Diffusion Models for Image Generation and Solving Inverse Problems

Stephone Christian

Note: For full subset of output images, please refer to the submitted pdf containing the code.

Abstract—Diffusion models have emerged as a leading class of generative models capable of producing high-quality images through iterative denoising. In this write up, we explore the variance-preserving (VP) formulation of denoising diffusion probabilistic models (DDPM) for unconditional image generation and solving linear inverse problems. We first explore the theoretical foundations by deriving key relationships in the forward and reverse diffusion processes. We then implement single-step image denoising and iterative DDPM sampling using a pretrained score-predicting network on the FFHQ-256 dataset. For image inpainting and deconvolution problems, we implement and compare three posterior sampling approaches: SDEdit, Score-based Annealed Langevin Dynamics (ScoreALD), and Diffusion Posterior Sampling (DPS). Our analysis shows that DPS achieves the best quantitative performance across both tasks, while SDEdit provides a simpler but less precise baseline. We provide qualitative and quantitative comparisons (PSNR, LPIPS) and discuss the trade-offs between these methods.

Index Terms—Diffusion Models, Inverse Problems, Image Inpainting, Deconvolution, Posterior Sampling

1 Introduction

Generative modeling has seen tremendous progress recently, with diffusion models emerging as a one of the dominant methods for image synthesis [1], [2], [3]. Diffusion models define a forward process that gradually adds Gaussian noise to the data and learn to reverse this process through iterative denoising. This formulation offers more stable training, better mode coverage, and strong sample quality.

More than unconditional generation, diffusion models can serve as powerful priors for solving inverse problems in the field of computational imaging and signal processing. An inverse problem is defined as follows: given a forward measurement model, $\mathbf{y} = \mathcal{A}(\mathbf{x}_0) + \mathbf{n}$, the objective is to recover the original image $\mathbf{x}_0$ from the corrupted observations, $\mathbf{y}$. Classical approaches often rely on hand-crafted priors (i.e. TV norm, sparsity, etc.), but diffusion models provide a learned, expressive prior through the score function $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$. Diffusion models are attractive for inverse problems as they often do not require task specific retraining as they are single pretrained models that can be applied to many different inverse problems.

In this write up, we use a pretrained diffusion model on the FFHQ-256 dataset [4] to:

1) Demonstrate single-step image denoising using score-based prediction,
2) Experiment with unconditional image generation using the DDPM reverse process [1],
3) Solve image inpainting and deconvolution invervse problems using SDEdit [5], ScoreALD [6], and DPS [7].

• Department of Electrical Engineering, Stanford University.

2 Related Work

2.1 Diffusion Models

Sohl-Dickstein et al. [8] first introduced Diffusion Probabilistic Models (DPM). The Authors proposed learning the reverse of a gradual noising process as a solution for training a generative model. Ho et al. [1] advanced this framework and approach with Denoising Diffusion Probabilistic Models (DDPM). DDPM showcased how effective a simple noise-prediction objective could be for image generation. Song et al. [2] showed parallels between score-based generative models and stochastic differential equations (SDEs). More precisely, the discrete time DDPM formulation was shown to be equivalent to a continuous time SDE. Dhariwal and Nichol [3] showed that diffusion models can outperform GANs on image synthesis when coupled with guidance.

2.2 Inverse Problems with Diffusion Priors

There exists many approached for using Diffusion Models as priors for inverse problems. SDEdit, ScoreALD, and DPS are three such examples that attempt to incorporate measurement consistency in the reverse sampling process.

SDEdit [5] is the simplest approach we analyse. SDEdit adds noise to the measurement up to an intermediate diffusion timestep, and then applies the standard reverse process. This implicitly conditions on the measurement but provides no explicit likelihood guidance. In practice this method requires tuning the noise level added to balance fidelity and realism.

ScoreALD [6] incorporates the likelihood gradient $\nabla_{\mathbf{x}} \log p(\mathbf{y}|\mathbf{x})$ into the Langevin dynamics at each reverse step, attenuated by an annealed step size. This technique encourages measurement consistency throughout the reverse process.

