Dual Ascent Diffusion for Inverse Problems

Minseo Kim Axel Levy Gordon Wetzstein
Stanford University
{kminseo, axlevy, gordonwz}@stanford.edu

Abstract

Ill-posed inverse problems are fundamental in many domains, ranging from astrophysics to medical imaging. Emerging diffusion models provide a powerful prior for solving these problems. Existing maximum-a-posteriori (MAP) or posterior sampling approaches, however, rely on different computational approximations, leading to inaccurate or suboptimal samples. To address this issue, we introduce a new approach to solving MAP problems with diffusion model priors using a dual ascent optimization framework. Our framework achieves better image quality as measured by various metrics for image restoration problems, it is more robust to high levels of measurement noise, it is faster, and it estimates solutions that represent the observations more faithfully than the state of the art.

1. Introduction

We are interested in solving inverse problems, where an unknown image or signal $\mathbf{x}$ is estimated from noisy and corrupted observations $\mathbf{y}$. These types of problems arise across science and engineering, for example, in image restoration [25], astrophysics [1], medical imaging [37], protein structure determination [14, 23, 26], among other domains. In all cases, a linear or nonlinear function $\mathcal{A}(\cdot)$ models a domain-specific image formation process. Although the likelihood of observations $p(\mathbf{y}|\mathbf{x})$ depends on the statistical model of the noise in the observations, closed-form expressions exist for specific cases. For example, the image formation model for zero-mean Gaussian i.i.d. noise with variance σ^2 is $\mathbf{y} = \mathcal{A}(\mathbf{x}) + \mathcal{N}(\mathbf{0}, \sigma^2)$ and its log-likelihood is $\log p(\mathbf{y}|\mathbf{x}) = \frac{-1}{2\sigma^2}\|\mathbf{y} - \mathcal{A}(\mathbf{x})\|_2^2$, up to an additive term that does not depend on $\mathbf{x}$. Given the likelihood of the observations and a prior $p(\mathbf{x})$, inverse problem solvers aim at either *maximizing* or *sampling from* the posterior $p(\mathbf{x}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{x})p(\mathbf{x})$ in a Bayesian framework. Most inverse problems are ill-posed, making the prior a crucial component of the solution-finding process.

Maximum-a-posteriori (MAP) approaches aim at maximizing the posterior to find the most likely solution given a prior. Early approaches used "hand-crafted" priors to promote smoothness, piece-wise constancy via Total Variation [3, 31], or sparsity in a transform domain [12], while most modern approaches use some form of neural network [44]. The plug-n-play (PnP) approach [39], for example implemented by the Alternating Direction Method of Multipliers (ADMM) [4] algorithm, is a popular and versatile framework to solve MAP problems by leveraging (Gaussian) denoisers as priors. MAP finds the single, most likely solution to an inverse problem. However, oftentimes one is interested in sampling from the posterior of all feasible solutions. For this reason, many recent works [6, 7, 15, 19, 21, 22, 27, 34, 35, 41, 43] focus on posterior sampling using powerful pretrained diffusion models as priors, as surveyed in [9]. While these recent diffusion posterior sampling methods show great promise, they are all fundamentally limited by the optimization framework that is used to combine the likelihood of the image formation model and the prior during optimization.

In this work, we do not aim to develop a method that provably samples from the posterior, but instead focus on deriving an optimization strategy that accurately and efficiently solves the MAP problem using a prior given by a pretrained diffusion model and a dual ascent–based optimization framework inspired by ADMM [4]. Our approach, dubbed *DDiff*, is faster and shown to achieve better reconstruction quality compared with the state of the art for image restoration problems, including single-image super resolution, inpainting, deblurring, phase retrieval, and high-dynamic range imaging. Moreover, DDiff is more robust to high levels of measurement noise, and our reconstructions more faithfully model the observations by exhibiting closer-to-zero residuals than existing methods. The latter is important because the log-likelihood of the observations $\log p(\mathbf{y}|\mathbf{x})$ is an indicator for the level of hallucination a generative prior, such as a diffusion model, introduces when computing a solution. Notably, we establish the first fixed-point convergence proof for a diffusion-based posterior optimization method and extend our framework to a latent setting for efficient inference in compact feature spaces.

2. Background on Inverse Problems

2.1. Maximum-a-Posteriori Solutions

A maximum-a-posteriori (MAP) solution aims to find the solution $\mathbf{x}_{\text{MAP}}$ that maximizes the posterior $p(\mathbf{x}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{x})\, p(\mathbf{x})$. Typically, this is done by minimizing the negative log-likelihood as

$$\begin{aligned}\mathbf{x}_{\text{MAP}} &= \underset{\mathbf{x}}{\operatorname{argmin}} \; -\big(\log p(\mathbf{y} \mid \mathbf{x}) + \log p(\mathbf{x})\big) \\ &= \frac{1}{2\sigma^2} \left\|\mathbf{y} - \mathcal{A}(\mathbf{x})\right\|_2^2 \; - \; \log p(\mathbf{x})\end{aligned} \tag{1}$$

The alternating direction method of multipliers (ADMM) [4] is a common approach to solving the MAP problem. ADMM attempts to blend the benefits of dual decomposition and augmented Lagrangian methods for constrained optimization. For this purpose, a slack variable $\mathbf{z}$ is introduced to split the objective function in Eq. 1 into a data fidelity term $-\log p(\mathbf{y}|\mathbf{x})$ and the log-prior term $-\log p(\mathbf{z})$, subject to $\mathbf{x} = \mathbf{z}$. ADMM then forms the Augmented Lagrangian of the split formulation as

$$\begin{aligned}L_\rho(\mathbf{x}, \mathbf{z}, \mathbf{u}) = &\frac{1}{2\sigma^2} \left\|\mathbf{y} - \mathcal{A}(\mathbf{x})\right\|_2^2 - \log p(\mathbf{z}) \\ &+ \frac{\rho}{2} \left\|\mathbf{x} - \mathbf{z} + \mathbf{u}\right\|_2^2 - \frac{\rho}{2} \left\|\mathbf{u}\right\|_2^2,\end{aligned} \tag{2}$$

where $\mathbf{u}$ is the dual variable and ρ is a hyperparameter that defines the strength of the soft constraints. ADMM then applies an alternating gradient descent approach to minimizing the Augmented Lagrangian, resulting in a set of updates on $\mathbf{x}, \mathbf{z}, \mathbf{u}$ that are applied in an iterative fashion:

$$\mathbf{x} \leftarrow \underset{\mathbf{x}}{\operatorname{argmin}} \frac{1}{2\sigma^2} \left\|\mathbf{y} - \mathcal{A}(\mathbf{x})\right\|_2^2 + \frac{\rho}{2} \|\mathbf{x} - \mathbf{z} + \mathbf{u}\|_2^2 \tag{3}$$

$$\mathbf{z} \leftarrow \underset{\mathbf{z}}{\operatorname{argmin}} - \log p(\mathbf{z}) + \frac{\rho}{2} \|\mathbf{x} - \mathbf{z} + \mathbf{u}\|_2^2 \tag{4}$$

$$= \mathcal{D}\left(\mathbf{x} + \mathbf{u}, \tilde{\sigma}^2 = \frac{1}{\rho}\right) \tag{5}$$

$$\mathbf{u} \leftarrow \mathbf{u} + \mathbf{x} - \mathbf{z}. \tag{6}$$

Here, the $\mathbf{x}$-update is an unconstrained least-squares problem that does not depend on the prior and which often has a closed-form, or at least an efficient, solution. An important insight of plug-and-play image restoration methods [5, 39] is the fact that the $\mathbf{z}$-update (Eq. 5) is a denoising problem on the variable $\mathbf{x} + \mathbf{u}$, which can be solved using any Gaussian denoiser $\mathcal{D}\left(\cdot, \tilde{\sigma}^2\right)$ assuming the noise level is $\tilde{\sigma}$.

