Inverse Imaging Tasks with Diffusion Models

Kien Deshpande

Abstract—Diffusion models have emerged as powerful tools for unconditional image generation. This project evaluates the use of diffusion-based image generation to solve inverse imaging problems using a pretrained, score-based, diffusion model. We produce noisy test images using a variance-preserving forward model (inpainting and deconvolution with Gaussian noise). We test diffusion models as priors for solving inverse problems, using three algorithms: SDEdit, ScoreALD, and Diffusion Posterior Sampling (DPS). SDEdit reconstructs a denoised image using the reverse diffusion process. ScoreALD and DPS include unique guidance factors that incorporate measurement information through scaled likelihood gradients in the reverse process. Qualitative and quantitative evaluations using PSNR and LPIPS metrics are conducted to assess reconstruction quality under different noise levels and parameters. The results demonstrate how diffusion models can serve as powerful priors in solving ill-posed imaging problems.

Index Terms—Computational Imaging, Diffusion Models, SDEdit, ScoreALD, Diffusion Posterior Sampling

1 Introduction

Diffusion models are used for artificial image generation given a noisy input, the output image being some interpolation of the training data. Large and diverse training datasets allow diffusion models to produce a wider span of artificial output images. A more useful application of diffusion models is to solve inverse imaging problems to recover a clean output from a corrupted input. An optical image may be corrupted by natural noise sources such as shot noise, read noise, and motion blur. These all have mathematical reprsentations as Poisson noise, Gaussian noise, and convolution with a blur kernel, respectively.

There are non-diffusion based methods that attempt to solve inverse imaging tasks including Wiener Filters, HQS, and ADMM. These methods are less generalizable (than diffusion-based approaches) and unstable in certain conditions. Wiener Filters denoise at the expense of resolution and become more unstable for high spatial frequencies. HQS and ADMM iteratively minimize a loss function with a regularization term to make the problem well-posed and penalize undesirable geometries in the reconstructed image. For example, an L1 regularizer is used when the original image is known to be sparse (astrophotography). These priors are simply and may fail to capture complex patterns found in natural images. Large datasets for diffusion model priors do not face this challenge.

This project will evaluate multiple diffusion-based methods to handle inverse imaging tasks. First, we will evaluate Denoising Diffusion Probabilistic Models (DDPM) [2] to recover clean images after injecting Gaussian noise through a variance-preserving forward model. Second, we will compare three algorithms' (SDEdit, ScoreALD, DPS) performance for inpainting and deconvolution with added Gaussian noise. SDEdit [4] reverses a partially noised image. ScoreALD [3] improves the standard DDPM reverse process by adding a guidance factor that incorporates measurements at the current time step. DPS [1] further improves ScoreALD by computing the expected value of the clean image $\hat{x}_0$ when calculating the measurement error at each time step. These methods will be compared using Peak Signal to Noise Ratio (PSNR) and Learned Perceptual Image Patch Similarity (LPIPS).

The experiment evaluates how well, qualitatively and quantitatively, these algorithms perform in reconstructing corrupted images. Peak Signal-to-Noise Ratio (PSNR) and Learned Perceptual Image Patch Similarity (LPIPS) are the metrics used to measure image reconstruction quality. The results are that DPS outperforms ScoreALD and SDEdit because of its improved guidance factor that exploits Tweedie's Formula. SDEdit reconstruction is vulnerable to the length of the forward noising process; this directly controls the tradeoff between faithful vs. realistic results.

• *K. Deshpande is with the Department of Electrical Engineering, Stanford University, Stanford, CA, 94305.*
E-mail: kiendes@stanford.edu

2 Related Work

This section will briefly discuss the previous work this paper uses for its experiments. Ho et al. [1] introduced DDPMs; a neural network learns the noise or score prediction function associated with the data distribution. Starting from a Gaussian noised image, DDPM applies the learned reverse process and iteratively produces a clean output image.