DPS [7] builds upon ScoreALD and makes a closer approximation of the posterior score $\nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t)$ by using

the estimated clean image $\hat{\mathbf{x}}_0$ at each step. Unlike ScoreALD, DPS normalizes the likelihood gradient, and uses the estimate clean image $\hat{\mathbf{x}}_0$ in the image formation model. These combined additions allow DPS to handle nonlinear forward models, and to also have more stable conditioning throughout the reverse process.

3 Methods

3.1 Variance-Preserving Diffusion Framework

We use the variance-preserving (VP) formulation detailed in the DDPM [1] framework. The forward process adds Gaussian noise according to a schedule $\{\beta_t\}_{t=1}^T$ with $T = 1000$ timesteps and a linear schedule:

$$\mathbf{x}_t = \sqrt{1-\beta_t}\,\mathbf{x}_{t-1} + \sqrt{\beta_t}\,\mathbf{z}_{t-1}, \quad \mathbf{z}_{t-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{1}$$

Using the following definitions, $\alpha_t = 1-\beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$, the forward process has a closed form expression (see Appendix A):

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\mathbf{z}, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{2}$$

The pretrained model outputs the score function $\mathbf{s}_\theta(\mathbf{x}_t, t) \approx \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$, which we use to calculate a clean image estimate using Tweedie's formula [9]:

$$\hat{\mathbf{x}}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}\big(\mathbf{x}_t + (1-\bar{\alpha}_t)\,\mathbf{s}_\theta(\mathbf{x}_t, t)\big). \tag{3}$$

3.2 Single-Step Denoising

Given a noisy observation $\mathbf{x}_t$ at timestep t, single-step denoising uses Eq. (3) to estimate $\hat{\mathbf{x}}_0$. We can think of this as using the pretrained score network as a Noise Predictor Network (NPN) for gaussian noise.

3.3 Unconditional Generation via DDPM

Starting from $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, the DDPM reverse process samples images iteratively:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}(\mathbf{x}_t + (1-\alpha_t)\,\mathbf{s}_\theta(\mathbf{x}_t, t)) + \sigma_t\,\mathbf{z}, \tag{4}$$

$\sigma_t^2 = \tilde{\beta}_t = \beta_t(1-\bar{\alpha}_{t-1})/(1-\bar{\alpha}_t)$ is the posterior variance and $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. We show that this formulation is the same as the posterior mean formulation used in the original DDPM paper in Appendix A. We also highlight the connection between the score-predicting network $\mathbf{s}_\theta$ and the noise-predicting network $\boldsymbol{\epsilon}_\theta$ from Algorithm 2 of [1] in Appendix A.

3.4 Inverse Problems

The linear inverse problems are of the form $\mathbf{y} = \mathcal{A}(\mathbf{x}_0) + \mathbf{n}$, where $\mathcal{A}$ is a known linear process and $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma_y^2\mathbf{I})$. We look into two problems:

- Inpainting: $\mathcal{A} = \mathbf{M}$ is a binary mask operator. We test a 50×50 box mask (at position $(150, 100)$) and a random mask removing 50% of pixels.
- Deconvolution: $\mathcal{A}$ is convolution with a Gaussian blur kernel (size 61×61, $\sigma_{\text{kernel}} = 3.0$).

We set the measurement noise to $\sigma_y = 0.05$ for all experiments.

TABLE 1: Single-step denoising metrics on FFHQ test image

Timestep t	PSNR (dB) ↑	LPIPS ↓
50	35.40	0.056
100	32.55	0.089
200	29.47	0.139

3.4.1 SDEdit

SDEdit [5] starts from the measurement $\mathbf{y}$, and noises it to an intermediate timestep t_{start}:

$$\mathbf{x}_{t_{\text{start}}} = \sqrt{\bar{\alpha}_{t_{\text{start}}}}\,\mathbf{y} + \sqrt{1-\bar{\alpha}_{t_{\text{start}}}}\,\mathbf{z}, \tag{5}$$

SDEdit then applies the standard DDPM reverse process from t_{start} down to $t = 0$ without additional conditioning. (intermediate timesteps) $t_{\text{start}} \in \{300, 500, 700\}$.