Dual ascent, for example implemented by ADMM, offers several benefits in traditional optimization, including the ability to leverage convexity in the dual problem, leading to simpler, more efficient, and more robust optimization methods [4].

2.2. Diffusion Models and Posterior Sampling

The key insight of diffusion models lies in the fact that one can sample from a target distribution $p_0(\mathbf{x})$ by first sampling $\mathbf{x}_T$ from another distribution p_T that is easy to sample from, e.g., a Gaussian, and iteratively applying a *reverse* diffusion step of the form

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t + (1 - \alpha_t)\, \mathbf{s}_\theta(\mathbf{x}_t, t)\right) + \sqrt{1 - \alpha_t}\, \epsilon \tag{7}$$

for $t = T, \ldots, 1$ to generate an image $\mathbf{x}_0$, where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The reverse diffusion process [2, 17] approximates the inverse trajectories of a corresponding *forward* diffusion $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\, \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\, \epsilon$, $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Here, we adopt the variance-preserving form of forward and reverse diffusion [18]. The factors α_t and $\bar{\alpha}_t = \prod_{s=0}^{t} \alpha_s$ are derived from the noise schedule of the diffusion model [18]. Importantly, the score network $\mathbf{s}_\theta(\mathbf{x}_t, t)$, defined by parameters θ, is a neural network that approximates the *score function* $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ [36]. This network is learned from training data in the diffusion model pretraining stage. To sample more efficiently, Denoising Diffusion Implicit Models (DDIM) [33] provides an alternative non-Markovian reverse parameterization of the diffusion process, replacing the Markovian formulation in Eq. 7:

$$\begin{aligned}\mathbf{x}_{t-1} = &\sqrt{\bar{\alpha}_{t-1}} \left(\frac{\mathbf{x}_t + (1 - \bar{\alpha}_t)\mathbf{s}_\theta(\mathbf{x}_t, t)}{\sqrt{\bar{\alpha}_t}}\right) \\ &- \sqrt{1 - \bar{\alpha}_{t-1} - \sigma_t^2} \left(\sqrt{1 - \bar{\alpha}_t} \cdot \mathbf{s}_\theta(\mathbf{x}_t, t)\right) + \sigma_t \epsilon,\end{aligned} \tag{8}$$

where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and σ_t is a schedule chosen at inference.

In a posterior sampling problem, we aim at sampling from the posterior $p(\mathbf{x}|\mathbf{y})$. For this purpose, many recent methods follow the approach described above, using the *posterior score* $\nabla_{\mathbf{x}} \log p_t(\mathbf{x}|\mathbf{y}) = \nabla_{\mathbf{x}} \log p(\mathbf{y}|\mathbf{x}_t = \mathbf{x}) + \nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ instead of $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$. The second term is equivalent to the unconditional score of the pretrained diffusion model, but the challenge lies in the first, i.e., the conditional score term. The conditional probability $p(\mathbf{y}|\mathbf{x}_t)$ can be written as a conditional expectation $\mathbb{E}_{\mathbf{x}_0 \sim p(\mathbf{x}_0|\mathbf{x}_t)}[p(\mathbf{y}|\mathbf{x}_0)]$, but approximating this expectation with Monte Carlo samples is computationally intractable (see Chung et al. [7], for example). For this reason, existing diffusion posterior sampling methods approximate this conditional distribution with a Dirac delta distribution concentrated on $\mathbf{x}_t$ [19] or on $\mathbb{E}_{\mathbf{x}_0 \sim p(\mathbf{x}_0|\mathbf{x}_t)}[\mathbf{x}_0]$ [7]. A number of other diffusion posterior sample methods have been proposed [9], each providing a different approximation for the conditional expectation. Most recently, Zhang et al. [43] introduced DAPS (Decoupled Annealing Posterior Sampling), a two-step iterative approach that mitigates the accumulation of errors along the sampling trajectory through a Markov chain Monte Carlo–based equilibration step at $t = 0$. DAPS demonstrated state-of-the-art results on both

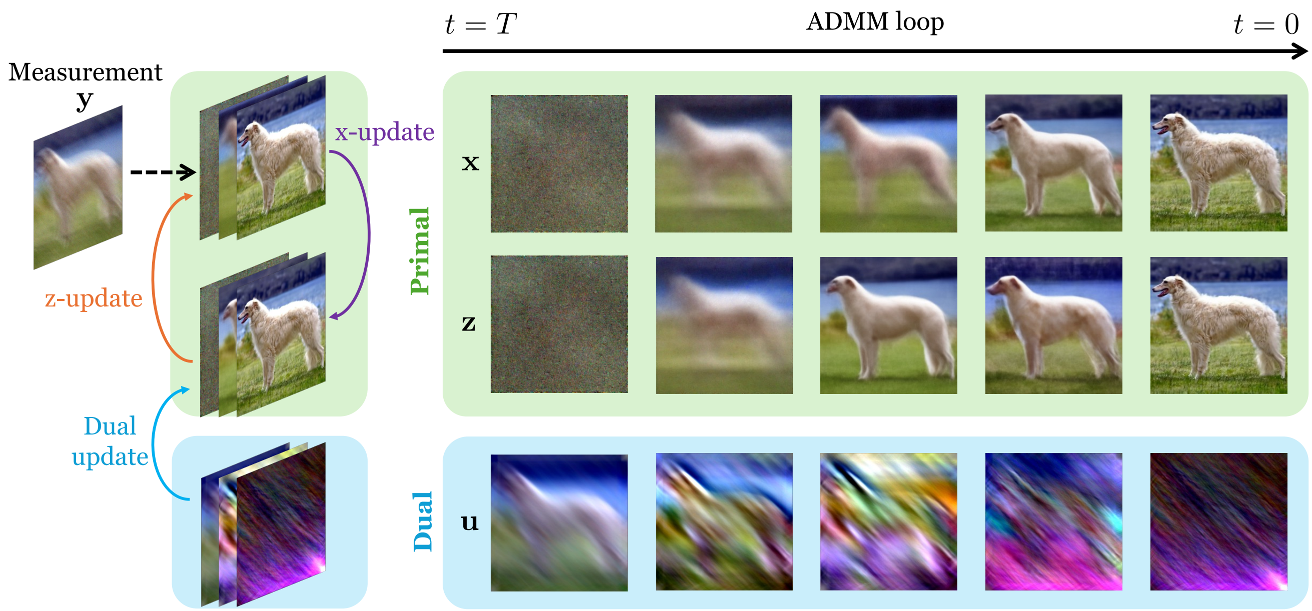


Figure 1. **Overview of DDiff.** This example illustrates motion deblurring. DDiff alternates between three updates ($\mathbf{x}$-, $\mathbf{z}$-, and dual updates) within each ADMM iteration. As iterations proceed, the primal variables $\mathbf{x}$ and $\mathbf{z}$ progressively align, while the dual variable $\mathbf{u}$ diminishes toward zero, indicating convergence to a fixed point $(\mathbf{x}^\star, \mathbf{z}^\star, \mathbf{u}^\star)$. This evolution visually demonstrates the fixed-point convergence behavior analyzed in Appendix A. Here, the ADMM iteration index and diffusion timestep are set equal by design.

