Cov}[\mathbf{z}_0 | \mathbf{z}_t] \} + \frac{\|\mathbf{y} - \mathcal{A}(\mathcal{D}(\mathbf{D}(\mathbf{z}_t, t)))\|^2}{2\sigma_y^2} \\
& + \frac{1}{2\sigma_y^2} \left(\text{Tr} \left\{ \frac{\partial \mathcal{A}(\mathcal{D}(\mathbf{D}(\mathbf{z}_t, t)))}{\partial \mathbf{D}(\mathbf{z}_t, t)} \text{Cov}[\mathbf{z}_0 | \mathbf{z}_t] \frac{\partial \mathcal{A}(\mathcal{D}(\mathbf{D}(\mathbf{z}_t, t)))^\top}{\partial \mathbf{D}(\mathbf{z}_t, t)} \right\} \right) + C
\end{aligned}$$

□

D.2 SIMPLIFIED VML AND LATENT VML-MAP FOR LDMS

Similar to the case of pixel diffusion models, the higher-order terms involving $\text{Cov}[\mathbf{z}_0 | \mathbf{z}_t]$ in VML^{LDM} converge to a constant as $t \rightarrow 0$, under mild assumptions on \mathcal{A} , and \mathcal{D} . Ignoring the higher-order terms, we define the simplified VML^{LDM} objective and its gradient as follows.

Simplified VML^{LDM} and its gradient. For a linear degradation matrix \mathbf{H} , the simplified VML^{LDM} (i.e., $\text{VML}_S^{\text{LDM}}$) and its gradient is given by

$$\begin{aligned}
\text{VML}_S^{\text{LDM}} & = -\log p(\mathbf{z}_t) - \frac{\|\mathbf{D}(\mathbf{z}_t, t) - \mathbf{z}_t\|^2}{2\sigma_t^2} + \frac{\|\mathbf{y} - \mathbf{H}\mathbf{D}(\mathbf{D}(\mathbf{z}_t, t))\|^2}{2\sigma_y^2} \\
\nabla_{\mathbf{z}_t} \text{VML}_S^{\text{LDM}} & = \underbrace{-\frac{\partial \mathbf{D}^\top(\mathbf{z}_t, t)}{\partial \mathbf{z}_t} \frac{\partial \mathbf{D}^\top(\mathbf{D}(\mathbf{z}_t, t))}{\partial \mathbf{D}(\mathbf{z}_t, t)} \mathbf{H}^\top (\mathbf{y} - \mathbf{H}\mathbf{D}(\mathbf{D}(\mathbf{z}_t, t)))}_{\text{measurement consistency gradient}} \underbrace{- \frac{\partial \mathbf{D}^\top(\mathbf{z}_t, t)}{\partial \mathbf{z}_t} \frac{(\mathbf{D}(\mathbf{z}_t, t) - \mathbf{z}_t)}{\sigma_t^2}}_{\text{prior gradient}}
\end{aligned}$$

Proof.