3.4.2 ScoreALD

ScoreALD [6] adds a likelihood gradient correction at each reverse step. After computing the standard DDPM posterior mean $\mathbf{x}_{t-1}$, we apply:

$$\mathbf{x}_{t-1} \leftarrow \mathbf{x}_{t-1} - \frac{1}{2(\sigma^2+\gamma_t^2)}\nabla_{\mathbf{x}_t}\|\mathcal{A}(\mathbf{x}_t) - \mathbf{y}\|_2^2, \tag{6}$$

where σ^2 is the measurement noise variance and γ_t^2 is an annealing term. We use an annealed schedule for γ_t to approximate the combined $(\sigma^2+\gamma^2)$ term, which balances the influence of the likelihood gradient across diffusion timesteps. For inpainting, γ_t is linearly spaced from 15 to 20 over the 1000 timesteps with an additional scaling factor of 0.06; for deconvolution, γ_t is linearly spaced from 10 to 15.

3.4.3 DPS

DPS [7] approximates the posterior score $\nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t)$ using the denoised estimate $\hat{\mathbf{x}}_0$. We update $\mathbf{x_t}$ as follows:

$$\mathbf{x}_t \leftarrow \mathbf{x}_t - \zeta \cdot \frac{\nabla_{\mathbf{x}_t}\|\mathcal{A}(\hat{\mathbf{x}}_0) - \mathbf{y}\|_2^2}{\|\nabla_{\mathbf{x}_t}\|\mathcal{A}(\hat{\mathbf{x}}_0) - \mathbf{y}\|_2^2\| + \epsilon}, \tag{7}$$

where ζ is scale factor and $\epsilon = 10^{-8}$ prevents division by zero. The gradient normalization helps stabilizes the conditioning. We use $\zeta = 1.0$ for box-mask inpainting, $\zeta = 0.1$ for random-mask inpainting, and $\zeta = 0.3$ for deconvolution. Intuitively, DPS using the estimated clean image $\hat{\mathbf{x}}_0$ in the image formation model makes more sense than using a noisy iterate. We should expect an improvement when benchmarking against ScoreALD.

4 Experiments

All experiments use a pretrained score-predicting U-Net [3] trained on the FFHQ-256 dataset [4] with $T = 1000$ diffusion steps and a linear β schedule. We evaluate reconstruction quality using PSNR and LPIPS [10]. Images are normalized to a range of $[-1, 1]$.

4.1 Single-Step Denoising

We apply the forward process using timesteps $t \in \{50, 100, 200\}$ and calculate the clean image $\hat{\mathbf{x}}_0$ using Eq. (3).

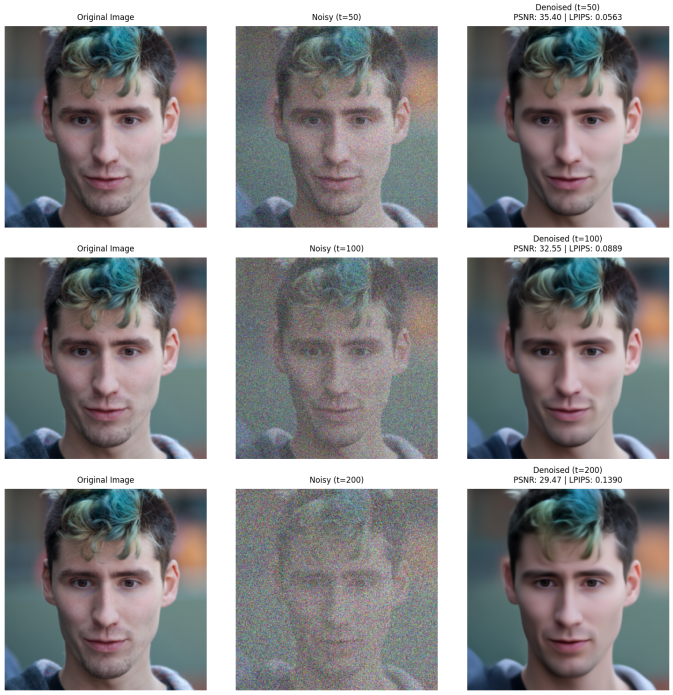

Fig. 1: Single-step denoising results at different noise timesteps.