linear and nonlinear inverse problems, but remains limited by the necessity to approximate the conditional probability $p(\mathbf{x}_0|\mathbf{x}_t)$ and by the finiteness of the number of MCMC steps.

3. Method

We derive our approach to solving MAP problems with pretrained diffusion model priors in the following.

3.1. Diffusion Plug-and-Play ADMM for Image Restoration

We dub the naive approach for using ADMM with a pretrained diffusion model *Diff-PnP-ADMM*. For this purpose, we apply the ADMM framework as discussed in Sec. 2.1 and use the pretrained diffusion model as a one-step denoiser $\mathcal{D}$ in the $\mathbf{z}$-update (i.e., Eq. 5). This is done by applying Tweedie's formula [13] and replacing the $\mathbf{z}$-update with

$$\mathbf{z} \leftarrow \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x} + \mathbf{u} + (1 - \bar{\alpha}_t)\mathbf{s}_\theta(\mathbf{x} + \mathbf{u}, t)) \tag{9}$$

where t decreases at each step of the ADMM loop, which implicitly constrains the relationship between the diffusion schedule α_t and the soft constraint parameter ρ.

3.2. Dual Ascent Diffusion (DDiff)

At its core, the ADMM method iterates over 3 steps. The $\mathbf{x}$-update corresponds to a data matching step (Eq. 3), the $\mathbf{z}$-update can be seen as a denoising step (Eq. 5) and the dual update stems from the introduction of the dual variable $\mathbf{u}$ for the constraint $\mathbf{x} = \mathbf{z}$ (Eq. 6). While the $\mathbf{x}$ and dual updates are straightforwardly derived from the original ADMM framework, our methodological contribution primarily consists of showing that a pretrained diffusion model can be used more efficiently in the $\mathbf{z}$-update than the naive approach (Eq. 9).

$\mathbf{x}$-update. In order for the method to be directly applicable to any differentiable forward model $\mathcal{A}$, whether linear or nonlinear, we replace the minimization problem of Eq. 3 with a single gradient step

$$\mathbf{x} \leftarrow \mathbf{v} - \gamma \nabla_{\mathbf{v}} \|\mathbf{y} - \mathcal{A}(\mathbf{v})\|_2^2, \quad \mathbf{v} = \mathbf{z} - \mathbf{u}, \tag{10}$$

where γ is a step size that can be adjusted at each iteration. This approach is also known as linearized ADMM [28].

$\mathbf{u}$-update. The update of the dual variable is readily available in Eq. 6.

$\mathbf{z}$-update. Following Eq. 5, the $\mathbf{z}$-update consists in denoising $\mathbf{x} + \mathbf{u}$ for a certain noise level $\tilde{\sigma}^2$. However, it is crucial to note that since the score model $\mathbf{s}_\theta(\mathbf{x}, t)$ is only trained on points sampled from p_t, it is a poor approximation of the true score whenever $\mathbf{x}$ is unlikely under p_t (i.e., when $p_t(\mathbf{x}) \ll 1$). In other words, the score model at time t is only accurate on points belonging to the diffusion manifold at time t [8]. Because $\mathbf{x} + \mathbf{u}$ does not in general belong to this manifold, we propose to replace the $\mathbf{z}$-update with

$$\mathbf{z} \leftarrow \frac{1}{\sqrt{\bar{\alpha}_t}} \left(\mathbf{x}_t + (1 - \bar{\alpha}_t) \mathbf{s}_\theta(\mathbf{x}_t, t) \right), \tag{11}$$

where $\mathbf{x}_t$ is defined recursively following $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and

$$\mathbf{x}_{t-1} \leftarrow \underbrace{\sqrt{\bar{\alpha}_{t-1}} \cdot \mathbf{x} + \sqrt{1 - \bar{\alpha}_{t-1} - \sigma_t^2} \cdot \hat{\epsilon} + \sigma_t \epsilon}_{\text{DDIM update}} + \underbrace{\sqrt{\bar{\alpha}_{t-1}} \cdot \mathbf{u}}_{\text{Re-scaled } \mathbf{u}}. \tag{12}$$

In this equation, $\hat{\epsilon} = (\mathbf{x}_t - \sqrt{\bar{\alpha}_t} \cdot \mathbf{x}) / \sqrt{1 - \bar{\alpha}_t}$ and σ_t is a hyperparameter. The first part of the right-hand side corresponds to the DDIM update where the "predicted $\mathbf{x}_0$" is $\mathbf{x}$. The second part adds the dual variable $\mathbf{u}$, re-scaled to match the signal level of $\mathbf{x}_{t-1}$.

Combined iteratively, these three steps define our method (Algorithm 1). We show that the DDiff updates converge to a fixed point $(\mathbf{x}^\star, \mathbf{z}^\star, \mathbf{u}^\star)$: under bounded denoiser and gradient assumptions, all three iterate sequences are Cauchy and converge to a limit (Theorem 1); notably, this result does not require a convex prior. Full assumptions, proofs, and technical details are in Appendix A.

3.3. DDiff with Latent Diffusion Model

Given a pretrained encoder $\mathcal{E}$ and decoder $\mathcal{D}$, latent diffusion models (LDMs) [29] are trained to model the distribution $p(\mathbf{z}_0)$ of latent variables $\mathbf{z}_0 = \mathcal{E}(\mathbf{x}_0)$, enabling reconstruction via $\mathbf{x}_0 = \mathcal{D}(\mathbf{z}_0)$. Leveraging latent diffusion models typically offers reduced memory and compute demands compared to pixel-space diffusion. Extending DDiff to this latent domain yields LatentDDiff, which performs alternating data-fidelity and denoising updates on compact latent representations while preserving the dual-ascent structure of DDiff. The full algorithmic formulation, quantitative results, and discussion of enforcing data consistency either in latent or pixel space are provided in Appendix C.

