# Image Processing using Diffusion Models

Won Sup Song

**Abstract**—Diffusion models have recently gained prominence as a powerful class of generative models capable of producing high-quality images through an iterative denoising process. In this project, we explore their application in two key areas: image generation and image restoration (inpainting and deconvolution). We first implement Denoising Diffusion Probabilistic Models (DDPMs) to generate images from noise, leveraging a stepwise denoising approach. For image restoration, we employ diffusion models as priors to solve inverse problems, comparing three different methods: SDEdit, a simple denoising-based approach; Score-ALD, which incorporates an additional correction step; and DPS, which refines posterior sampling for improved perceptual quality.
Our results demonstrate that diffusion models not only achieve state-of-the-art generative performance but also achives high performance in terms of both perceptual quality (LPIPS) and reconstruction accuracy (PSNR). Notably, DPS provides the most stable and high-quality reconstructions, whereas Score-ALD exhibits saturation artifacts, and SDEdit struggles to maintain fidelity to the ground truth. However, a key limitation is that these methods are highly dependent on the training dataset and may not generalize well to varying image sizes or unseen data distributions. Additionally, while DPS produces superior results, its computational cost remains a challenge for real-time applications.

**Index Terms**—Diffusion Model, DDPM, SDEdit, Score-ALD, DPS

## 1 Introduction

Diffusion models have emerged as a powerful generative framework for synthesizing high-quality images by **iteratively denoising noisy inputs**. Originally developed for image generation, these models have demonstrated exceptional performance in various applications, including **image restoration** tasks such as inpainting and deconvolution. Unlike traditional approaches, which often rely on explicit priors or handcrafted regularization, diffusion models learn to **reverse the noise addition process** through a probabilistic framework, enabling them to serve as strong generative priors for inverse problems.

This project explores two primary applications of diffusion models. First, we use **Denoising Diffusion Probabilistic Models (DDPMs)** [1] to generate high-quality images by sampling from noise and iteratively refining them through a learned denoising process. Second, we leverage pre-trained diffusion models to recover missing or degraded image information. Specifically, we implement and compare three methods for **inpainting and deconvolution**: **SDEdit** [2], an intuitive approach that refines noisy measurements through guided denoising; extbfScore-ALD [3], a more rigorous method that introduces an additional correction step; and **DPS** [4], a state-of-the-art technique that enhances posterior sampling to improve reconstruction fidelity.

Our goal is to analyze the effectiveness of diffusion models in both **generative and restorative tasks**, evaluating their performance using **quantitative metrics** such as **Peak Signal-to-Noise Ratio (PSNR) and Learned Perceptual Image Patch Similarity (LPIPS)**, as well as **qualitative visual comparisons**. Our findings show that diffusion models successfully generate high-quality images, with DDPMs demonstrating strong fidelity in the reverse process. For image restoration, DPS outperforms Score-ALD and SDEdit in terms of both perceptual and quantitative metrics, making it the most reliable approach. However, a key limitation is that diffusion models are trained on fixed-size images and may not generalize well to varying resolutions or unseen data distributions. Furthermore, while DPS provides high-quality restoration, its computational cost remains a challenge.

## 2 Related Work

Diffusion models have recently emerged as a powerful generative modeling framework, demonstrating state-of-the-art performance in **image synthesis, denoising, and inverse problem-solving**. Our work builds upon foundational research in diffusion models, particularly in **image generation and restoration**.

**Denoising Diffusion Probabilistic Models (DDPMs)** were first introduced by **Ho et al. (2020)** [1] as an alternative to traditional generative models like GANs and VAEs. Unlike GANs, which suffer from **mode collapse** and training instability, diffusion models **iteratively refine images from noise**, producing high-diversity, high-fidelity outputs. Further research has focused on improving sampling efficiency, such as **denoising diffusion implicit models (DDIMs)** [5], which reduce the number of steps required for high-quality image generation.