$$\begin{aligned}
\nabla_{\mathbf{z}_t} \text{VML}_S^{\text{LDM}} & = \{-\nabla_{\mathbf{z}_t} \log p(\mathbf{z}_t)\} - \left\{ \nabla_{\mathbf{z}_t} \frac{\|\mathbf{D}(\mathbf{z}_t, t) - \mathbf{z}_t\|^2}{2\sigma_t^2} \right\} + \left\{ \nabla_{\mathbf{z}_t} \frac{\|\mathbf{y} - \mathbf{H}\mathbf{D}(\mathbf{D}(\mathbf{z}_t, t))\|^2}{2\sigma_y^2} \right\} \\
\nabla_{\mathbf{z}_t} \text{VML}_S^{\text{LDM}} & = \left\{ -\frac{\mathbf{D}(\mathbf{z}_t, t) - \mathbf{z}_t}{\sigma_t^2} \right\} - \left\{ \left(\frac{\partial \mathbf{D}^\top(\mathbf{z}_t, t)}{\partial \mathbf{z}_t} \frac{(\mathbf{D}(\mathbf{z}_t, t) - \mathbf{z}_t)}{\sigma_t^2} \right) - \frac{\mathbf{D}(\mathbf{z}_t, t) - \mathbf{z}_t}{\sigma_t^2} \right\} \\
& + \left\{ -\frac{\partial \mathbf{D}^\top(\mathbf{z}_t, t)}{\partial \mathbf{z}_t} \frac{\partial \mathbf{D}^\top(\mathbf{D}(\mathbf{z}_t, t))}{\partial \mathbf{D}(\mathbf{z}_t, t)} \mathbf{H}^\top (\mathbf{y} - \mathbf{H}\mathbf{D}(\mathbf{D}(\mathbf{z}_t, t)))}{\sigma_y^2} \right\} \\
\nabla_{\mathbf{z}_t} \text{VML}_S^{\text{LDM}} & = -\frac{\partial \mathbf{D}^\top(\mathbf{z}_t, t)}{\partial \mathbf{z}_t} \frac{\partial \mathbf{D}^\top(\mathbf{D}(\mathbf{z}_t, t))}{\partial \mathbf{D}(\mathbf{z}_t, t)} \mathbf{H}^\top (\mathbf{y} - \mathbf{H}\mathbf{D}(\mathbf{D}(\mathbf{z}_t, t))) - \frac{\partial \mathbf{D}^\top(\mathbf{z}_t, t)}{\partial \mathbf{z}_t} \frac{(\mathbf{D}(\mathbf{z}_t, t) - \mathbf{z}_t)}{\sigma_t^2}
\end{aligned}$$

□

We present LatentVML-MAP (Algorithm 2) as an extension of VML-MAP (Algorithm 1) to LDMS. In principle, LatentVML-MAP minimizes $\text{VML}_S^{\text{LDM}}$ at each reverse diffusion step to find $\mathbf{z}_0^* = \arg \max_{\mathbf{z}_0} \log p(\mathbf{z}_0 | \mathbf{y})$ and finally, uses the decoder \mathcal{D} to return $\mathbf{x}_0 = \mathcal{D}(\mathbf{z}_0^*)$.

The inputs to Algorithm 2 consists of the latent diffusion denoiser $\mathbf{D}_\theta(\cdot, \cdot)$, the decoder $\mathcal{D}(\cdot)$, the linear degradation matrix \mathbf{H} , the measurement \mathbf{y} with noise variance σ_y^2 , the diffusion noise schedule $\sigma(\cdot)$, the total number of reverse diffusion steps N with the discretized time step schedule specified by $t_{i \in \{0, \dots, N\}}$, where $t_0 = 0$, the gradient descent iterations per step given by K , and the learning rate γ . We use the notations $\sigma(t)$ and σ_t interchangeably.

Algorithm 2: LatentVML-MAP**Input:** $D_\theta(\cdot, \cdot), \mathcal{D}(\cdot), H, \mathbf{y}, \sigma_{\mathbf{y}}, \sigma(\cdot), t_{i \in \{0, \dots, N\}}, K, \gamma$ **Output:** \mathbf{x}_{t_0} **Initialize** $\mathbf{z}_{t_N} \sim \mathcal{N}(\mathbf{0}, \sigma_{t_N}^2 \mathbf{I})$ **for** $i \leftarrow N$ **to** 1 **do** **for** $j \leftarrow 1$ **to** K **do** $\mathbf{z}_{t_i} \leftarrow \mathbf{z}_{t_i} - \gamma \cdot \nabla_{\mathbf{z}_{t_i}} \text{VML}_S^{\text{LDM}}$

/* see Appendix D.2 */

end $\mathbf{z}_{t_{i-1}} \sim \mathcal{N}(D_\theta(\mathbf{z}_{t_i}, t_i), \sigma_{t_{i-1}}^2 \mathbf{I})$ **end** $\mathbf{x}_{t_0} = \mathcal{D}(\mathbf{z}_{t_0})$ **Return** \mathbf{x}_{t_0}