3.4. Comparison to Other Diffusion-based Variable Splitting Methods

Prior methods, such as DiffPIR [46], DCDP [24], and PnP-DM [42], are MAP-based optimization methods, specifically built upon the half-quadratic splitting (HQS) method [16], which lacks dual variables in its formulation. Our work naturally extends existing frameworks, transforming them from HQS-style approaches to those using dual variables, including ADMM. By incorporating Lagrange multipliers that accumulate constraint violations across the iterations, DDiff offers improved empirical performance on challenging inverse problems where measurement consistency is crucial. In particular, we note that removing the dual update from DDiff would exactly emulate DiffPIR [46] (with the right choice of σ_t and γ_t, see Appendix E).

Algorithm 1 DDiff

Require: T, $\mathcal{A}(\cdot)$, $\{\sigma_t\}_{t=1}^T$, $\{\bar{\alpha}_t\}_{t=1}^T$, $\mathbf{s}_\theta$, $\mathbf{y}$, $\{\gamma_t\}_{t=1}^T$, t_0

Initialize $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \mathbf{u} = \mathbf{0}$.
for $t = T - 1$ **to** 0 **do**
  $\mathbf{z} \leftarrow \frac{1}{\sqrt{\bar{\alpha}_t}} \left(\mathbf{x}_t + (1 - \bar{\alpha}_t) \mathbf{s}_\theta(\mathbf{x}_t, t) \right)$
  ▷ *Denoising step (Eq. 11)*
  $\mathbf{x} \leftarrow \mathbf{z} - \mathbf{u} - \gamma_t \nabla_{\mathbf{v}=\mathbf{z}-\mathbf{u}} \|\mathbf{y} - \mathcal{A}(\mathbf{v})\|^2$
  ▷ *Measurement step (Eq. 10)*
  $\hat{\epsilon} \leftarrow \frac{1}{\sqrt{1-\bar{\alpha}_t}} \left(\mathbf{x}_t - \sqrt{\bar{\alpha}_t} \cdot \mathbf{x} \right)$
  $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ **if** $t > t_0$ **else** $\epsilon = 0$
  $\mathbf{x}_{t-1} \leftarrow \sqrt{\bar{\alpha}_{t-1}} \cdot \mathbf{x} + \sqrt{1 - \bar{\alpha}_{t-1} - \sigma_t^2} \cdot \hat{\epsilon} + \sigma_t \epsilon + \sqrt{\bar{\alpha}_{t-1}} \cdot \mathbf{u}$ ▷ *Reverse diffusion (Eq. 12)*
  $\mathbf{u} \leftarrow \mathbf{u} + \mathbf{x} - \mathbf{z}$ ▷ *Dual update (Eq. 6)*
end for
return $\mathbf{x}_0$

4. Experiments

4.1. Experimental Setup

Datasets and metrics. We evaluate our method on two image datasets, FFHQ 256×256 [20] and ImageNet 256×256 [10]. For pixel-space diffusion models, we utilize pretrained models from [7] on the FFHQ dataset and from [11] on the ImageNet dataset. For latent diffusion models, we used unconditional LDM-VQ4 (autoencoder with a downsampling factor of 4) pretrained models from [32] for FFHQ and [29] for ImageNet. We randomly selected 100 images from the validation set for both datasets, and the images were normalized to $[-1, 1]$. Our main evaluation metrics include peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM), learned perceptual image patch similarity (LPIPS) [45], and residual error, defined as $\|\mathbf{y} - \mathcal{A}(\mathbf{x})\|_2^2 - \sigma^2$ (see Appendix B for details), which quantifies the degree of data consistency.

Inverse problems. Our method is evaluated on multiple inverse problems. **Linear tasks** include super-resolution ($4\times$ downsampling), Gaussian deblurring, motion deblurring, inpainting with a 128×128 box, and inpainting with a random mask that removes 70% of the pixels. **Nonlinear tasks** consist of phase retrieval (oversampling ratio of 2.0), for which we report the best result out of five runs due to the intrinsic instability of the task, nonlinear deblurring, and high dynamic range ($2\times$ dynamic range). All measurements include additive Gaussian noise ($\sigma = 0.05$).

Baselines. We compare against the state of the art: DAPS [43], DMPlug [40], DCDP [24], RED-diff [27], DDRM [22], DPS [7], and DiffPIR [46]. Notably, DiffPIR and DDRM were not proven to handle nonlinear tasks. We closely evaluate our method against DAPS as it achieves the best results