Fig. 2: Generated images using the pretrained DDPM with $T = 1000$.

4.2 Unconditional Image Generation

We generate images by through the following routine: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Generated image shapes are 256×256. We run the full 1000 steps of the DDPM reverse process from an image of standard gaussian noise.

Overall, the generated images look plausibly realistic, however there are some artifacts present in the images.

4.3 Posterior Sampling for Inverse Problems

We evaluate SDEdit, ScoreALD, and DPS on inpainting and deconvolution. Figure 3 and Table 2 show SDEdit results across $t_{\text{start}} \in \{300, 500, 700\}$. At higher intermediate timesteps the images appear more realistic but exhibit reduced fidelity to the measurement; at lower t_{start} fidelity improves but artifacts can persist. Figure 4 and Table 3 summarize ScoreALD, which incorporates annealed likelihood gradients at each step and improves upon SDEdit. Figure 5 and Table 4 show DPS results, which achieve substantially higher PSNR and lower LPIPS than both baselines.

TABLE 2: SDEdit reconstruction metrics

Task	t_{start}	PSNR (dB) ↑	LPIPS ↓
Inpainting	300	23.40	0.125
	500	20.76	0.209
	700	16.63	0.295
Deconvolution	300	23.48	0.170
	500	19.96	0.235
	700	16.08	0.289

TABLE 3: ScoreALD reconstruction metrics

Task	PSNR (dB) ↑	LPIPS ↓
Inpainting (box)	22.43	0.133
Deconvolution	26.60	0.077

TABLE 4: DPS reconstruction metrics

Task	PSNR (dB) ↑	LPIPS ↓
Inpainting (box)	34.23	0.025
Deconvolution	28.82	0.060

TABLE 5: Comparison of posterior sampling methods on inpainting (box mask) and deconvolution

	Inpainting		Deconvolution	
Method	PSNR ↑	LPIPS ↓	PSNR ↑	LPIPS ↓
SDEdit ($t_s = 500$)	20.76	0.209	19.96	0.235
ScoreALD	22.43	0.133	26.60	0.077
DPS	34.23	0.025	28.82	0.060

4.4 Comparison of Methods

Table 5 compares the three methods on inpainting (box mask) and deconvolution. DPS achieves the best performance by a wide margin: +11.6 dB PSNR over ScoreALD and +13.5 dB over SDEdit on inpainting, with LPIPS reduced by roughly 5×. This gap arises from several factors. SDEdit provides no explicit measurement consistency as it only noises and denoises the measurement. ScoreALD improves on this by injecting annealed likelihood gradients, but it operates on the noisy iterate $\mathbf{x}_t$, which is not an ideal measurement to feed into the image formation model. DPS instead computes the likelihood on the denoised estimate $\hat{\mathbf{x}}_0$, which creates more informative gradients throughout the reverse process.DPS formulation also normalizes these gradients and therefore exhibits more stable conditioning during the reverse process. DPS unsurprisingly consistently delivers the highest quality reconstructions.

5 Discussion and Conclusion

We explored three approaches for using pretrained diffusion models to solve inverse problems. SDEdit provides a simple baseline by noising and denoising measurements, but lacks explicit measurement consistency. ScoreALD incorporates annealed likelihood gradients for improved fidelity, while DPS further enhances performance through normalized gradient

Fig. 3: SDEdit results for inpainting with intermediate timestep 500.