D.3 EXPERIMENTS ON CELEBA

We conduct experiments on 100 test images from the CelebA (Liu et al., 2015) dataset on Half-mask inpainting, Box-mask inpainting, $4\times$ Super-resolution, and Deblurring tasks using the pre-trained latent diffusion model and the autoencoder from Rombach et al. (2022). For LatentVML-MAP, we fix $N = 20$, $K = 50$, and use the EDM scheduler with $\sigma_{min} = 0.002$, $\sigma_{max} = 80$, and $\rho = 7$. For Resample (Song et al., 2024), we follow the default settings with 500 time steps, and for LatentDAPS (Zhang et al., 2025), instead of the default setting of 100 neural function evaluations, we select 1000 NFEs to allow for a fair comparison, keeping other parameters fixed. We set $\sigma_y = 1e-9$ for all the methods. For LatentVML-MAP, we report the best learning rate configuration for each task as $\gamma_0 \cdot \sigma_y^2$, with γ_0 as follows: Half-mask inpainting ($\gamma_0 = 0.1$), Box-mask inpainting ($\gamma_0 = 0.1$), $4\times$ Super-resolution ($\gamma_0 = 0.9$), Deblurring ($\gamma_0 = 0.075$).

Table 7: Evaluation of LDM-based image restoration methods on Half-mask inpainting, Box-mask inpainting, $4\times$ super-resolution, Deblurring on 100 images of CelebA256. Excluding Resample, we denote the best values in **bold**, second best values underlined.

| Dataset | Method | Half-Inpaint | | Box-Inpaint | | $4\times$ Sup-res | | Deblurring | |
|---------|----------------------|--------------|--------------|--------------|--------------|-------------------|--------------|--------------|--------------|
| | | LPIPS↓ | FID↓ | LPIPS↓ | FID↓ | LPIPS↓ | FID↓ | LPIPS↓ | FID↓ |
| CelebA | Resample | 0.235 | 55.80 | 0.092 | 53.12 | 0.084 | 41.85 | 0.202 | 63.41 |
| | LatentDAPS-1K | <u>0.240</u> | <u>54.47</u> | 0.070 | <u>37.69</u> | 0.114 | <u>38.72</u> | 0.328 | 104.2 |
| | Resample w/o PO | 0.259 | 64.32 | 0.106 | 68.30 | <u>0.112</u> | 50.95 | 0.210 | 65.36 |
| | LatentVML-MAP | 0.208 | 47.04 | <u>0.074</u> | 34.73 | 0.099 | 36.82 | <u>0.212</u> | <u>71.43</u> |

For LatentVML-MAP, we observed that even for linear inverse problems, the optimization becomes more challenging due to the non-linearity of the LDM decoder. As a result, we noticed that the final reconstructed images are blurry and inconsistent with the measurement \mathbf{y} . We also observed this pattern with LatentDAPS, but not with Resample, as it uses pixel-space optimization. Note that Resample also requires the LDM encoder to project the pixel-space-optimized result back into the latent space, unlike LatentDAPS and LatentVML-MAP. For a fair comparison, in our experiments,

- we also report the performance of Resample with its pixel-space optimization replaced with latent-space optimization (see Song et al. (2024)), denoted as **Resample w/o PO**
- we project the final reconstructed images of all methods onto the measurement subspace to ensure that all the reconstructed images are consistent with the measurements.

For inpainting, the measurement subspace projection implies that we paste the observed pixels back into the reconstructed images. For super-resolution and deblurring, we paste back the observations in the spectral space of the linear operator by using the SVD accessible operators from Kawar et al. (2022). Results from Table 7 validate the effectiveness of LatentVML-MAP in practice. We believe that the performance bottleneck primarily exists due to the challenging optimization and that improved optimization techniques can further enhance the performance of LatentVML-MAP.



Figure 16: Left to right: original image, measurement, restored images with LatentDAPS-1K, Resample w/o PO, and LatentVML-MAP. Zoom in for the best view.

1401
1402
1403