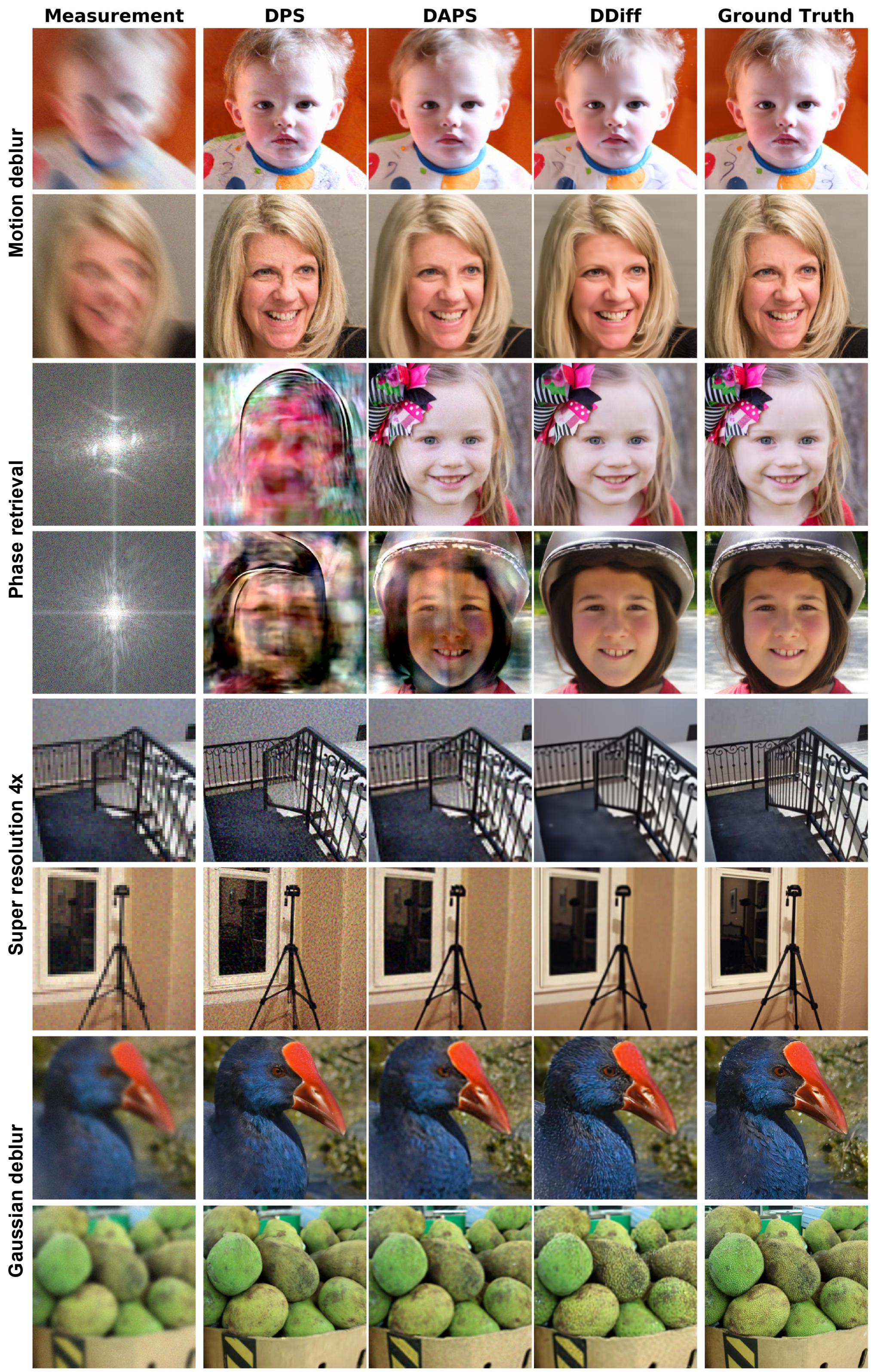


Figure 2. **Qualitative results.** DDiff demonstrates sharper and cleaner results compared to DPS [7] and DAPS [43]. All tasks are run with a noise of standard deviation $\sigma = 0.05$.

Table 1. **Quantitative evaluation.** Comparing different methods for 5 linear and 3 nonlinear tasks on FFHQ and ImageNet datasets. This evaluation uses 100 validation images and reports the average metric value. The best and second-best results are distinguished by **bold** and underlined marks, respectively. All tasks are run with a noise of standard deviation $\sigma = 0.05$. See Appendix D for 95% confidence intervals.

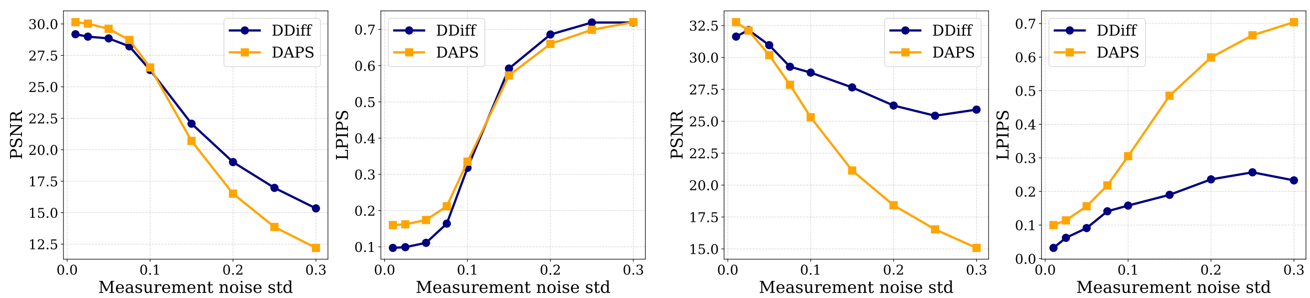
Task	Method	FFHQ				ImageNet			
		PSNR (↑)	SSIM (↑)	LPIPS (↓)	Residual (↓)	PSNR (↑)	SSIM (↑)	LPIPS (↓)	Residual (↓)
Super Resolution 4×	DDiff (ours)	**30.07**	**0.824**	0.211	**0.0028**	**25.81**	**0.656**	0.396	**0.0038**
	DAPS	29.34	0.783	**0.190**	0.0029	25.44	0.636	**0.295**	0.0047
	DMPlug	28.55	0.742	0.220	0.0038	24.22	0.649	0.432	0.0039
	DPS	24.42	0.486	0.346	0.0050	21.10	0.351	0.408	0.0052
	DiffPIR	23.71	0.440	0.423	0.0087	20.75	0.312	0.517	0.0093
	DCDP	26.65	0.641	0.410	0.0079	23.51	0.542	0.460	0.0060
	DDRM	26.32	0.763	0.286	0.0075	22.26	0.513	0.473	0.0072
Inpainting (Box)	DDiff (ours)	**24.88**	**0.831**	0.110	0.0077	21.15	**0.743**	0.240	**0.0119**
	DAPS	24.12	0.742	0.174	0.0099	**21.22**	0.714	**0.230**	0.0150
	DPS	23.68	0.810	**0.079**	**0.0033**	19.63	0.725	0.254	0.0412
	DiffPIR	19.02	0.527	0.252	0.0106	16.02	0.520	0.329	0.0145
	DCDP	23.67	0.729	0.232	0.0101	20.45	0.732	0.248	0.0132
	DDRM	22.15	0.701	0.209	0.0099	18.52	0.713	0.254	0.0194
	RED-diff	14.57	0.578	0.586	0.0451	13.98	0.628	0.367	0.0437
Inpainting (Random)	DDiff (ours)	**33.08**	**0.877**	**0.050**	**0.0205**	**28.39**	**0.758**	**0.136**	**0.0241**
	DAPS	30.76	0.801	0.156	0.0293	27.32	0.725	0.189	0.0788
	DMPlug	31.65	0.852	0.137	0.0290	26.09	0.740	0.245	0.0316
	DPS	30.79	0.807	0.083	0.0217	27.31	0.737	0.235	0.0980
	DiffPIR	18.53	0.362	0.622	0.0264	15.82	0.191	0.842	0.1080
	DCDP	25.67	0.757	0.224	0.0398	20.40	0.723	0.253	0.1001
Gaussian Deblurring	DDiff (ours)	28.87	**0.800**	**0.119**	**0.0026**	22.29	0.471	0.415	**0.0046**
	DAPS	**29.63**	0.789	0.177	0.0027	**25.90**	**0.658**	**0.269**	0.0084
	DMPlug	22.98	0.537	0.288	0.0036	14.82	0.188	0.680	0.0147
	DPS	27.77	0.704	0.140	0.0029	21.07	0.528	0.392	0.0073
	DiffPIR	26.16	0.624	0.297	0.0031	21.64	0.393	0.497	0.0093
	DCDP	16.75	0.173	0.701	0.0141	16.06	0.183	0.674	0.0195
	DDRM	24.87	0.725	0.246	0.0052	21.14	0.457	0.464	0.0113
	RED-diff	12.18	0.149	1.232	0.0348	12.22	0.128	0.613	0.0256
Motion Deblurring	DDiff (ours)	28.24	0.785	**0.129**	**0.0058**	24.16	0.585	0.242	**0.0079**
	DAPS	**29.17**	**0.797**	0.186	0.0059	**26.61**	**0.710**	**0.241**	0.0085
	DMPlug	21.95	0.512	0.304	0.0076	14.81	0.170	0.696	0.0199
	DPS	27.93	0.714	0.130	0.0061	23.36	0.611	0.321	0.0082
	DiffPIR	22.01	0.327	0.499	0.0074	18.93	0.248	0.586	0.0084
	DCDP	9.536	0.039	0.855	0.0547	9.491	0.066	0.771	0.0511
Phase Retrieval	DDiff (ours)	**29.94**	**0.816**	**0.120**	**0.0040**	18.54	**0.494**	**0.262**	**0.0059**
	DAPS	29.60	0.768	0.182	0.0049	**20.23**	0.449	0.397	0.0085
	DPS	22.24	0.540	0.307	0.0514	16.03	0.396	0.444	0.1040
	DiffPIR	10.04	0.036	0.783	0.1811	9.61	0.021	0.794	0.2410
	DCDP	15.20	0.420	0.616	0.1060	11.63	0.201	0.700	0.1220
	RED-diff	14.88	0.386	0.656	0.1721	13.89	0.266	0.639	0.1121
Nonlinear Deblurring	DDiff (ours)	**31.48**	**0.873**	**0.120**	**0.0027**	**29.68**	**0.805**	**0.207**	**0.0035**
	DAPS	28.45	0.764	0.188	0.0042	27.28	0.718	0.213	0.0048
	DMPlug	27.17	0.791	0.187	0.0051	22.99	0.603	0.366	0.0104
	DPS	25.39	0.643	0.258	0.0095	28.42	0.691	0.271	0.0053
	DiffPIR	19.79	0.331	0.583	0.0273	22.13	0.459	0.435	0.0167
	DCDP	28.87	0.852	0.177	0.0038	25.22	0.700	0.299	0.0105
	RED-diff	29.89	0.783	0.185	0.0040	28.07	0.624	0.306	0.0051
High Dynamic Range	DDiff (ours)	26.05	**0.873**	**0.129**	**0.0459**	**26.50**	0.800	**0.108**	**0.0541**
	DAPS	**27.39**	0.846	0.163	0.0505	26.10	**0.825**	0.171	0.0717
	DPS	25.79	0.793	0.165	0.0734	22.72	0.721	0.273	0.1951
	DiffPIR	17.69	0.645	0.296	0.1292	18.23	0.637	0.289	0.2105
	RED-diff	21.28	0.431	0.359	0.0921	21.02	0.567	0.479	0.1322

