Solving Inverse Problems in Imaging with Diffusion Models

Matthew M. Sato, satomm@stanford.edu

Abstract—Diffusion models are a promising method for solving inverse problems in imaging. Given a noisy measurement, diffusion models can incrementally denoise the image by removing some of the noise at each step of the process. In this project, the notation and diffusion model processes used by various approaches are unified. Then, a variety of diffusion model approaches are applied to the inverse problem, with varying levels of conditioning on the noisy measurement. These approaches are demonstrated on a diffusion model trained on human faces. The ability to generate an image of a human face without conditioning on the noisy measurement is first shown as a baseline. Then, SDEdit, a method which uses a heuristic to guide the image towards the original noisy image, is examined. Last, three methods (ILVR, ScoreALD, and DPS) which explicitly estimate the gradient of the log likelihood are investigated and compared. Ultimately, this project shows that diffusion models are an effective method for solving inverse problems in imaging.

Index Terms—Computational Imaging, Inverse Problems, Diffusion Models

1 Introduction

Noise in images is a longstanding problem, with imperfections or limitations of cameras adding noise to images. For example, a moving camera or moving object may create motion blur, an improperly focused camera may create a blurry image, or an obstruction may block a part of an image. Solving the inverse problem, or reconstructing a denoised image from the noisy image, is a difficult problem because there are an infinite number of possible solutions based on the captured image.

Although many approaches have been proposed to solve inverse problems in imaging, the recent advent of diffusion models show significant promise. Diffusion models are a generative machine learning method and learn to iteratively denoise an image, transforming noise to a clearer representation of an image at each step of the process. To train a diffusion model, noise is gradually added to images and a deep neural network learns to reverse this process. Diffusion models are most popular for their applications in image and video generation [1], where diffusion models transform noise into a realistic image. As investigated in this project, diffusion models can be extended to solve inverse problems by conditioning the process on a noisy measured image.

Exactly conditioning the denoising process on the measured image is intractable. However, both heuristics and approximations have been proposed for the conditioning step. This project compares different heuristics/approximations and evaluates their effectiveness for solving inverse problems, specifically the inpainting and deconvolution problems. Since diffusion models are relatively new for solving inverse problems, this project does not compare the diffusion models to existing approaches, but rather compares different diffusion model appraoches. In particular, the SDEdit [2], ILVR [3], ScoreALD [4], and DPS [5] methods are compared.

In this project, unconditioned and conditioned diffusion models are explained and notation used by different methods are unified. The forward noising process and unconditional image generation is shown as a baseline. A heuristic approach (SDEedit) which guides the image generation towards the original image without explicit conditioning is investigated. Last, three methods (ILVR, ScoreALD, DPS) that solve the inverse problem by conditioning on a noisy captured image are compared. The remainder of this paper includes Related Work in Section 2, Methods in Section 3, Evaluation Metrics in Section 4, Results in Section 5, and a Discussion and Conclusion in Section 6.

2 Related Work

The inverse problem for imaging is not new, with many approaches proposed over the years. Popular methods include optimization-based approaches, such as the Half Quadratic Splitting (HQS) method and the Alternating Direction Methods of Multipliers (ADMM) [6]. More recently, with the explosion of computing power and data, neural networks trained under supervised learning have been proposed for solving the inverse problem [7]. However, the HQS/ADMM approaches can be slow to converge (if at all) and the deep learning approach performs poorly for out-of-distribution samples.

Generative adversarial networks (GANs) are a related method that are also generative. GANs simultaneously trains a generator to generate an image and a discriminator to detect generated images. GANs have been proposed for solving inverse problems for imaging [8]. However, GANs are notoriously unstable and difficult to train, limiting their practical use.

More recently, diffusion models have been proposed for image generation. The first description of using diffusion models for images was by [9], where the basic framework for diffusion models was proposed. Besides the approaches for inverse problems examined in this project, a non-exhaustive list of other techniques include Score-SDE [10],

- *This is the final project for the Winter 2025 iteration of EE367 at Stanford*

IIGDM [11], BlindDPS [12], and Moment Matching [13]. All of these methods condition the diffusion model by explicitly making some approximation to match the measurement.

3 METHODS

In this section, the basic formulation of diffusion models for solving inverse problems is presented. Since notation can vary widely, the approaches are unified with a common notation. Then, the methods examined in this project for solving inverse problems are described.

3.1 Diffusion Models

The diffusion model can be separated into a forward noise process and a reverse denoising process. In the forward noise process, noise is gradually added to an image using a forward noise model. In this case, $\boldsymbol{x}_0$ is the original unnoisy image and $\boldsymbol{x}_t$ is the image after t steps of added noise. This project uses the variance-preserving (VP) formulation of diffusion models. In the VP formulation provided by [9], the forward noise model is described by

$$\boldsymbol{x}_t = \sqrt{1-\beta_t}\boldsymbol{x}_{t-1} + \sqrt{\beta_t}\boldsymbol{z}_{t-1}, \tag{1}$$

where β_t is the noise schedule and $\boldsymbol{z}_{t-1} \sim \mathcal{N}(0, I)$. This forward noise model is computationally efficient for any t if rewritten into an equivalent formulation depending only on the original image and a single noise term:

$$\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{z}, \tag{2}$$

where $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$, and $\boldsymbol{z} \sim \mathcal{N}(0, I)$. The derivation for this formulation is provided in the Appendix.