SDEdit [2] applies a forward process to corrupt the input with Gaussian noise. It then applies the DDPM reverse diffusion process to reconstruct an image consistent with the prior. Reconstruction quality depends on the level of injected noise, which controls the tradeoff between faithfulness and realism (with respect to the input image).

More recent work has explored the use of deep generative models as priors to improve inverse imaging accuracy. Jalal et al. [3] proposed ScoreALD. ScoreALD incorporates measurement information into the diffusion sampling process. This approach samples from a posterior distribution conditioned on observed measurements.

Chung et al. [4] introduced Diffusion Posterior Sampling (DPS), which improves the accuracy of posterior sampling (compared to ScoreALD). DPS computes the likelihood gradient using an estimate of the clean image obtained through Tweedie's formula. This means the forward model (when

Fig. 1. Diffusion denoising example. From left to right: clean image x_0, noisy image x_t at timestep $t = 300$, and the denoised reconstruction.

calculating measurement error) operates on a denoised estimate instead of the noisy image of the current time step.

3 Proposed Method

3.1 Denoising Diffusion Probabilistic Models (DDPM)

DDPM is the baseline logic underlying SDEdit, ScoreALD, and DPS later tested in this paper. DDPM adds Gaussian noise to a clean input image using a variance-preserving forward process with noise $\mathbf{z}_{t-1} \sim \mathcal{N}(0, I)$:

$$x_t = \sqrt{1-\beta_t}\mathbf{x}_{t-1} + \sqrt{\beta_t}\mathbf{z}_{t-1}, \quad t = 1, 2, ...T$$
$$x_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\mathbf{z}, \quad t = 1, 2, ...T$$

For large T more Gaussian noise is added during noisy image formation. Unlike Ho et al. [1], our DDPM implementation uses a score prediction network, s_θ, when computing x_{t-1}. This is equivalent to the noise prediction network used by Ho et al. [1], as shown in the appendix.

Algorithm 1 Reverse, Denoising Process

$x_T \sim \mathcal{N}(0, I)$
for $t = T, \dots, 1$ **do**
  $z \sim \mathcal{N}(0, I)$ if $t > 1$, else $z = 0$
  $x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t + (1-\alpha_t)s_\theta(x_t, t)\right) + \sqrt{1-\alpha_t}\, z$
end for
return x_0

Shown in Figure 3.1, the forward process produces the center noisy image for $T = 300$ using an iterative or single step process depending whether parameters $\bar{\alpha}$ or β are used. The reverse process iteratively denoises the corrupted image as in Algorithm 3.1.

3.2 SDEdit

SDEdit is an image editing process using the DDPM reverse process shown in Algorithm 3.1. The amount of added Gaussian noise, parameterized by T, controls whether the reconstructed image is more faithful to the input or realistic with regards to the diffusion model. As T increases (image becomes increasingly noisy), there is "more" unconditional generation and the output may minimally resemble the input. These effects will later be discussed.

3.3 ScoreALD

ScoreALD improves the unconditional DDPM reverse process by adding a guidance factor that incorporates measurements at the current time-step. This increases reconstruction accuracy compared to unconditional inverse imaging.

The $||\mathcal{A}(x_t) - y||^2$ term in algorithm 2 computes the measurement error by passing the current noisy image, x_t, through the forward model and comparing it to the current measurement y. Note that the forward model is originally applied to the clean image, x_0, meaning this simplification assumes $\nabla_x \log p(y \mid x_0) \approx \nabla_x \log p(y \mid x_t)$. This is more problematic in earlier steps of the reverse diffusion process where x_t may have high amounts of noise and be dissimilar to x_0. **In our implementation, the annealing factor (coefficient of the gradient term) was multiplied by 0.01 to reduce guidance factor step size and improve results.**

Algorithm 2 Score Annealed Langevin Dynamics (ScoreALD)