among the baselines. For latent diffusion adaptation of our method, we compare against LatentDAPS [43], PSLD [30], and ReSample [32]. Note that PSLD cannot handle nonlinear inverse tasks.

4.2. Main Results

Quantitative evaluation for the linear and nonlinear tasks on FFHQ and ImageNet datasets are shown in Table 1. Our method outperforms the baselines on the vast majority of the tasks, especially in terms of perceptual similarity and residual error. This is further demonstrated in Fig. 2, where we show a qualitative comparison between the baselines and our method. Overall, DDiff reconstructs finer details with fewer visual artifacts. Additional qualitative examples and hyperparameter details are provided in the supplements.

Moreover, DDiff exhibits significantly increased robust-



(a) Gaussian deblurring

(b) Phase retrieval

Figure 3. **Effect of measurement noise level.** DDiff demonstrates greater robustness as noise increases. This evaluation uses 10 FFHQ validation images.

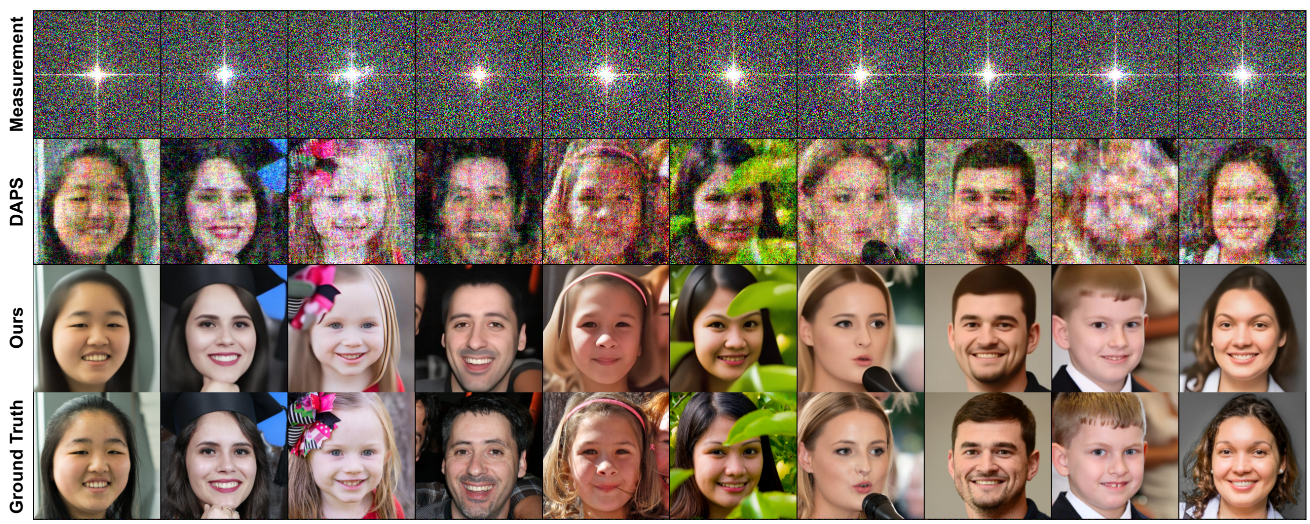


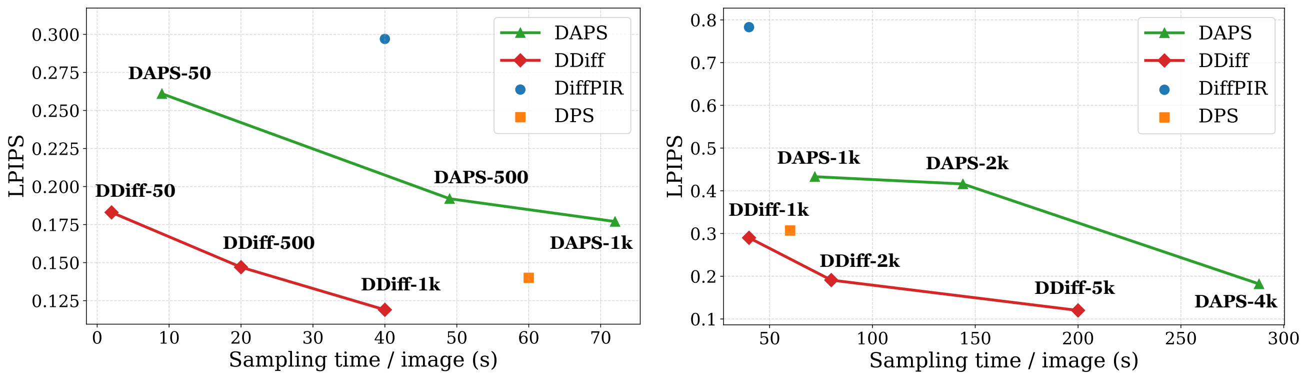
Figure 4. **Qualitative results at high measurement noise level.** We compare our method (DDiff) with DAPS [43] on the phase retrieval task at $\sigma = 0.3$, using 10 randomly selected validation images. Despite the severe measurement noise, DDiff successfully recovers coherent global structures and facial semantics, whereas DAPS reconstructions exhibit strong corruption and noise artifacts.

ness to higher measurement noise. As shown in Fig. 3, although PSNR may be slightly lower than DAPS in the very low measurement noise region (approximately $\sigma < 0.05$), PSNR degrades less rapidly as σ increases. LPIPS is generally lower for DDiff than that of DAPS in all levels of noise. LPIPS follows a similar trend, especially on the phase retrieval task, where DDiff divides LPIPS by 3 in the high-noise region ($\sigma = 0.3$). To visualize the reconstruction quality of DDiff compared to DAPS at the extreme case of measurement noise level ($\sigma = 0.3$), refer to Fig. 4. These results suggest that DDiff could be more appropriate for solving challenging inverse problems where measurement noise is prominent, such as low-dose CT reconstruction.