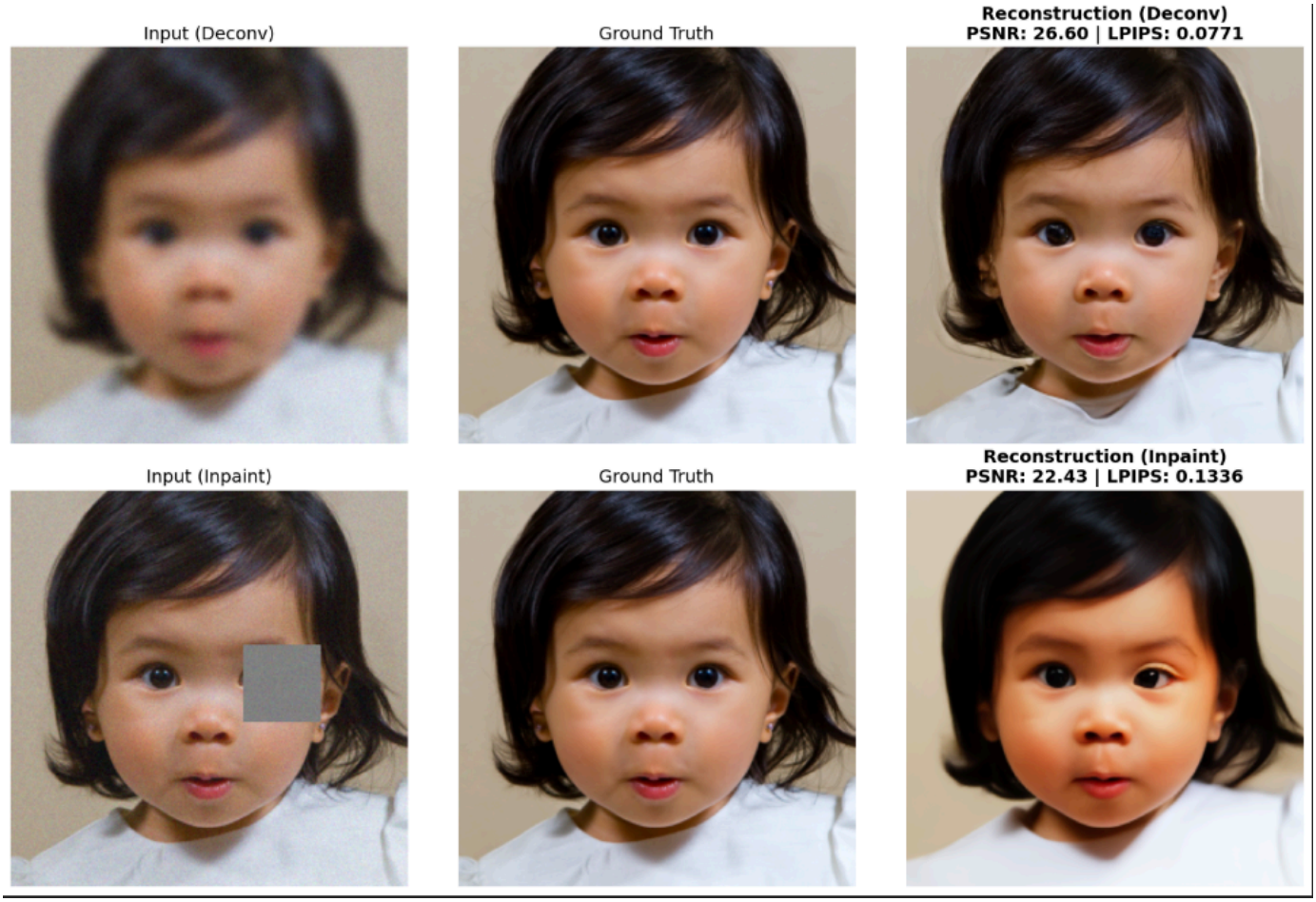

Fig. 4: ScoreALD results for inpainting with box mask and deconvolution at noise level 500 timesteps.

updates on the denoised estimate. Our experiments on image inpainting and deconvolution show that DPS consistently achieves the highest PSNR and lowest LPIPS across all tasks, confirming the benefits of its posterior sampling formulation.

References

[1] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in Advances in Neural Information Processing Systems (NeurIPS), vol. 33, 2020, pp. 6840–6851.

[2] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," International Conference on Learning Representations (ICLR), 2021.