$x_T \sim \mathcal{N}(0, I)$
for $t = T, \dots, 1$ **do**
  $z \sim \mathcal{N}(0, I)$ if $t > 1$, else $z = 0$
  $x'_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t + (1-\alpha_t)s_\theta(x_t, t)\right) + \sqrt{1-\alpha_t}\, z$
  $x_{t-1} = x'_{t-1} - \frac{1}{2(\sigma^2+\gamma_t^2)}\nabla_{x_t}\left\|\mathcal{A}(x_t) - y\right\|^2$
end for
return x_0

3.4 Diffusion Posterior Sampling (DPS)

Like ScoreALD, DPS extends the baseline DDPM process by adding a unique guidance factor as shown in algorithm 3.4. The improvement in this guidance factor is that the gradient term uses an estimate of x_0 passed through the forward model when computing the measurement error. This is far more accurate than ScoreALD especially for highly noisy x_t.

Algorithm 3 Diffusion Posterior Sampling (DPS)

$x_T \sim \mathcal{N}(0, I)$
for $t = T, \dots, 1$ **do**
  $z \sim \mathcal{N}(0, I)$ if $t > 1$, else $z = 0$
  $\hat{x}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}\left(x_t + (1-\bar{\alpha}_t)s_\theta(x_t, t)\right)$
  $x'_{t-1} = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)}{1-\bar{\alpha}_t}\hat{x}_0 + \sqrt{1-\alpha_t}\, z$
  $x_{t-1} = x'_{t-1} - \zeta_t\nabla_{x_t}\left\|\mathcal{A}(\hat{x}_0) - y\right\|^2$
end for
return x_0

DPS computes the estimate of the clean image, $\hat{x}_0$, using Tweedie's Formula:

$$\hat{x}_0 = \mathbb{E}[x_0 \mid x_t] = \frac{1}{\sqrt{\bar{\alpha}_t}}\left(x_t + (1-\bar{\alpha}_t)\nabla_{x_t}\log p_t(x_t)\right) \quad (1)$$

4 Experimental Results

4.1 DDPM Results

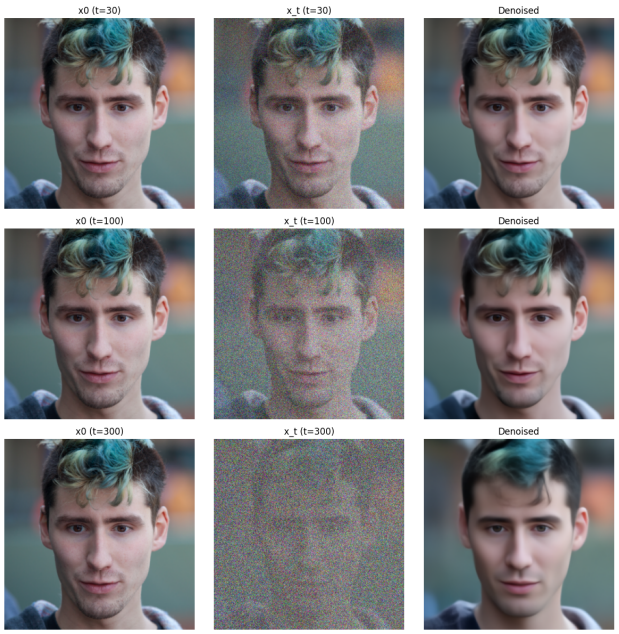

Fig. 2. DDPM single-pass denoise for $T = 30, 100, 300$.

4.2 SDEdit Results

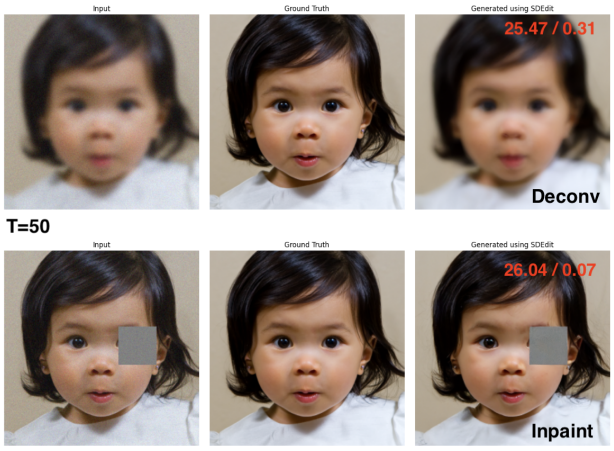

Fig. 3. SDEdit for $T = 50$, inpaint and deconvolution results.