4.3. Ablation Studies

Evaluation of time efficiency and quality of samples. One of the most limiting factors of sampling speed for diffusion-based methods is the number of neural function evaluations (NFEs). It measures the number of score model evaluations performed during inference. Therefore, we assess the average sampling time and sample quality of our method and of existing baselines as a function of the number of NFEs in Fig. 5. For the same number of NFEs, DDiff achieves better perceptual quality and faster sampling speed compared to DAPS. The enhanced computational efficiency is a consequence of two key factors: reduced backpropagation requirements for gradient calculations and the absence of supplementary MCMC procedures within sampling iterations, both of which are present in the DAPS framework.

Effect of dual variable and noising step. To illustrate the importance of dual variable $\mathbf{u}$ and the additional noising step (Eq. 12) in our algorithm, we conduct an experiment to compare DDiff (with noise, with $\mathbf{u}$) to three other vari-



(a) Gaussian deblurring

(b) Phase retrieval

Figure 5. **Evaluation of time efficiency and quality of samples.** The y-axis shows LPIPS value and the x-axis shows the time (in sec.) taken to generate one sample image on a GeForce RTX 2080 Ti 12GB GPU. The number after the method name (500, 2k, etc.) indicates the NFEs. This evaluation uses 100 FFHQ validation images.

Table 2. **Ablating the dual variable and noising step.** We compare DDiff (with noise, with $\mathbf{u}$) to other variants, including Diff-PnP-HQS (no noise, no $\mathbf{u}$), Diff-PnP-ADMM (no noise, with $\mathbf{u}$), and DDiff-HQS (with noise, no $\mathbf{u}$). Full algorithmic descriptions of each variant are provided in Appendix F. All methods are compared in a fair manner by using the same DDIM noise schedule. This evaluation uses 10 FFHQ validation images and reports the average metric value. The best and second-best results are distinguished by **bold** and underlined marks, respectively. DDiff, with the dual variable and noising step, significantly outperforms other methods.

Method	Super Resolution 4×				Inpainting (Box)				Inpainting (Random)				Gaussian Deblurring			
	PSNR ↑	SSIM ↑	LPIPS ↓	Res. ↓	PSNR ↑	SSIM ↑	LPIPS ↓	Res. ↓	PSNR ↑	SSIM ↑	LPIPS ↓	Res. ↓	PSNR ↑	SSIM ↑	LPIPS ↓	Res. ↓
DDiff (ours)	**30.10**	**0.821**	**0.199**	**0.0034**	**26.01**	**0.855**	**0.087**	**0.0032**	**33.21**	**0.883**	**0.043**	**0.0028**	**29.08**	**0.801**	**0.112**	**0.0025**
DDiff-HQS	25.45	0.558	0.369	0.0057	19.01	0.566	0.221	0.0091	15.37	0.254	0.790	0.0167	26.32	0.642	0.267	0.0031
Diff-PnP-ADMM	13.79	0.434	0.572	0.0304	12.72	0.546	0.486	0.0298	16.79	0.540	0.402	0.0087	6.18	0.077	0.820	0.2552
Diff-PnP-HQS	14.04	0.447	0.535	0.0287	12.61	0.557	0.476	0.0271	15.85	0.513	0.431	0.0168	5.82	0.036	0.858	0.2812

Method	Motion Deblurring				Phase Retrieval				Nonlinear Deblurring				High Dynamic Range			
	PSNR ↑	SSIM ↑	LPIPS ↓	Res. ↓	PSNR ↑	SSIM ↑	LPIPS ↓	Res. ↓	PSNR ↑	SSIM ↑	LPIPS ↓	Res. ↓	PSNR ↑	SSIM ↑	LPIPS ↓	Res. ↓
DDiff (ours)	**28.14**	**0.779**	**0.112**	**0.0026**	**30.58**	**0.834**	**0.096**	**0.0028**	**31.25**	**0.867**	**0.111**	**0.0033**	**26.95**	**0.877**	**0.109**	**0.0072**
DDiff-HQS	21.75	0.396	0.397	0.0036	12.86	0.114	0.679	0.0057	13.51	0.186	0.543	0.1252	17.76	0.672	0.284	0.1260
Diff-PnP-ADMM	6.35	0.089	0.810	0.1840	12.84	0.266	0.637	0.0081	5.23	0.007	0.830	0.2463	7.85	0.126	0.875	0.4797
Diff-PnP-HQS	6.39	0.091	0.813	0.1752	13.63	0.287	0.597	0.0077	6.02	0.086	0.827	0.2537	7.56	0.098	0.885	0.5287

ants: Diff-PnP-HQS (no noise, no $\mathbf{u}$), Diff-PnP-ADMM (no noise, with $\mathbf{u}$), and DDiff-HQS (with noise, no $\mathbf{u}$) in Table 2. Our analysis indicates that the dual variable implementation alone decreases performance, as evidenced by Diff-PnP-ADMM's inferior metrics compared to Diff-PnP-HQS. Without the noising step, the dual variable introduces high-frequency artifacts that compromise the diffusion model's efficacy. However, when incorporated alongside the noising step, the dual variable yields a substantial enhancement in performance, as evidenced by the superior quantitative metrics achieved by DDiff compared to DDiff-HQS. Our experimental results confirm that both components—dual variable and noising step—must be implemented together, as demonstrated by the significant performance improvement observed when transitioning from Diff-PnP-HQS to DDiff.

5. Conclusion

In this work, we introduce DDiff, a dual-ascent framework for solving diffusion model-based inverse problems that achieves better quantitative and qualitative reconstruction quality than the state of the art, especially at high measurement noise levels. By jointly optimizing in the primal and dual spaces and carefully adapting the ADMM $\mathbf{z}$-update, our approach effectively leverages pretrained diffusion priors to improve perceptual quality, measurement consistency, and runtime. DDiff is highly robust to noise and computationally efficient, making it practical for solving large-scale, complex inverse problems. Future work includes extending the framework to higher-dimensional data, such as 3D or video representations, which presents additional challenges and opportunities.