[3] P. Dhariwal and A. Nichol, "Diffusion models beat GANs on image synthesis," Advances in Neural Information Processing Systems (NeurIPS), vol. 34, pp. 8780–8794, 2021.

[4] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," IEEE/CVF Con-

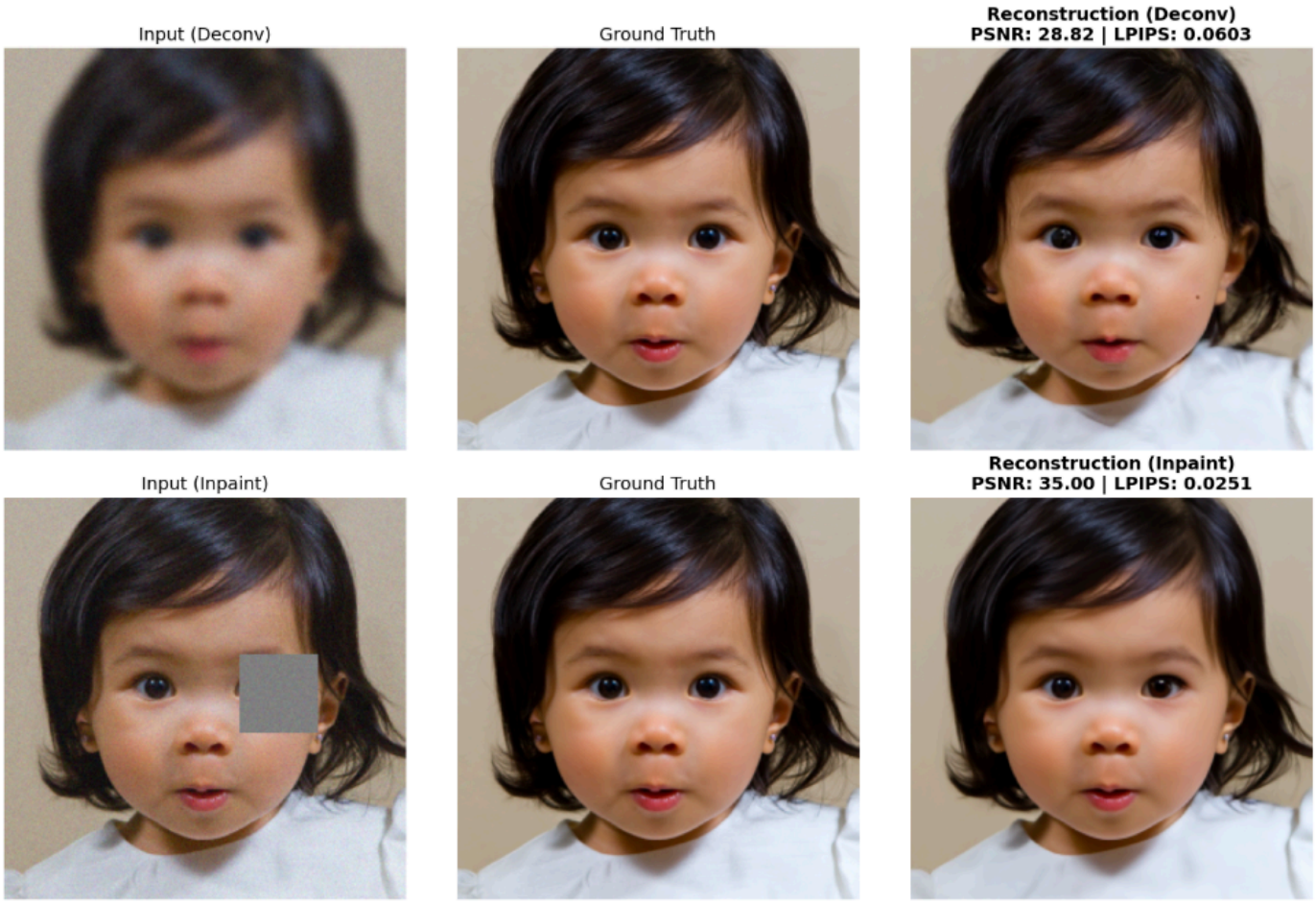

Fig. 5: DPS results with $T = 1000$ timesteps. Inpainting with box mask ($\zeta = 1.0$) and deconvolution with Gaussian blur ($\zeta = 0.3$).