In the reverse denoising process, a diffusion model iteratively reverses the noising process of (2). Tweedie's formula provides the estimate for the completely denoised image at time t:

$$\hat{\boldsymbol{x}}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}(\boldsymbol{x}_t + (1-\bar{\alpha}_t)\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)) \tag{3}$$

The learnable function of a diffusion model is the score function, $\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)$, which is trained to match $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)$. Details on training the score function are beyond the scope of this project. Using this score function, the approximation for the denoised image is

$$\hat{\boldsymbol{x}}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}(\boldsymbol{x}_t + (1-\bar{\alpha}_t)\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)) \tag{4}$$

and the incremental denoising step can be written as

$$\boldsymbol{x}_{t-1} = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\boldsymbol{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)}{1-\bar{\alpha}_t}\hat{\boldsymbol{x}}_0 + \sigma\boldsymbol{z} \tag{5}$$

where the Gaussian noise $\boldsymbol{z}$ adds robustness and guarantees unique denoising steps when the algorithm is run repeatedly. An alternative but equivalent denoising step can be derived by substituting $\hat{\boldsymbol{x}}_0$ from (4) into (5) (derived in the Appendix), such that a single step of the reverse diffusion process can be written:

$$\boldsymbol{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t + (1-\alpha_t)\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)\right) + \sigma\boldsymbol{z}. \tag{6}$$

Algorithm 1 Reverse Diffusion

$\boldsymbol{x}_T \sim \mathcal{N}(0, I)$
for $t = T$ to 1 **do**
  $\boldsymbol{z} \sim \mathcal{N}(0, I)$ if $t > 1$, else $\boldsymbol{z} = 0$
  $\hat{\boldsymbol{x}}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}(\boldsymbol{x}_t + (1-\bar{\alpha}_t)\boldsymbol{s}_\theta(\boldsymbol{x}_t, t))$
  $\boldsymbol{x}'_{t-1} = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\boldsymbol{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)}{1-\bar{\alpha}_t}\hat{\boldsymbol{x}}_0 + \sigma\boldsymbol{z}$
  $\boldsymbol{x}_{t-1} = \boldsymbol{x}'_{t-1} + \zeta_t g(\boldsymbol{x}_t, \boldsymbol{y})$
end for
return $\boldsymbol{x}_0$

Although the two formulations are equivalent, the denoising step using (4) and (5) is used in the remainder of this project.

The reverse diffusion process described has been formulated using the score function, $\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)$; however, a noise-prediction network, $\boldsymbol{\epsilon}_\phi(\boldsymbol{x}_t, t)$, can be learned instead. In this case, a single reverse denoising step is

$$\boldsymbol{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)\right), \tag{7}$$

where the equivalence of the formulation in (6) and (7) is derived in the Appendix. Since these approaches are equivalent, the score function formulation is used for the remainder of this project.

3.2 Conditioned Diffusion Models

Thus far, the *unconditional* denoising process has been described. However, to solve an inverse problem, the diffusion process should be conditioned on the original noisy measurement, $\boldsymbol{y}$. The image formation model is $\boldsymbol{y} = A(\boldsymbol{x}) + \boldsymbol{n}$, where $A(\cdot)$ is the noisy measurement operator and $\boldsymbol{n}$ is zero mean Gaussian noise. Although $A(\cdot)$ can be both non-linear and linear, only linear operators are considered in this project since only some of the methods are applicable for non-linear operators.

To condition the diffusion process on the measurement, $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)$ in (3) is replaced with $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{y})$. Using Baye's rule,

$$\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{y}) = \underbrace{\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)}_{\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)} + \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{y}|\boldsymbol{x}_t) \tag{8}$$

The gradient of the log likelihood,$\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{y}|\boldsymbol{x}_t)$, is intractable, requiring an approximation. The approaches explored in this project use various approximations for $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{y}|\boldsymbol{x}_t)$. To create a common notation for the different approaches, let

$$\zeta_t g(\boldsymbol{x}_t, \boldsymbol{y}) \approx \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{y}|\boldsymbol{x}_t), \tag{9}$$

where $g(\boldsymbol{x}_t, \boldsymbol{y})$ is the approximation for the gradient of the log likelihood and ζ_t is an approximation dependent scaling term. Then, the denoising process described in (4) and (5) can be updated to include the measurement conditioning. The complete denoising process is summarized in Algorithm 1 and the remaining subsections details the different heuristics or approximations used by different approaches for $g(\boldsymbol{x}_t, \boldsymbol{y})$ and ζ_t.

TABLE 1
The posterior approximation and scale for the various methods.

Method	$g(\boldsymbol{x}_t, \boldsymbol{y})$	ζ_t
SDEdit	0	0
ILVR	$\phi_N(\boldsymbol{y}_{t-1}) - \phi_N(\boldsymbol{x}'_{t-1})$	1
ScoreALD	$-\nabla_{x_t}\|\boldsymbol{y} - A(\boldsymbol{x}_t)\|_2^2$	$\frac{1}{\sigma^2+\gamma_t^2}$
DPS	$-\nabla_{x_t}\|\boldsymbol{y} - A(\hat{\boldsymbol{x}}_0)\|_2^2$	$\frac{\zeta}{\|\boldsymbol{y}-A(\hat{\boldsymbol{x}}_0)\|_2}$

3.3 Conditioned Diffusion Model Methods

In this subsection, the different approaches for solving the conditioned diffusion problem for inverse problems are described. The choices for $g(\boldsymbol{x}_t, \boldsymbol{y})$ and ζ_t are summarized in Table 1 for all the methods described in this subsection.