Several notable works have explored diffusion models in **image restoration**. **SDEdit** [2] is a simple method that refines noisy measurements through guided denoising. However, its reliance on mid-generation guidance often results in reconstructions that diverge from the original image structure. **Score-ALD** [3] introduces an additional **correction step** to improve reconstruction accuracy, but its estimation step causes small instability. **DPS** [4] develops a **posterior sampling framework** that integrates measurement consistency, leading to **state-of-the-art performance in image restoration tasks**.

- *W. Song is with the Department of Center for Global & Online Education, Stanford University, Stanford, CA, 94305.*
*E-mail: wsong2@stanford.edu*

# 3 Proposed Method

Our approach leverages **Denoising Diffusion Probabilistic Models (DDPMs)** for both **image generation and restoration**. The process consists of two steps:

1) **Forward Noising Process**: Gradually adds Gaussian noise to an image, corrupting its structure.
2) **Reverse Denoising Process**: The model learns to progressively remove noise, reconstructing the original image from a noisy sample.

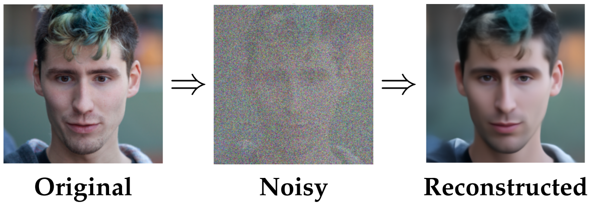

Fig. 1: Illustration of the full diffusion process: the forward step gradually adds noise, while the reverse step reconstructs the image.

## 3.1 Denoising Diffusion Probabilistic Model (DDPM)

Denoising Diffusion Probabilistic Models (DDPMs) are a class of generative models that iteratively denoise a Gaussian noise sample to generate realistic images.

The diffusion model consists of a **forward process** (adding noise) and a **reverse process** (denoising to reconstruct the original image). We visualize these processes using images with smaller, centered arrows positioned in the middle of the images for better alignment.

### *3.1.1 Forward Process - Noising*

The forward diffusion process is defined as:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{\alpha_t}x_{t-1}, (1-\alpha_t)I). \tag{1}$$

Expanding recursively over multiple steps:

$$x_t = \sqrt{\alpha_t}x_{t-1} + \sqrt{1-\alpha_t}\epsilon_{t-1}, \quad \epsilon_{t-1} \sim \mathcal{N}(0, I). \tag{2}$$

By iterating this process:

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, I) \tag{3}$$

where $\bar{\alpha}_t$ is the noise schedule. The model then denoises $x_t$ back to an estimate $\hat{x}_0$ using the learned score function.

### *3.1.2 Reverse Process - Denoising SDE*

To reverse the diffusion process, we use a learned neural network to predict the noise , defining the posterior:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)). \tag{4}$$

We derive the mean function:

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}t}}\epsilon\theta(x_t, t)\right). \tag{5}$$

This shows that noise prediction enables image recovery.

Finally, the DDPM algorithm performs iterative sampling from a Gaussian distribution noise, refining each step with a learned score function. Note that we will only explore the case for Variance Preserving (VP). The process is formalized in the following algorithm:

**DDPM Reverse Diffusion w/ Score Predictor**

```
    $\mathbf{x}_T \sim \mathcal{N}(0, I)$
    for $t = T, \ldots, 1$ do
        $\mathbf{z} \sim \mathcal{N}(0, I)$ if $t > 1$, else $\mathbf{z} = 0$
        $\hat{x}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t + (1-\bar{\alpha}_t)s_\theta(x_t, t))$
        $x_{t-1} = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)}{1-\bar{\alpha}_t}\hat{x}_0 + \sqrt{1-\alpha_t}\mathbf{z}$
    end for
return $\mathbf{x}_0$
```

In this work, we leverage **DDPMs** for **image generation** by applying the *reverse diffusion process* to random Gaussian noise. The reverse function of DDPM progressively refines the noise into a coherent image by iteratively denoising through learned distributions, ultimately reconstructing high-quality and realistic samples.