References

[1] Kazunori Akiyama, Antxon Alberdi, Walter Alef, Keiichi Asada, Rebecca Azulay, Anne-Kathrin Baczko, David Ball, Mislav Baloković, John Barrett, Dan Bintley, et al. First m87 event horizon telescope results. iv. imaging the central supermassive black hole. *The Astrophysical Journal Letters*, 875(1):L4, 2019. 1

[2] Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982. 2

[3] Amir Beck and Marc Teboulle. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *IEEE transactions on image processing*, 18(11):2419–2434, 2009. 1

[4] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011. 1, 2

[5] Stanley H Chan, Xiran Wang, and Omar A Elgendy. Plug-and-play admm for image restoration: Fixed-point convergence and applications. *IEEE Transactions on Computational Imaging*, 3(1):84–98, 2016. 2

[6] Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. Ilvr: Conditioning method for denoising diffusion probabilistic models. *arXiv preprint arXiv:2108.02938*, 2021. 1

[7] Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. *arXiv preprint arXiv:2209.14687*, 2022. 1, 2, 4, 5, 9, 10

[8] Hyungjin Chung, Byeongsu Sim, Dohoon Ryu, and Jong Chul Ye. Improving diffusion models for inverse problems using manifold constraints. *Advances in Neural Information Processing Systems*, 35:25683–25696, 2022. 3

[9] Giannis Daras, Hyungjin Chung, Chieh-Hsin Lai, Yuki Mitsufuji, Jong Chul Ye, Peyman Milanfar, Alexandros G Dimakis, and Mauricio Delbracio. A survey on diffusion models for inverse problems. *arXiv preprint arXiv:2410.00083*, 2024. 1, 2

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 4

[11] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, 2021. 4

[12] David L Donoho. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006. 1

[13] Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011. 3

[14] Alisia Fadini, Minhuan Li, Airlie J McCoy, Thomas C Terwilliger, Randy J Read, Doeke Hekstra, and Mohammed AlQuraishi. Alphafold as a prior: Experimental structure determination conditioned on a pretrained neural network. *bioRxiv*, pages 2025–02, 2025. 1

[15] Berthy T Feng, Jamie Smith, Michael Rubinstein, Huiwen Chang, Katherine L Bouman, and William T Freeman. Score-based diffusion models as principled priors for inverse imaging. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10520–10531, 2023. 1

[16] Donald Geman and Chengda Yang. Nonlinear image recovery with half-quadratic regularization. *IEEE transactions on Image Processing*, 4(7):932–946, 1995. 4

[17] Ulrich G Haussmann and Etienne Pardoux. Time reversal of diffusions. *The Annals of Probability*, pages 1188–1205, 1986. 2

[18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arxiv:2006.11239*, 2020. 2

[19] Ajil Jalal, Marius Arvinte, Giannis Daras, Eric Price, Alexandros G Dimakis, and Jon Tamir. Robust compressed sensing mri with deep generative priors. *Advances in Neural Information Processing Systems*, 34:14938–14954, 2021. 1, 2

[20] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(12):4217–4228, 2021. 4

[21] Bahjat Kawar, Gregory Vaksman, and Michael Elad. Snips: Solving noisy inverse problems stochastically. *Advances in Neural Information Processing Systems*, 34:21757–21769, 2021. 1

[22] Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. *Advances in Neural Information Processing Systems*, 35:23593–23606, 2022. 1, 4, 10

[23] Axel Levy, Eric R Chan, Sara Fridovich-Keil, Frédéric Poitevin, Ellen D Zhong, and Gordon Wetzstein. Solving inverse problems in protein space using diffusion-based priors. *arXiv preprint arXiv:2406.04239*, 2024. 1

[24] Xiang Li, Soo Min Kwon, Ismail R Alkhouri, Saiprasad Ravishanka, and Qing Qu. Decoupled data consistency with diffusion purification for image restoration. *arXiv preprint arXiv:2403.06054*, 2024. 4, 7, 10

[25] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1833–1844, 2021. 1

[26] Sai Advaith Maddipatla, Nadav Bojan Sellam, Sanketh Vedula, Ailie Marx, and Alex Bronstein. Generative modeling of protein ensembles guided by crystallographic electron densities. *arXiv preprint arXiv:2412.13223*, 2024. 1

[27] Morteza Mardani, Jiaming Song, Jan Kautz, and Arash Vahdat. A variational perspective on solving inverse problems with diffusion models. *arXiv preprint arXiv:2305.04391*, 2023. 1, 4, 10

[28] Neal Parikh, Stephen Boyd, et al. Proximal algorithms. *Foundations and trends® in Optimization*, 1(3):127–239, 2014. 3

[29] Robin Rombach, A. Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *2022 IEEE/CVF Conference*

on Computer Vision and Pattern Recognition (CVPR)*, pages 10674–10685, 2021. 4, 6

[30] Litu Rout, Negin Raoof, Giannis Daras, Constantine Caramanis, Alex Dimakis, and Sanjay Shakkottai. Solving linear inverse problems provably via posterior sampling with latent diffusion models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 6, 10

[31] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992. 1

[32] Bowen Song, Soo Min Kwon, Zecheng Zhang, Xinyu Hu, Qing Qu, and Liyue Shen. Solving inverse problems with latent diffusion models via hard data consistency. In *The Twelfth International Conference on Learning Representations*, 2024. 4, 6, 10

[33] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2

[34] Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *International Conference on Learning Representations*, 2023. 1

[35] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 1

[36] Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. *Advances in neural information processing systems*, 34:1415–1428, 2021. 2

[37] Yang Song, Liyue Shen, Lei Xing, and Stefano Ermon. Solving inverse problems in medical imaging with score-based generative models. *arXiv preprint arXiv:2111.08005*, 2021. 1

[38] Phong Tran, Anh Tran, Quynh Phung, and Minh Hoai. Explore image deblurring via encoded blur kernel space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 9

[39] Singanallur V Venkatakrishnan, Charles A Bouman, and Brendt Wohlberg. Plug-and-play priors for model based reconstruction. In *2013 IEEE global conference on signal and information processing*, pages 945–948. IEEE, 2013. 1, 2, 7

[40] Hengkang Wang, Xu Zhang, Taihui Li, Yuxiang Wan, Tiancong Chen, and Ju Sun. DMPlug: A plug-in method for solving inverse problems with diffusion models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 4, 10

[41] Yinhuai Wang, Jiwen Yu, and Jian Zhang. Zero-shot image restoration using denoising diffusion null-space model. *arXiv preprint arXiv:2212.00490*, 2022. 1

[42] Zihui Wu, Yu Sun, Yifan Chen, Bingliang Zhang, Yisong Yue, and Katherine Bouman. Principled probabilistic imaging using diffusion models as plug-and-play priors. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 4

[43] Bingliang Zhang, Wenda Chu, Julius Berner, Chenlin Meng, Anima Anandkumar, and Yang Song. Improving diffusion inverse problem solving with decoupled noise annealing. *arXiv preprint arXiv:2407.01521*, 2024. 1, 2, 4, 5, 6, 7, 8, 9, 10

[44] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE transactions on image processing*, 26(7):3142–3155, 2017. 1

[45] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. 4

[46] Yuanzhi Zhu, Kai Zhang, Jingyun Liang, Jiezhang Cao, Bihan Wen, Radu Timofte, and Luc Van Gool. Denoising diffusion models for plug-and-play image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1219–1229, 2023. 4, 10