3.3.1 SDEdit

The first method is SDEdit [2], which does not attempt to approximate $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{y}|\boldsymbol{x}_t)$, but instead uses a heuristic to guide the reverse diffusion process towards the measured image. SDEdit starts at an intermediate denoising step and instead of initializing $\boldsymbol{x}_T$ with random noise, SDEdit combines the noise with the measured image by using (2). Thus, line 1 in Algorithm 1 is replaced with

$$x_T = \sqrt{\bar{\alpha}_T}\boldsymbol{y} + \sqrt{1-\bar{\alpha}_T}\boldsymbol{z}, \tag{10}$$

where $\boldsymbol{z} \sim \mathcal{N}(0, I)$. The remaining steps remain the same, with $g(\boldsymbol{x}_t, \boldsymbol{y}) \equiv 0$ and $\zeta_t = 0 \quad \forall \quad t$. The starting step T is a hyperparameter which should be tuned. A larger T results in more initial noise leading to larger realism, while a smaller T has less noise and is more faithful to the original measurement.

3.3.2 ILVR

ILVR [3] is the first method investigated in this project that explicitly conditions the denoising on the measurement. ILVR uses a low-pass filtering operation, $\phi_N(\cdot)$, which is a sequence of downsampling and upsampling by a factor of N. The goal of ILVR is to match the downsampled version of the generated image to the downsampled version of the measured image: $\phi_N(\boldsymbol{x}_0) = \phi_N(\boldsymbol{y})$. This matching is enforced at each step of the reverse diffusion process: $\phi_N(\boldsymbol{x}_t) = \phi_N(\boldsymbol{y}_t)$, where $\boldsymbol{y}_t$ uses the forward noise process described in (2). Thus, the approximation becomes

$$g(\boldsymbol{x}_t, \boldsymbol{y}) = \phi_N(\boldsymbol{y}_{t-1}) - \phi_N(\boldsymbol{x}'_{t-1}), \tag{11}$$

with $\zeta_t = 1$. This process can be viewed as removing the low-frequency component of the current approximation and adding the low-frequency component of the associated measurement. This correction term is only applied for $t > b$, where b is the stopping time for the correction. The stopping time, b, and the downsampling rate, N, are both hyperparameters of this method.

3.3.3 ScoreALD

In ScoreALD [4], the gradient of the log likelihood is estimated by using the current $\boldsymbol{x}_t$:

$$\zeta_t g(\boldsymbol{x}_t, \boldsymbol{y}) \approx \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{y}|\boldsymbol{x}_t) \approx -\frac{1}{\sigma^2+\gamma_t^2}\nabla_{x_t}\|\boldsymbol{y} - A(\boldsymbol{x}_t)\|_2^2 \tag{12}$$

where the estimate can be computed with backpropagation. The approximation in (12) is only correct for $t = 0$, and the authors propose an annealing term, γ_t, that reduces the scale ζ_t for large t when the approximation is poor. The set of annealing terms $\{\gamma_t\}$ is a hyperparameter that must be chosen for each problem.

3.3.4 DPS

The DPS method [5] estimates the gradient of the log likelihood term similarly to ScoreALD. However, instead of using $\boldsymbol{x}_t$, DPS uses the estimate of the denoised imaged, $\hat{\boldsymbol{x}}_0$:

$$\zeta_t g(\boldsymbol{x}_t, \boldsymbol{y}) \approx \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{y}|\boldsymbol{x}_t) \approx -\zeta_t \nabla_{x_t}\|\boldsymbol{y} - A(\hat{\boldsymbol{x}}_0)\|_2^2, \tag{13}$$

where $\hat{\boldsymbol{x}}_0$ is computed as in line 4 of Algorithm 1 and the gradient is computed with backpropagation. The ζ_t term is a hyperparameter, and the authors suggest using $\zeta_t = \frac{\zeta}{\|\boldsymbol{y}-A(\hat{\boldsymbol{x}}_0)\|}$, with $\zeta \in [0.1, 1.0]$. Furthermore, the authors quantify an upper bound error on the approximation, although the details are beyond the scope of this project.

4 Evaluation and Comparison of the Methods

To evaluate the different solution methods for solving inverse problems with diffusion models, two metrics are used: peak signal-to-noise-ratio (PSNR) and Learned Perceptual Image Patch Similarity (LPIPS) distance. PSNR is a commonly used metric to compare different image signals and is defined as

$$\text{PSNR} = 10\log_{10}\left(\frac{MAX^2}{MSE}\right), \tag{14}$$

where MAX is the maximum possible value of a pixel and MSE is the mean squared error between the image and the ground truth. Although PSNR is a commonly used metric and can evaluate the similarity between two images, the metric is criticized for not representing visual quality well, since a reconstruction with a large PSNR may not look good to a human observer.

LPIPS, however, is a better metric for analyzing the visual quality. LPIPS compares the distance between activation layers from a neural network for the reconstructed image and the ground truth [14]. The key idea for LPIPS is that the activation layers of a neural network aligns more closely to the features that the human eye observes. Thus, a lower LPIPS value indicates a better reconstruction of a noisy image.