ference on Computer Vision and Pattern Recognition (CVPR), pp. 4401–4410, 2019.
[5] C. Meng, Y. He, Y. Song, J. Song, J. Wu, J.-Y. Zhu, and S. Ermon, "SDEdit: Guided image synthesis and editing with stochastic differential equations," International Conference on Learning Representations (ICLR), 2022.
[6] A. Jalal, M. Arvinte, G. Daras, E. Price, A. G. Dimakis, and J. Tamir, "Robust compressed sensing MRI with deep generative priors," Advances in Neural Information Processing Systems (NeurIPS), 2021.
[7] H. Chung, J. Kim, M. T. McCann, M. L. Klasky, and J. C. Ye, "Diffusion posterior sampling for general noisy inverse problems," in International Conference on Learning Representations (ICLR), 2023.
[8] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in International Conference on Machine Learning (ICML), 2015, pp. 2256–2265.
[9] B. Efron, "Tweedie's formula and selection bias," Journal of the American Statistical Association, vol. 106, no. 496, pp. 1602–1614, 2011.
[10] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 586–595, 2018.

Appendix

1.1.

We prove that the forward process in Eq. (1) admits the closed-form marginal in Eq. (2).

Starting from the single-step forward process with $\alpha_t = 1 - \beta_t$:

$$\mathbf{x}_t = \sqrt{\alpha_t}\,\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\,\mathbf{z}_{t-1}, \quad \mathbf{z}_{t-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{8}$$

Substituting the expression for $\mathbf{x}_{t-1}$ into $\mathbf{x}_t$:

$$\begin{aligned}\mathbf{x}_t &= \sqrt{\alpha_t}\big(\sqrt{\alpha_{t-1}}\,\mathbf{x}_{t-2} + \sqrt{1-\alpha_{t-1}}\,\mathbf{z}_{t-2}\big) + \sqrt{1-\alpha_t}\,\mathbf{z}_{t-1} \\ &= \sqrt{\alpha_t\alpha_{t-1}}\,\mathbf{x}_{t-2} + \sqrt{\alpha_t(1-\alpha_{t-1})}\,\mathbf{z}_{t-2} + \sqrt{1-\alpha_t}\,\mathbf{z}_{t-1}.\end{aligned} \tag{9}$$

Since $\mathbf{z}_{t-2}$ and $\mathbf{z}_{t-1}$ are independent standard Gaussians, the sum of the two noise terms is also Gaussian with zero mean and variance:

$$\alpha_t(1-\alpha_{t-1}) + (1-\alpha_t) = 1 - \alpha_t\alpha_{t-1}. \tag{10}$$

Therefore, we can write:

$$\mathbf{x}_t = \sqrt{\alpha_t\alpha_{t-1}}\,\mathbf{x}_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\,\mathbf{z}', \quad \mathbf{z}' \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{11}$$

Applying this recursion down to $t = 0$ and defining $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$:

$$\boxed{\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\mathbf{z}, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).} \tag{12}$$

1.2.

We prove that the DDPM reverse step

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\big(\mathbf{x}_t + (1-\alpha_t)\,\mathbf{s}_\theta(\mathbf{x}_t, t)\big) \tag{13}$$

is equivalent to the posterior mean parameterization

$$\mathbf{x}_{t-1} = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\,\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)}{1-\bar{\alpha}_t}\,\hat{\mathbf{x}}_0, \tag{14}$$

where $\hat{\mathbf{x}}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}\big(\mathbf{x}_t + (1-\bar{\alpha}_t)\,\mathbf{s}_\theta(\mathbf{x}_t, t)\big)$.