Beyond its generative capability, the **reverse function** of DDPM can also be employed for **image restoration**. Instead of starting from pure Gaussian noise, the process is initialized with a degraded or ”erroneous” image and refined through iterative denoising. By conditioning the diffusion process on corrupted inputs, DDPM can gradually recover the underlying structure of the image while reducing noise, artifacts, and missing details.

For **image restoration**, we implement three methods. **SDEdit** [2] applies a forward diffusion step to add noise to the input image before performing reverse diffusion to generate a refined output. **Score-ALD** [3] introduces an additional correction step to improve reconstruction accuracy by iteratively aligning with measurement constraints. **DPS** [4] enhances posterior sampling, ensuring higher perceptual quality and improved fidelity to the original image.

## 3.2 SDEdit

**SDEdit** [2] operates by applying a two-step process consisting of a forward diffusion step followed by a reverse diffusion step. Unlike Score-ALD and DPS, which explicitly take measurements to guide the diffusion process, SDEdit first corrupts the input image by adding noise using the forward diffusion process. Then, it applies the reverse diffusion process to reconstruct an image that adheres to the provided measurement data.

This method differs from traditional inverse problem-solving approaches as it does not incorporate iterative sampling or explicit likelihood optimization. Instead, it relies solely on the learned generative prior of the diffusion model to refine the corrupted image. While this approach allows for simple image editing and controlled modification of image content, it does not explicitly solve inverse problems in the same way as Score-ALD and DPS. Consequently, SDEdit often struggles with producing reconstructions that closely resemble the ground truth, particularly when the measurement data is significantly altered or sparse.

### 3.3 Score-ALD

**Score-ALD** [3] refines the denoising process by incorporating an additional correction step that ensures measurement consistency. The update rule at step $t$ is given by:

$$x_{t-1} = x_t - \frac{1}{\sigma_t^2 + \gamma_t^2} \nabla_{x_t} ||Ax_t - y||^2 \tag{6}$$

where $A$ is the degradation operator, $y$ is the observation, and $\sigma_t$ controls the noise level at step $t$. This additional gradient term enforces consistency between the estimated reconstruction and the measurement.

### 3.4 DPS

**DPS** [4] further refines posterior sampling by enforcing data fidelity more explicitly. The update step in DPS is:

$$x_{t-1} = x_t - \frac{\zeta_t}{2\sigma_t^2} \nabla_{x_t} ||A\hat{x}_0 - y||^2 \tag{7}$$

where $\zeta_t$ is a step size parameter. This method effectively balances diffusion-based denoising with data consistency, leading to improved image reconstructions.

## 4 Experimental Results

To evaluate the effectiveness of the proposed methods, we conduct experiments on image generation and image restoration tasks, focusing on inpainting and deconvolution. We use **Peak Signal-to-Noise Ratio (PSNR)** and **Learned Perceptual Image Patch Similarity (LPIPS)** as evaluation metrics. PSNR measures reconstruction fidelity, with higher values indicating better recovery of original images. LPIPS assesses perceptual similarity, where lower values correspond to images that are more visually similar to ground truth.

### 4.1 Forward Process  Reverse Process

As shown in Figure 2, it can be seen that as more noise is added, the reconstruction quality doesn't resemble the original image. This is evident from the PSNR values, which decrease as the timestep increases, and the LPIPS values, which increase as the timestep increases.

### 4.2 Image Generation

To demonstrate the effectiveness of the diffusion model in image generation, we visualize four generated samples obtained from different noise initializations using the DDPM sampling process. The diversity in these outputs highlights the generative capacity of the model. Results can be seen in Figure 3. It can be seen that for (256 × 256) size images, the reverse process from the DDPM (variance preserving) produces high-fidelity images from a complete noise image. From this, it can be concluded that the diffusion model has shown good results in image generation. Since there is no ground truth to compare against, PSNR and LPIPS values are not evaluated.