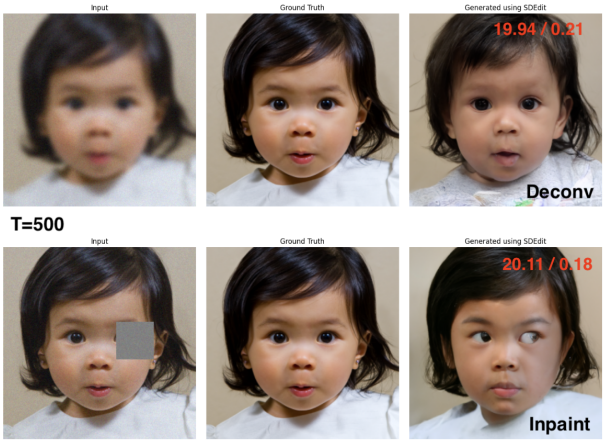

Fig. 4. SDEdit for $T = 500$, inpaint and deconvolution results.

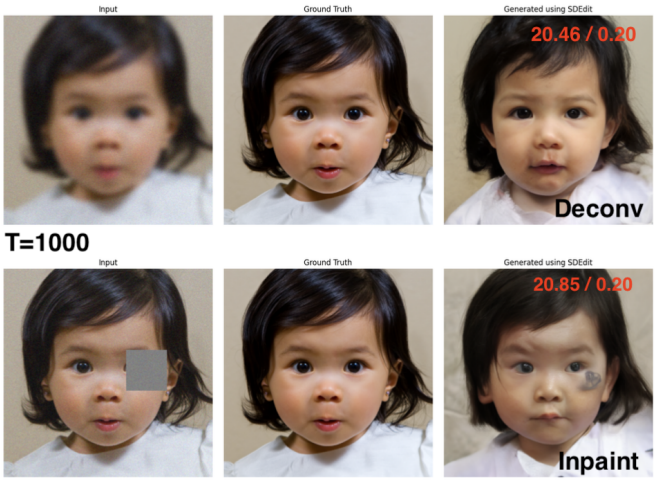

Fig. 5. SDEdit for $T = 1000$, inpaint and deconvolution results.

4.3 ScoreALD Results

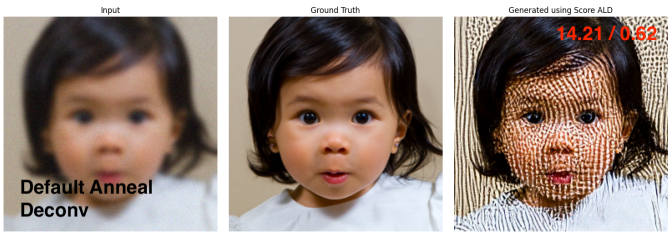

Fig. 6. Deconv ScoreALD for default annealing factor.

Fig. 7. Deconv ScoreALD for (0.01)*(annealing factor).

Fig. 8. Inpaint ScoreALD for default annealing factor.

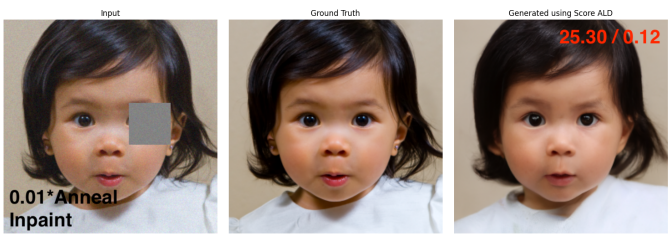

Fig. 9. Inpaint ScoreALD for (0.01)*(annealing factor).