Two different forward noise models are considered for evaluating the methods: inpainting and deconvolution. For inpainting, a 50×50 pixel box is masked out of the original image. In deconvolution, a Gaussian blur kernel of size 61 and standard deviation 3 is applied to the image. The PSNR and LPIPS values are computed for each method and the reconstructed images are qualitatively compared. The different methods are all evaluated on the same image for consistency. The ground truth and noisy measurements of the image to be tested is shown in Fig. 1.

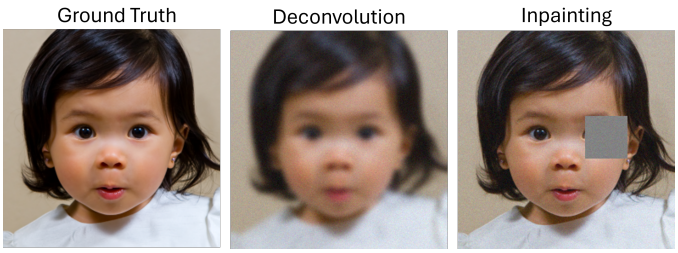

Fig. 1. The ground truth, inpainting, and denconvolution images.

5 Results

In this section, the proposed methods are evaluated. First, unconditional generation with a diffusion model is demonstrated along with estimates of a denoised image. Then, the results from the proposed methods, SDEdit, ILVR, ScoreALD, and DPS, are shown and compared.

5.1 Diffusion Model

A pretrained diffusion model from [5] is used. The diffusion model is trained on the Flickr-Faces-HQ (FFHQ) dataset [15], a dataset with a wide variation of human faces. This diffusion model is trained to learn the score function, $\boldsymbol{s}_\theta(\boldsymbol{x}_t, t) = \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)$, rather than the noise function. Implementation of the diffusion model and inverse problem methods are implemented with PyTorch.

First, the diffusion model's ability to unconditionally generate an image from noise is demonstrated. Specifically, no posterior sampling is used and $g(\boldsymbol{x}_t, \boldsymbol{y}) = 0$ in line 6 of Algorithm 1. Starting from $\boldsymbol{x}_{1000} \sim \mathcal{N}(0, I)$ and unconditionally denoising for 1000 steps, Fig. 2 shows several examples of denoised images. As seen in Fig. 2, each time the algorithm is executed a different image is generated due to the random starting noise. Fig. 2 demonstrates the importance of conditioning the denoising process on the measurement to solve the inverse problem instead of generating a random image.

Next, the estimate of the denoised image at a given time step is investigated. First, noise is added to the image at time step t using (2). Then, the estimate for the denoised image is computed using (4). An example of this estimate is shown for a human face and a red panda face in Fig. 3, with the resulting PSNR and LPIPS shown in Table 2. Predictably, the PSNR and LPIPS show better results when predicting for a smaller t, since there is less noise in the image. As t becomes larger, the predicted denoised image is farther from the ground truth. As expected, the results for the red panda are worse than for the human face, which can be attributed to the diffusion model being trained on human faces rather than animal faces.

5.2 Results of Investigated Methods

This subsection shows the results from using SDEdit, ILVR, ScoreALD, and DPS on the inverse problem shown in Fig. 1. The quantitative results (PSNR and LPIPS) are summarized for each method in Table 3.

Fig. 2. Images created through unconditional denoising using $T = 1000$.

TABLE 2
Evaluation metrics for the estimated denoised image at various noising steps.

	Human		Red Panda	
t	PSNR	LPIPS	PSNR	LPIPS
30	37.4	0.0360	34.4	0.0473
100	32.6	0.0886	29.2	0.219
300	27.2	0.204	25.1	0.574
500	23.5	0.327	21.4	0.603

5.2.1 SDEdit

SDEdit is implemented with start times $T = \{250, 500, 750\}$ to demonstrate the tradeoff between realism and faithfulness. The results shown in Fig. 4 show that with a smaller starting time, the denoised image is more faithful to the original measurement but less real. For $T = 250$, this results in a poor denoising with the deconvolution result still showing blur and the inpainting result still showing a blocked out region. As the starting time becomes larger, the image become more real but less faithful to the measurement. The images for $T = 750$ show the most human looking faces, but do not resemble the girl from the measurement. Choosing an intermediate starting time, $T = 500$, results in a compromise between realism and faithfulness. These images don't show any residual deconvolution or inpainting, but only somewhat resemble the original girl.

The quantitative results summarized in Table 3 reflect the tradeoff between realism and faithfulness. The PSNR and LPIPS show the best values for a small T, showing that these are the most faithful to the ground truth. What these quantitative measures fail to capture are the realness that are lacking in the results for $T = 250$.

5.2.2 ILVR

ILVR is implemented for downsampling/upsampling rates of $N = \{4, 8, 16\}$. The denoising process starts from $T =$