Proof. Substituting the expression for $\hat{\mathbf{x}}_0$ into Eq. (14):

$$\mathbf{x}_{t-1} = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\,\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)}{1-\bar{\alpha}_t}\cdot\frac{1}{\sqrt{\bar{\alpha}_t}}\big(\mathbf{x}_t + (1-\bar{\alpha}_t)\,\mathbf{s}_\theta(\mathbf{x}_t, t)\big). \tag{15}$$

Using $\bar{\alpha}_t = \alpha_t\,\bar{\alpha}_{t-1}$, we simplify the prefactor on the second term:

$$\frac{\sqrt{\bar{\alpha}_{t-1}}}{\sqrt{\bar{\alpha}_t}} = \frac{\sqrt{\bar{\alpha}_{t-1}}}{\sqrt{\alpha_t\,\bar{\alpha}_{t-1}}} = \frac{1}{\sqrt{\alpha_t}}. \tag{16}$$

Substituting back and collecting the coefficient of $\mathbf{x}_t$:

$$\begin{aligned}\text{coeff of } \mathbf{x}_t &= \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} + \frac{(1-\alpha_t)}{(1-\bar{\alpha}_t)\sqrt{\alpha_t}} \\ &= \frac{\alpha_t(1-\bar{\alpha}_{t-1}) + (1-\alpha_t)}{(1-\bar{\alpha}_t)\sqrt{\alpha_t}} \\ &= \frac{\alpha_t - \bar{\alpha}_t + 1 - \alpha_t}{(1-\bar{\alpha}_t)\sqrt{\alpha_t}} = \frac{1-\bar{\alpha}_t}{(1-\bar{\alpha}_t)\sqrt{\alpha_t}} = \frac{1}{\sqrt{\alpha_t}}.\end{aligned} \tag{17}$$

For the coefficient of $\mathbf{s}_\theta(\mathbf{x}_t, t)$:

$$\frac{(1-\alpha_t)(1-\bar{\alpha}_t)}{(1-\bar{\alpha}_t)\sqrt{\alpha_t}} = \frac{1-\alpha_t}{\sqrt{\alpha_t}}. \tag{18}$$

Combining:

$$\boxed{\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\big(\mathbf{x}_t + (1-\alpha_t)\,\mathbf{s}_\theta(\mathbf{x}_t, t)\big).} \quad \square \tag{19}$$

1.3.

We prove the equivalence between the score-predicting and noise-predicting formulations:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\big(\mathbf{x}_t + (1-\alpha_t)\,\mathbf{s}_\theta(\mathbf{x}_t, t)\big) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\,\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right). \tag{20}$$

Proof. From the VP forward process (Eq. (2)):

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{21}$$

The score of the conditional distribution $q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t}\,\mathbf{x}_0,\ (1-\bar{\alpha}_t)\mathbf{I})$ is:

$$\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t|\mathbf{x}_0) = -\frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0}{1-\bar{\alpha}_t} = -\frac{\boldsymbol{\epsilon}}{\sqrt{1-\bar{\alpha}_t}}. \tag{22}$$

By Tweedie's formula [9], the marginal score satisfies:

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) = \mathbb{E}\Big[\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t|\mathbf{x}_0)\,\Big|\,\mathbf{x}_t\Big] = -\frac{\mathbb{E}[\boldsymbol{\epsilon}|\mathbf{x}_t]}{\sqrt{1-\bar{\alpha}_t}}. \tag{23}$$

Since $\mathbf{s}_\theta(\mathbf{x}_t, t) \approx \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$ and $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \approx \mathbb{E}[\boldsymbol{\epsilon}|\mathbf{x}_t]$, we have:

$$\mathbf{s}_\theta(\mathbf{x}_t, t) = -\frac{\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)}{\sqrt{1-\bar{\alpha}_t}}. \tag{24}$$

Substituting Eq. (24) into the score-based reverse step:

$$\begin{aligned}\mathbf{x}_{t-1} &= \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t + (1-\alpha_t)\left(-\frac{\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)}{\sqrt{1-\bar{\alpha}_t}}\right)\right) \\ &= \boxed{\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\,\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right).} \quad \square\end{aligned} \tag{25}$$