4.4 Diffusion Posterior Sampling (DPS) Results

Fig. 10. Deconv DPS for scale=0.03.

Fig. 11. Inpaint DPS for scale=1.0.

5 Discussion

TABLE 1
Performance of diffusion methods.

		PSNR	LPIPS
DDPM (T=300)	–	27.23	0.33
SDEdit (T=500)	Deconv	19.94	0.21
	Inpaint	20.11	0.18
ScoreALD	Deconv	23.39	0.13
	Inpaint	31.42	0.13
DPS	Deconv	28.56	0.06
	Inpaint	34.41	0.02

5.1 DDPM

DDPM adequetely reconstructs the images for $T = 30, 100, 300$ with PSNR=$37.44, 32.68, 27.28$. While DDPM can denoise the inputs, there is noticeable loss of color and texture variety as the noise level increases. For $T = 30$, the reconstructed image resembles the clean image with minimal loss of detail. For $T = 300$, the subject loses facial hair texture and has a blurred skin tone. The general trend is the image appears increasingly smoothed and artificial as T increases.

5.2 SDEdit

The SDEdit results are generally poor compared to ScoreALD and DPS, though the results demonstrate how increasing noise level affects the tradeoff between faithfulness and realism. At $T = 50$, there are strong PSNR/LPIPS values purely because SDEdit returns a close-to-original image, but the blurring and inpainting are unaddressed. For high noise levels, PSNR/LPIPS worsen but the process amends the blurring and inpainting, but with noticeable differences over the right eye and cheek.

5.3 ScoreALD

ScoreALD results improve on SDEdit. Fig. 6 shows a poor PSNR of 14.21 when using the default annealing factor. The default annealing factor leaves high frequency fringes throughout the image. Decreasing the ScoreALD guidance term by a factor of 0.01 in the code improved results for deconvolution. This change increased PSNR from 14.21 to 23.39, and decreased LPIPS by a factor of ~ 4. Despite the quantitative improvement, the deconvolved image in Figure 7 still has inaccuracies in the eye and mouth shape as well as skin tone.

For the inpainting task, the default ScoreALD annealing factor better reconstructed the image with a PSNR of 31.42 and slightly noticeable error in the covered eyelid.

5.4 DPS

DPS produced the best results by far, using a scale factor of 0.03 for the deconvolution task and 1.0 for the inpainting task. DPS achieved PSNRs of 28.56 and 34.41 for deconvolution and inpainting, respectively. The DPS improvement is also qualitative – unlike ScoreALD (Figure 7), the deconvolution output has minimal error in skin tone, mouth, and eye shape. The DPS quantitative and qualitative improvements are also true for inpainting, with a 34.41 PSNR.

The improved reconstruction is from DPS usage of Tweedie's formula to estimate the clean image ($\hat{x}_0$) at each time-step. Feeding $\hat{x}_0$ through the forward model to compute measurement error is far more accurate in early stages of the reverse process, compared to ScoreALD which feeds a highly noisy x_t through the forward model.

5.5 Comparison

A key difference between the tested algorithms is how measurement information is incorporated into the reverse diffusion process. This difference is what drives the disparities in results. SDEdit reconstruction depends on the training date prior and amount of injected noise. It does not sample from a posterior distribution conditioned on measurements. ScoreALD improves reconstruction by incorporating measurement gradients in the reverse process updates. These gradients are computed using the noisy x_t. DPS computes the likelihood gradient using an estimate of the clean image $\hat{x}_0$. This significantly improves measurement consistency early in the reverse process, when x_t contains large amounts of noise. The experimental results confirm that this modification leads to both improved qualitative and quantitative results.