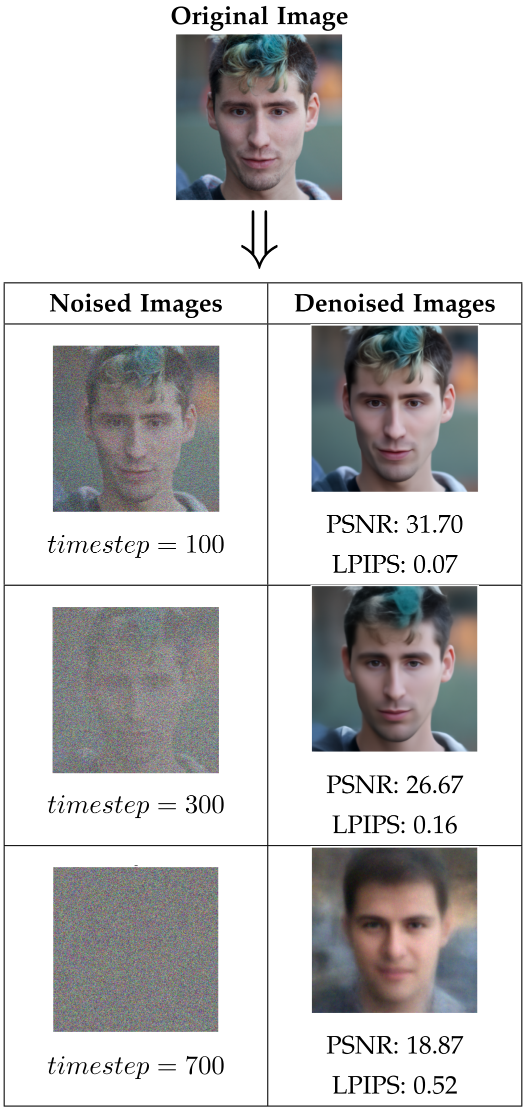

Fig. 2: Illustration of the full diffusion process: the left column shows noisy images at different timesteps, and the right column presents their corresponding denoised reconstructions with PSNR and LPIPS values. The original image is placed above with a downward arrow indicating the process.

### 4.3 Inpainting and Deconvolution Results

To evaluate the effectiveness of different image restoration methods in both inpainting and deconvolution, we present qualitative results in structured tables. Each row corresponds to a different method, while each column represents different evaluation metrics, including qualitative images and their corresponding PSNR and LPIPS values. Results can be seen in Figure 4.

It can be seen that by using "erroneous"images (for this experiment, images used for inpainting and deconvolution), they can serve as measurements that guide the diffusion model so that its gradient during training follows the measurement. SDEdit operates differently from Score-ALD and DPS in that it does not explicitly take measurements to guide the diffusion process. Instead, it first applies a forward diffusion step by adding noise to the input image and then

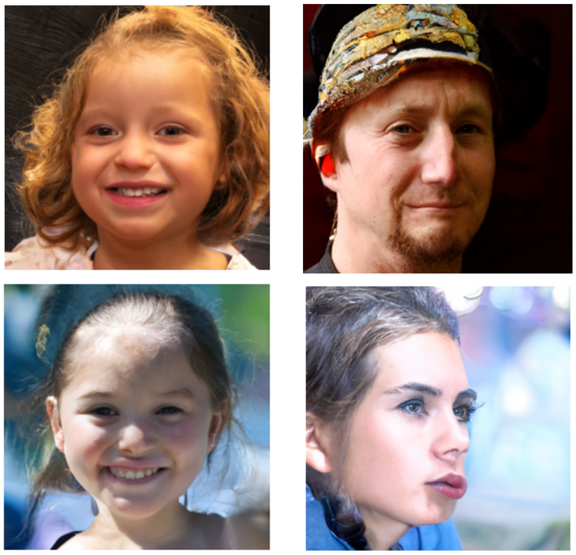

Fig. 3: Generated images using DDPM sampling from different noise initializations. The diversity in samples demonstrates the model's generative capabilities.

performs a reverse diffusion step using the measurement data as an initial condition. While this approach allows for controlled modifications of image content, it does not iteratively refine the image based on measurement constraints like Score-ALD and DPS.