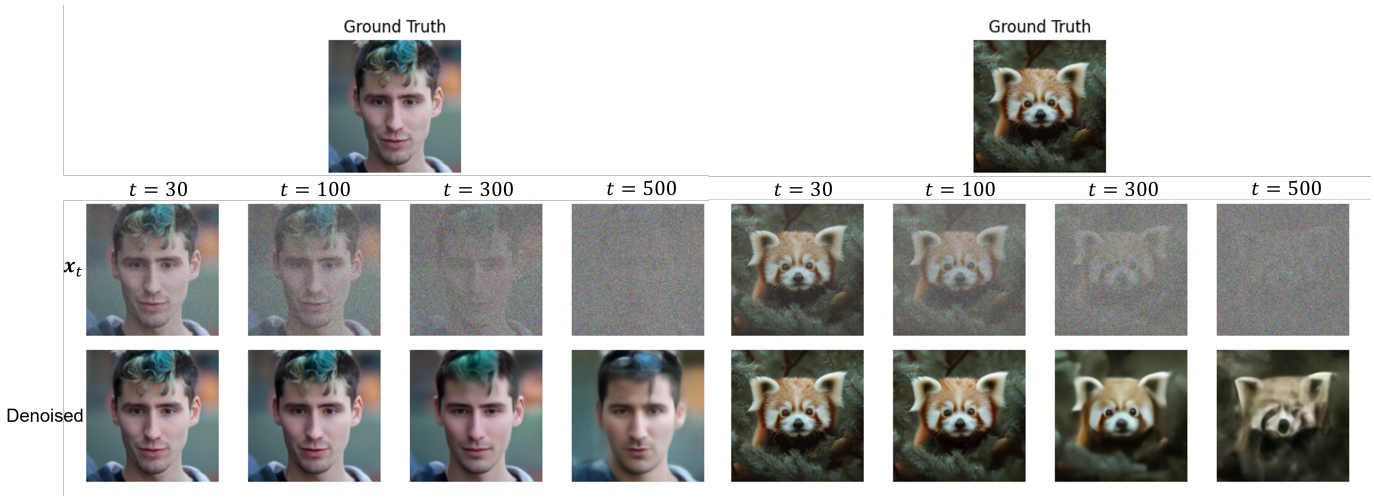

Fig. 3. Estimates of the denoised images at various levels of noise.

TABLE 3
A summary of the PSNR and LPIPS for all the investigated methods.

	SDEdit			ILVR			ScoreALD			DPS		
	T	PSNR	LPIPS	N	PSNR	LPIPS	Anneal Sch.	PSNR	LPIPS	ζ	PSNR	LPIPS
Inpainting	250	23.5	0.139	4	20.8	0.197	[10,15]	24.3	0.110	0.1	29.7	0.073
	500	20.4	0.186	8	20.2	0.189	[17, 22]	26.3	0.079	0.3	34.6	0.028
	750	14.6	0.410	16	19.6	0.240				**1**	**36.3**	**0.010**
Deconvolution	250	23.8	0.183	4	23.3	0.180	[10,15]	23.8	0.138	0.1	25.1	0.091
	500	20.2	0.233	8	23.3	0.144	[15,20]	21.7	0.158	0.3	27.0	0.078
	750	14.2	0.400	16	20.7	0.192				**1**	**28.3**	**0.054**

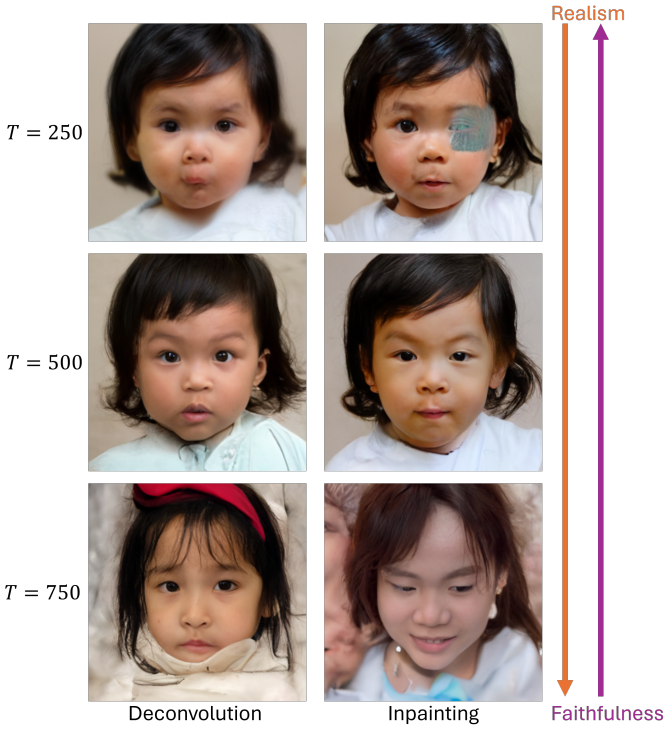

Fig. 4. The results from SDEdit using different starting times.

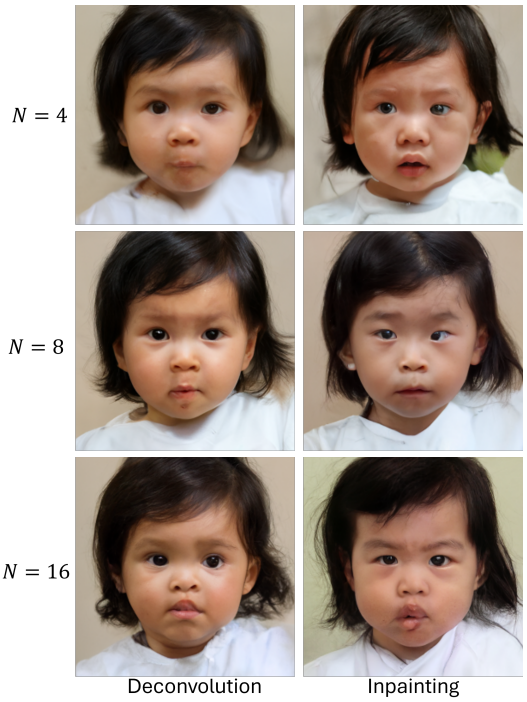

Fig. 5. The results from ILVR. The ILVR adjustment was stopped at $t = 300$ for deconvolution and $t = 500$ for inpainting.