5.6 Limitations

Despite strong results from ScoreALD and DPS, several limitations remain. First, diffusion-based methods do not provide an exact solution to the inverse imaging problem. Instead, they generate reconstructions using a learned generative prior and a sampling process conditioned on the observed measurements. The reconstructed image may be realistic per the learned data distribution but not necessarily identical to the ground truth.

Second, diffusion-based reconstruction methods require many iterative sampling steps, making them computationally expensive compared to classical inverse imaging techniques. This computational cost may limit their practicality for large-scale applications, even if the algorithms are parallelized.

6 Conclusion

7 Conclusion

This paper evaluated the use of diffusion models to solve inverse imaging problems. Three algorithms, SDEdit, Score-ALD, and Diffusion Posterior Sampling (DPS), were tested on inpainting and deconvolution with Gaussian noise. The results show that incorporating measurement likelihoods into the diffusion process significantly improves reconstruction quality. DPS achieved the best performance both quantitatively (PSNR/LPIPS) and qualitatively, due to its use of Tweedie's formula to estimate the clean image when computing gradients.

The results highlight the potential of diffusion models as useful approaches for solving ill-posed imaging problems.

7.1 Future Work

- Improved algorithm implementation to reduce runtime and complexity (optimize code, run on better hardware e.g. GPUs)
- Evaluate performance against size and variety of training dataset (how do these methods perform when an input image cannot be interpolated from training data?)
- Test further algorithms with conditioned denoising processes and unique guidance factors.

8 Acknowledgments

The authors would like to thank professor Gordon Wetzstein and TA Minseo (Sonia) Kim for organizing and teaching EE367 Computational Imaging.

References

[1] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[2] 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," in *International Conference on Learning Representations (ICLR)*, 2022.

[3] 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*, 2021.

[4] H. Chung, J. C. 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.

9 Appendix

9.1 Writing Forward Model in Terms of $\bar{\alpha}$

We write the forward diffusion process in terms of α instead of β:

$$x_t = \sqrt{1-\beta_t}\, x_{t-1} + \sqrt{\beta_t}\, z_{t-1}. \tag{2}$$

Expanding this expression for $t = 2$ gives

$$x_2 = \sqrt{1-\beta_2}\, x_1 + \sqrt{\beta_2}\, z_1 \tag{3}$$

$$= \sqrt{1-\beta_2}\left(\sqrt{1-\beta_1}\, x_0 + \sqrt{\beta_1}\, z_0\right) + \sqrt{\beta_2}\, z_1. \tag{4}$$

Repeating this substitution recursively until reaching x_0 yields the general form for $t > 0$:

$$x_t = \left(\prod_{i=1}^{t}\sqrt{1-\beta_i}\right) x_0 + \sum_{j=1}^{t}\left(\prod_{k=j+1}^{t}\sqrt{1-\beta_k}\right)\sqrt{\beta_j}\, z_{j-1}. \tag{5}$$

Using known definitions (provided in EE367),

$$\alpha_t = 1 - \beta_t, \tag{6}$$

$$\bar{\alpha}_t = \prod_{i=1}^{t}\alpha_i = \prod_{i=1}^{t}(1-\beta_i), \tag{7}$$

$$z_t = z \sim \mathcal{N}(0, I). \tag{8}$$

The first term of Eq. 5 simplifies directly:

$$\sqrt{\bar{\alpha}_t} = \prod_{i=1}^{t}\sqrt{1-\beta_i}. \tag{9}$$

Substituting $\alpha_j = 1 - \beta_j$ into the second term gives

$$\sum_{j=1}^{t}\sqrt{1-\alpha_j}\left(\prod_{k=j+1}^{t}\sqrt{\alpha_k}\right) = \sqrt{1-\bar{\alpha}_t}. \tag{10}$$

Therefore, the forward diffusion process can be written in the compact closed form

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1-\bar{\alpha}_t}\, z. \tag{11}$$

9.2 Equivalent Forms for Reverse Update, x_{t-1}

Using the definitions from Problem 8.1 and $\bar{\alpha}_t = \alpha_t\bar{\alpha}_{t-1}$, start with the expression for x_{t-1} and substitute $\hat{x}_0$.