As a result, the reconstructions produced by SDEdit tend to diverge from the ground truth compared to Score-ALD and DPS. Since the process is dependent on how much noise is added during the forward diffusion step, SDEdit struggles to fully restore missing information, particularly when the measurement is significantly altered. Unlike Score-ALD and DPS, which continuously refine their outputs through iterative sampling, SDEdit applies a single-shot reverse diffusion process, which may not always align with the underlying structure of the original image.

One notable observation in SDEdit is that when less noise is added during the forward diffusion step, the reconstructed image appears more similar to the input. This results in improved PSNR and LPIPS values but may retain unwanted artifacts and noise. Conversely, when more noise is added, the final image becomes more visually plausible but deviates further from the original, leading to a lower PSNR and higher LPIPS values.

Score-ALD and DPS resolve this issue by incorporating iterative sampling, leading to improved PSNR and LPIPS values compared to SDEdit. However, Score-ALD is an approximation of the log-likelihood term introduced in the update function, leading to a lower reconstruction quality compared to the DPS method. Score-ALD also has higher saturation compared to DPS, which produces great image quality. This is because DPS directly utilizes the score function (gradient of the log-likelihood) of the diffusion model to iteratively refine noisy data into a more plausible clean sample. Score-ALD, on the other hand, introduces adversarial training in latent space, which can lead to instability during training and inference. DPS avoids the need for adversarial optimization, making it more stable and easier to train.

It should be noted that DPS works for linear and nonlinear inverse problems with Gaussian or Poisson noise, so it can be expected that DPS performance would be good for other images with Poisson noise.

## 5 Discussion

The experimental results highlight the strengths and limitations of different diffusion-based methods in image generation and restoration. The results from the forward and reverse processes indicate that as noise is added, image reconstruction becomes more challenging, as evidenced by decreasing PSNR and increasing LPIPS values. This aligns with the expectation that greater noise levels lead to significant degradation of perceptual and structural information.

In terms of image generation, the diffusion model demonstrates strong generative capabilities, producing high-fidelity images from pure noise. The ability of the model to generate diverse outputs from different noise initializations underscores its effectiveness. However, since there is no ground truth for generated images, traditional quantitative metrics like PSNR and LPIPS are not applicable, and subjective evaluation is necessary.

For image restoration, the comparison between SDEdit, Score-ALD, and DPS reveals crucial differences in their approaches and performance. SDEdit's method of introducing measurements mid-way through the generation process results in images that do not closely resemble the ground truth. On the other hand, Score-ALD and DPS incorporate measurements from the beginning, allowing for a more controlled restoration process. The observed differences in PSNR and LPIPS values confirm that DPS is the most stable and effective method, as it avoids adversarial training and iteratively refines the image while solving inverse problems more effectively.

Despite the promising results, there are limitations to consider. The generated and restored images are highly dependent on the training data, limiting generalization to images of different resolutions, structures, or domains outside the training set. Specifically, the methods work well for the trained image sizes but may not generalize effectively to larger or smaller resolutions. Furthermore, while DPS provides improved restoration quality, its computational cost is higher due to iterative refinement. Additionally, adversarial-based methods like Score-ALD introduce instability, making them less reliable in certain cases. Lastly, while diffusion models show great promise in image restoration, their reliance on iterative denoising makes real-time applications challenging, necessitating future work on efficiency improvements.

## 6 Conclusion

The findings of this study demonstrate the potential of diffusion models in both image generation and restoration tasks. While the diffusion model effectively generates high-fidelity images from pure noise, its application in restoration depends on how the measurement constraints are integrated. DPS emerges as the most reliable approach, outperforming SDEdit and Score-ALD in terms of both stability and reconstruction quality.