1000, and the conditioning is applied for $t \in [300, 1000]$ for deconvolution and $t \in [500, 1000]$ for inpainting. The results are shown in Fig. 5 and the quantitative results are summarized in Table 3. Like SDEdit, faithfulness decreases as N increases. A larger downsampling factor leads to a correction that is less like the original measurement.

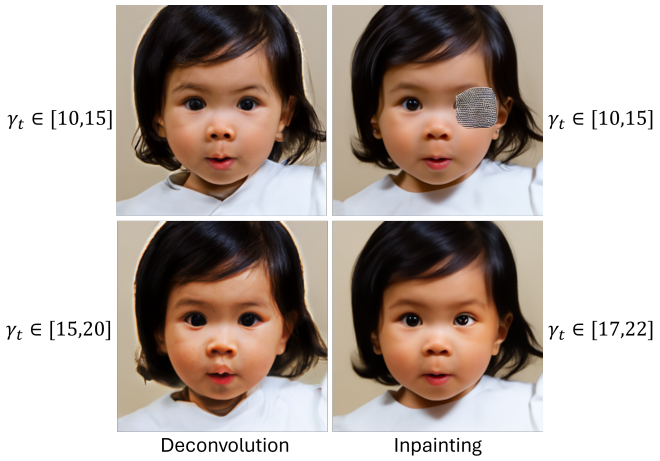

Fig. 6. The results from ScoreALD using different annealing schedules, γ_t.

However, with too small of a downsampling ($N = 4$), the noise of the original measurement is not filtered enough and artifacts of noise remain in the denoised image: blurriness for the deconvolution problem and an inpainting artifact for the inpainting problem. The PSNR and LPIPS values show that the intermediate $N = 8$ downsampling rate images have the best quantitative results, where this level of downsampling guides the denoising process towards the measurement while removing the deconvolution or inpainting. ILVR results in PSNR and LPIPS that are comparable with SDEdit and do not show significant improvement.

5.2.3 ScoreALD

ScoreALD is implemented and tested for different annealing schedules with $T = 1000$. The final denoised result are highly sensitive to the annealing schedule and this hyperparameter must be tuned to yield acceptable results. The annealing schedule is a linear schedule $\gamma_t \in [a, b]$, where $\gamma_T = b$ and $\gamma_0 = a$. The results for deconvolution and inpainting under various annealing schedules are shown in Fig. 6 and the quantitative results are in Table 3. ScoreALD results in images that most closely resemble the ground truth of all methods thus far. The PSNR and LPIPS show significant improvements compared to SDEdit and ILVR. However, the importance of choosing a proper annealing schedule must be emphasized, since a poor annealing schedule may result in poor results such as the inpainting solution for $\gamma_t \in [10, 15]$. Because the approximation error of the gradient of log likelihood may be large, a large enough γ_t is required to reduce the contributions of $g(\boldsymbol{x}_t, \boldsymbol{y})$ when the approximation error is large.

5.2.4 DPS

DPS is implemented and evaluated using $T = 1000$ and different scales, ζ. As shown in Fig. 7, the resulting images are very close to the original ground truth. Additionally, the denoised images are not very sensitive to the hyperparameter, ζ. The quantitative results in Table 3 support the qualitative results, with the PSNR and LPIPS outperforming all other methods no matter the choice of ζ. The choice of scale, ζ, does affect the results, but even a poor choice of ζ produces a good reconstruction. The approximation of $g(\boldsymbol{x}_t, \boldsymbol{y})$ clearly does a good job in guiding the diffusion process towards the ground truth, diminishing the dependence on choosing a good ζ.

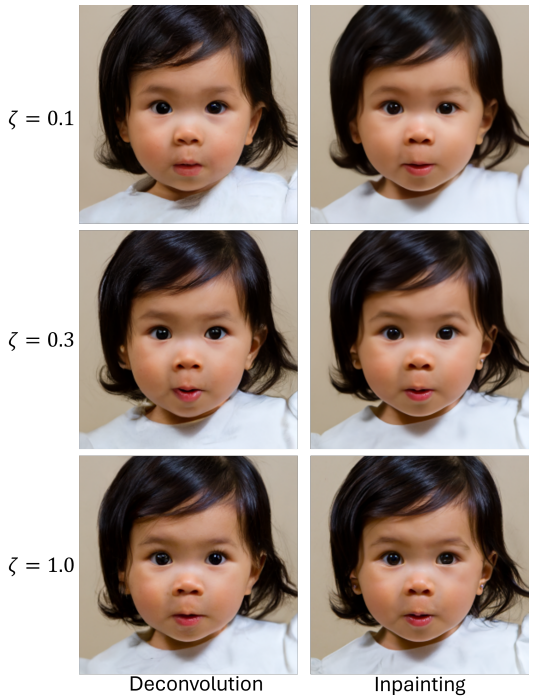

Fig. 7. The results from DPS using different scale values, ζ.

6 Discussion and Conclusion

In this project, several methods are shown for solving inverse problems with a diffusion model conditioned on a noisy measurement image. The methods are evaluated on a noisy human face. First, the SDEdit and ILVR methods are shown. SDEdit does not approximate the gradient of the log likelihood, and suffers from a tradeoff between realism and faithfulness. ILVR attempts to match the low-frequency components of the reconstructed image and the measured image. ILVR shows varying results based on the choice of downsampling rate.