$$x_{t-1} = \frac{\sqrt{\alpha_t(1-\bar{\alpha}_{t-1})}}{1-\bar{\alpha}_t}x_t + \frac{\sqrt{\bar{\alpha}_{t-1}(1-\alpha_t)}}{1-\bar{\alpha}_t}\hat{x}_0 \tag{12}$$

Substitute the estimate of x_0:

$$\hat{x}_0 = \frac{x_t + (1-\bar{\alpha}_t)s_\theta(x_t, t)}{\sqrt{\bar{\alpha}_t}}. \tag{13}$$

Substituting this into the previous equation gives

$$x_{t-1} = \frac{\sqrt{\alpha_t(1-\bar{\alpha}_{t-1})}}{1-\bar{\alpha}_t}x_t + \frac{\sqrt{\bar{\alpha}_{t-1}(1-\alpha_t)}}{(1-\bar{\alpha}_t)\sqrt{\bar{\alpha}_t}}\left(x_t + (1-\bar{\alpha}_t)s_\theta(x_t, t)\right). \tag{14}$$

Expanding the terms,

$$x_{t-1} = \left(\frac{\sqrt{\alpha_t(1-\bar{\alpha}_{t-1})}}{1-\bar{\alpha}_t} + \frac{\sqrt{\bar{\alpha}_{t-1}(1-\alpha_t)}}{(1-\bar{\alpha}_t)\sqrt{\bar{\alpha}_t}}\right)x_t + \frac{\sqrt{\bar{\alpha}_{t-1}(1-\alpha_t)}}{\sqrt{\bar{\alpha}_t}}s_\theta(x_t, t). \tag{15}$$

In the right-hand side, substitute

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

Then

$$x_{t-1} = \left(\frac{\sqrt{\alpha_t(1-\bar{\alpha}_{t-1})}}{1-\bar{\alpha}_t} + \frac{\sqrt{\bar{\alpha}_{t-1}(1-\alpha_t)}}{(1-\bar{\alpha}_t)\sqrt{\bar{\alpha}_t}}\right)x_t + \frac{1-\alpha_t}{\sqrt{\alpha_t}}s_\theta(x_t, t). \tag{17}$$

Now simplify the coefficient of x_t:

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

Thus the final expression for the reverse update is

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - (1-\alpha_t) s_\theta(x_t, t) \right). \tag{19}$$

9.3 Score–Noise Prediction Equivalence in DDPM

The forward diffusion process is

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\, \mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\, \epsilon. \tag{20}$$

Solving for ϵ gives

$$\epsilon = \frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}}. \tag{21}$$

Using Tweedie's formula, posterior mean estimate of $\mathbf{x}_0$ is

$$\hat{\mathbf{x}}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}} \left(\mathbf{x}_t + (1-\bar{\alpha}_t) \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) \right) \tag{22}$$

$$= \frac{1}{\sqrt{\bar{\alpha}_t}} \left(\mathbf{x}_t + (1-\bar{\alpha}_t) s_\theta(\mathbf{x}_t, t) \right). \tag{23}$$

Substituting $\hat{\mathbf{x}}_0$ into the expression for ϵ,

$$\epsilon = \frac{\mathbf{x}_t - (\mathbf{x}_t + (1-\bar{\alpha}_t) s_\theta(\mathbf{x}_t, t))}{\sqrt{1-\bar{\alpha}_t}} \tag{24}$$

$$= -\sqrt{1-\bar{\alpha}_t}\, s_\theta(\mathbf{x}_t, t). \tag{25}$$

Rearranging gives the equivalence between score prediction and noise prediction:

$$s_\theta(\mathbf{x}_t, t) = -\frac{1}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(\mathbf{x}_t, t). \tag{26}$$

The DDPM reverse step can be parameterized using either a score network or a noise prediction network.