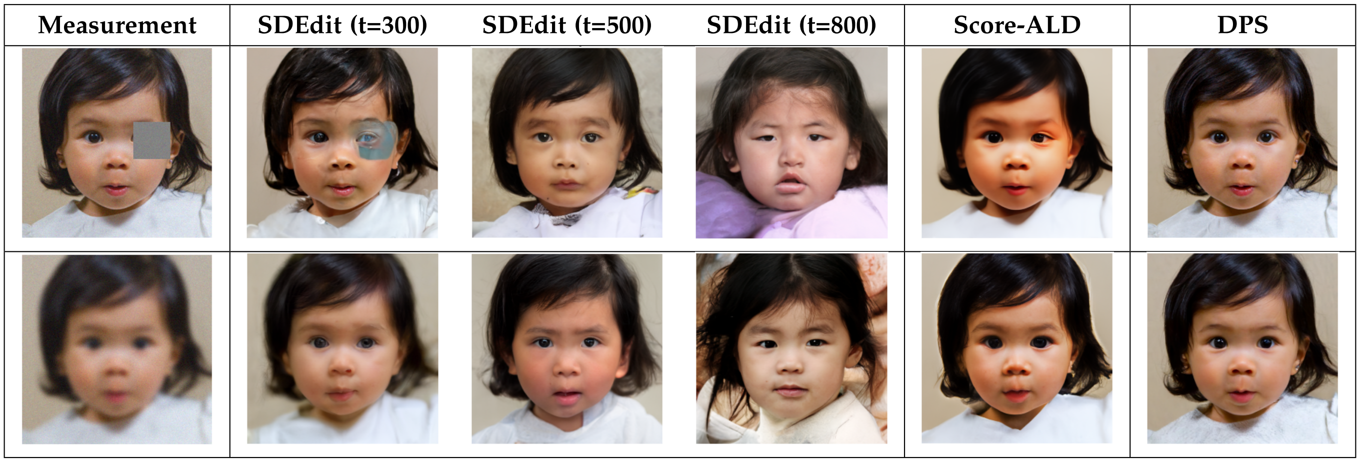

Fig. 4: Comparison of inpainting and deconvolution results. The first row represents inpainting, where degraded images (leftmost column) are restored using various methods. The second row represents deconvolution, restoring blurred images through different approaches. Each column showcases a different method: SDEdit at different noise levels (t=300, 500, 800), Score-ALD, and DPS.

| Metric | SDEdit (t=300) | SDEdit (t=500) | SDEdit (t=800) | Score-ALD | DPS |
|---|---|---|---|---|---|
| **Inpainting PSNR** | 22.76 | 20.72 | 14.05 | 22.75 | 34.93 |
| **Inpainting LPIPS** | 0.14 | 0.18 | 0.32 | 0.12 | 0.02 |
| **Deconvolution PSNR** | 24.45 | 22.34 | 13.32 | 22.49 | 28.36 |
| **Deconvolution LPIPS** | 0.21 | 0.20 | 0.37 | 0.14 | 0.07 |

Fig. 5: Comparison of inpainting and deconvolution results.

Future work could explore further refinements in diffusion-based restoration techniques, especially in handling different types of noise and extending these methods to more complex image structures. Additionally, addressing generalization issues by training models on diverse datasets and varying image sizes could improve adaptability. Lastly, optimizing the computational efficiency of diffusion-based approaches will be crucial for real-time and large-scale applications.

## Acknowledgments

The author would like to thank the teaching staff of EE367 for designing the project outline and the the groundwork that guided this diffusion project.

## References

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

[2] C. Meng, Y. He, J. Song, Y. Song, J. Wu, J.-Y. Zhu, and S. Ermon, "Sdedit: Guided image synthesis and editing with stochastic differential equations," *Advances in Neural Information Processing Systems*, vol. 35, pp. 15 831–15 843, 2022.

[3] A. Jalal, M. Arvinte, G. Daras, E. Price, A. G. Dimakis, and J. I. Tamir, "Robust compressed sensing mri with deep generative priors," *arXiv preprint arXiv:2102.08471*, 2021.

[4] H. Chung, J. Kim, M. T. McCann, S. Klasky, and J. C. Ye, "Diffusion posterior sampling for general noisy inverse problems," *International Conference on Learning Representations (ICLR)*, 2023.

[5] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," *arXiv preprint arXiv:2010.02502*, 2020.