DPS produces the best results on the test image. The DPS approximation for the gradient of the log likelihood has proven error bounds, which may contribute to its superior performance. DPS is also notable due to its low sensitivity to hyperparameter choice, ζ. The second best method is ScoreALD, with an approximation similar to DPS that is less accurate. ScoreALD requires a good choice of annealing schedule to produce consistent results. While DPS and ScoreALD outperform the other methods, these two methods are also more computationally expensive. DPS and ScoreALD require backpropagation for computing $g(\boldsymbol{x}_t, \boldsymbol{y})$, which results in a slower denoising process. Note, however, that the original ScoreALD paper [4] shows a closed form solution $g(\boldsymbol{x}_t, \boldsymbol{y}) \approx A^H(\boldsymbol{y} - A\boldsymbol{x}_t)$ that may be faster.

Ultimately, this project has shown various approaches to solving the inverse problem for imaging using diffusion models. As shown, generative diffusion models can

reconstruct high quality images from a noisy ground truth, providing a new method for solving inverse problems.

REFERENCES

[1] H. Cao, C. Tan, Z. Gao, Y. Xu, G. Chen, P.-A. Heng, and S. Z. Li, "A Survey on Generative Diffusion Models," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 7, pp. 2814–2830, Jul. 2024.

[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*, 2022.

[3] J. Choi, S. Kim, Y. Jeong, Y. Gwon, and S. Yoon, "ILVR: Conditioning Method for Denoising Diffusion Probabilistic Models," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 14 347–14 356.

[4] A. Jalal, M. Arvinte, G. Daras, E. Price, A. G. Dimakis, and J. Tamir, "Robust Compressed Sensing MRI with Deep Generative Priors," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. W. Vaughan, Eds., vol. 34, 2021, pp. 14 938–14 954.

[5] 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*, 2023.

[6] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[7] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[8] V. Shah and C. Hegde, "Solving Linear Inverse Problems Using Gan Priors: An Algorithm with Provable Guarantees," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2018, pp. 4609–4613, iSSN: 2379-190X.

[9] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS '20, Dec. 2020, pp. 6840–6851.

[10] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-Based Generative Modeling through Stochastic Differential Equations," in *International Conference on Learning Representations*, 2021.

[11] J. Song, A. Vahdat, M. Mardani, and J. Kautz, "Pseudoinverse-Guided Diffusion Models for Inverse Problems," in *International Conference on Learning Representations*, 2023.

[12] H. Chung, J. Kim, S. Kim, and J. C. Ye, "Parallel Diffusion Models of Operator and Image for Blind Inverse Problems," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 6059–6069.

[13] F. Rozet, G. Andry, F. Lanusse, and G. Louppe, "Learning Diffusion Priors from Observations by Expectation Maximization," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[14] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 586–595, iSSN: 2575-7075.

[15] T. Karras, S. Laine, and T. Aila, "A Style-Based Generator Architecture for Generative Adversarial Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 12, pp. 4217–4228, Dec. 2021.

APPENDIX

In this appendix, several derivations relevant for implementing the diffusion model are shown.

First, the equivalency of (1) and (2) is demonstrated. Recall, the forward noise model of a variance-preserving diffusion model is:

$$\boldsymbol{x}_t = \sqrt{1-\beta_t}\boldsymbol{x}_{t-1} + \sqrt{\beta_t}\boldsymbol{z}_{t-1} = \sqrt{\alpha_t}\boldsymbol{x}_{t-1} + \sqrt{\beta_t}\boldsymbol{z}_{t-1},$$

where $\alpha_t = 1-\beta_t$ and $\boldsymbol{z}_i \sim \mathcal{N}(0, I)$. This formulation can be rewritten as a conditional distribution that depends on only $t=0$:

$$\begin{aligned}
\boldsymbol{x}_t &= \sqrt{\alpha_t}\boldsymbol{x}_{t-1} + \sqrt{\beta_t}\boldsymbol{z}_{t-1} \\
&= \sqrt{\alpha_t}(\sqrt{\alpha_{t-1}}\boldsymbol{x}_{t-2} + \sqrt{\beta_{t-1}}\boldsymbol{z}_{t-2}) + \sqrt{\beta_t}\boldsymbol{z}_{t-1} \\
&= \sqrt{\alpha_t\alpha_{t-1}}\boldsymbol{x}_{t-2} + \sqrt{\alpha_t\beta_{t-1}}\boldsymbol{z}_{t-2} + \sqrt{\beta_t}\boldsymbol{z}_{t-1} \\
&\vdots \\
&= \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \underbrace{\sqrt{\beta_t}\boldsymbol{z}_{t-1} + \sqrt{\alpha_t\beta_{t-1}}\boldsymbol{z}_{t-2} + \cdots + \sqrt{\prod_{i=2}^{t}\alpha_i\beta_1}\boldsymbol{z}_0}_{(\star)}
\end{aligned}$$

where $\bar{\alpha}_t = \prod_{i=1}^{t}\alpha_i$. Recall, $\boldsymbol{z}_i \sim \mathcal{N}(0, I)$, and using the properties of normal distributions results in

$$(\star) \sim \mathcal{N}(0, \underbrace{(\beta_t + \alpha_t\beta_{t-1} + \alpha_t\alpha_{t-1}\beta_{t-2} + \cdots + \prod_{i=2}^{t}\alpha_i\beta_1)}_{(\diamond)} I)$$

Simplifying the covariance term:

$$\begin{aligned}
(\diamond) &= \beta_t + \alpha_t\beta_{t-1} + \alpha_t\alpha_{t-1}\beta_{t-2} + \cdots + \prod_{i=2}^{t}\alpha_i\beta_1 \\
&= 1 - \alpha_t + \alpha_t\beta_{t-1} + \alpha_t\alpha_{t-1}\beta_{t-2} + \cdots + \prod_{i=2}^{t}\alpha_i\beta_1 \\
&= 1 - \alpha_t\left(1 - \beta_{t-1} - \alpha_{t-1}\beta_{t-2} - \cdots - \prod_{i=2}^{t-1}\alpha_i\beta_1\right) \\
&= 1 - \alpha_t\alpha_{t-1}\left(1 - \beta_{t-2} - \cdots - \prod_{i=2}^{t-2}\alpha_i\beta_1\right) \\
&\vdots \\
&= 1 - \alpha_t\alpha_{t-1}\cdots\alpha_2(1-\beta_1) \\
&= 1 - \bar{\alpha}_t
\end{aligned}$$

Thus, $(\star) \sim \mathcal{N}(0, (1-\bar{\alpha}_t)I) = \sqrt{1-\bar{\alpha}_t}\mathcal{N}(0, I)$ and the forward diffusion step can be written as

$$\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{z},$$

where $\boldsymbol{z} \sim \mathcal{N}(0, I)$.

■

Next, the equivalency of the reverse diffusion process of (4)/(5) and (6) is shown. First, note that $\sqrt{\bar{\alpha}_{t-1}/\bar{\alpha}_t} = \sqrt{\bar{\alpha}_{t-1}/(\alpha_t\bar{\alpha}_{t-1})} = 1/\sqrt{\alpha_t}$. Substituting $\hat{\boldsymbol{x}}_0$ from (4) into

the expression for $\boldsymbol{x}_{t-1}$ in (5):

$$\begin{aligned}
\boldsymbol{x}_{t-1} &= \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\boldsymbol{x}_t \\
&\quad + \frac{\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)}{\sqrt{\bar{\alpha}_t}(1-\bar{\alpha}_t)}(\boldsymbol{x}_t + (1-\bar{\alpha}_t)\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)) \\
&= \left(\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} + \frac{\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)}{\sqrt{\bar{\alpha}_t}(1-\bar{\alpha}_t)}\right)\boldsymbol{x}_t \\
&\quad + \frac{\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)}{\sqrt{\bar{\alpha}_t}}\boldsymbol{s}_\theta(\boldsymbol{x}_t, t) \\
&= \frac{1}{(1-\bar{\alpha}_t)}\left(\sqrt{\alpha_t}(1-\bar{\alpha}_t/\alpha_t) + \frac{(1-\alpha_t)}{\sqrt{\alpha_t}}\right)\boldsymbol{x}_t \\
&\quad + \frac{(1-\alpha_t)}{\sqrt{\alpha_t}}\boldsymbol{s}_\theta(\boldsymbol{x}_t, t) \\
&= \frac{1}{(1-\bar{\alpha}_t)}\left(\sqrt{\alpha_t} - \bar{\alpha}_t/\sqrt{\alpha_t} + 1/\sqrt{\alpha_t} - \sqrt{\alpha_t}\right)\boldsymbol{x}_t \\
&\quad + \frac{(1-\alpha_t)}{\sqrt{\alpha_t}}\boldsymbol{s}_\theta(\boldsymbol{x}_t, t) \\
&= \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t + (1-\alpha_t)\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)\right),
\end{aligned}$$

which is the formulation of (6).

■

Last, the equivalency of the reverse diffusion process using the score function and noise-prediction network is shown. Notice that if $\boldsymbol{s}_\phi(\boldsymbol{x}_t, t) = -\frac{\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)}{\sqrt{1-\bar{\alpha}_t}}$ is derived, the equivalency of (6) and (7) is also derived. To do this, start from the forward diffusion process of (2):

$$\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\phi(\boldsymbol{x}_t, t)$$

Substituting Tweedie's formula,

$$\begin{aligned}
\boldsymbol{x}_0 &= \frac{1}{\sqrt{\bar{\alpha}_t}}(\boldsymbol{x}_t + (1-\bar{\alpha}_t)\nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{x}_t)) \\
&= \frac{1}{\sqrt{\bar{\alpha}_t}}(\boldsymbol{x}_t + (1-\bar{\alpha}_t)s_\theta(\boldsymbol{x}_t, t)),
\end{aligned}$$

into the forward diffusion process:

$$\begin{aligned}
&\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\frac{1}{\sqrt{\bar{\alpha}_t}}(\boldsymbol{x}_t + (1-\bar{\alpha}_t)s_\theta(\boldsymbol{x}_t, t) + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\phi(\boldsymbol{x}_t, t) \\
\Rightarrow &(1-\bar{\alpha}_t)s_\theta(\boldsymbol{x}_t, t) = \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\phi(\boldsymbol{x}_t, t) \\
\Rightarrow &s_\theta(\boldsymbol{x}_t, t) = \frac{1}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\phi(\boldsymbol{x}_t, t)
\end{aligned}$$

which shows the formulations are the same.

■