## Extra Information

### Part 1: Forward Diffusion Process

**Proposition 1:** Given the forward diffusion process:

$$X_t = \sqrt{1-\beta_t}X_{t-1} + \sqrt{\beta_t}Z_{t-1}, \quad Z_{t-1} \sim \mathcal{N}(0, I)$$

Prove that this can be rewritten as:

$$X_t = \sqrt{\bar{\alpha}_t}X_0 + \sqrt{1-\bar{\alpha}_t}Z$$

**Proof:** The forward diffusion process is defined as:

$$X_t = \sqrt{1-\beta_t}X_{t-1} + \sqrt{\beta_t}Z_{t-1}, \quad t = 1, 2, \ldots, T \tag{8}$$

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

We define:

$$\alpha_t = 1 - \beta_t, \quad \bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i. \tag{9}$$

Rewriting the equation recursively:

$$X_t = \sqrt{\alpha_t}X_{t-1} + \sqrt{1-\alpha_t}Z_{t-1}, \quad t = 1, 2, \ldots, T. \tag{10}$$

Expanding further:

$$X_t = \sqrt{\alpha_t}\left(\sqrt{\alpha_{t-1}}X_{t-2} + \sqrt{1-\alpha_{t-1}}Z_{t-2}\right) + \sqrt{1-\alpha_t}Z_{t-1} \tag{11}$$

$$= \sqrt{\alpha_t\alpha_{t-1}}X_{t-2} + \sqrt{1-\alpha_t}Z_{t-1} + \sqrt{\alpha_t(1-\alpha_{t-1})}Z_{t-2}. \tag{12}$$

Following the recursive pattern:

$$\begin{aligned} X_t = {} & \sqrt{\alpha_t \alpha_{t-1} \alpha_{t-2}} X_{t-3} \\ & + \sqrt{1-\alpha_t} Z_{t-1} + \sqrt{\alpha_t (1-\alpha_{t-1})} Z_{t-2} \\ & + \sqrt{\alpha_t \alpha_{t-1} (1-\alpha_{t-2})} Z_{t-3}. \end{aligned} \tag{13}$$

Generalizing, we obtain:

$$X_t = \sqrt{\prod_{i=1}^{t} \alpha_i} X_0 + \sum_{i=1}^{t} \left( \sqrt{1-\alpha_i} \prod_{j=i+1}^{t} \sqrt{\alpha_j} \right) Z_{i-1}. \tag{14}$$

Since $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$, this simplifies to:

$$X_t = \sqrt{\bar{\alpha}_t} X_0 + \sqrt{1-\bar{\alpha}_t} Z. \tag{15}$$

## Part 2: Equivalence of Two Reverse Diffusion Forms

**Proposition 2:** Prove that:

$$\hat{X}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}} (X_t + (1-\bar{\alpha}_t) \delta_\theta(X_t, t))$$

is equivalent to:

$$X_{t-1} = \frac{1}{\sqrt{\alpha_t}} (X_t + (1-\alpha_t) \delta_\theta(X_t, t))$$

and:

$$X_{t-1} = \frac{\sqrt{\alpha_t (1-\bar{\alpha}_{t-1})}}{1-\bar{\alpha}_t} X_t + \frac{\sqrt{\bar{\alpha}_{t-1} (1-\alpha_t)}}{1-\bar{\alpha}_t} \hat{X}_0$$

**Proof:**

We start with the estimated clean sample:

$$\hat{x}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}} \left( X_t + (1-\bar{\alpha}_t) S_\theta(X_t, t) \right). \tag{16}$$

Next, we define the expression for $X_{t-1}$:

$$X_{t-1} = \frac{\sqrt{\alpha_t (1-\bar{\alpha}_{t-1})}}{1-\bar{\alpha}_t} X_t + \frac{\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)}{1-\bar{\alpha}_t} \hat{x}_0. \tag{17}$$

Expanding $\hat{x}_0$:

$$\begin{aligned} X_{t-1} = {} & \frac{\sqrt{\alpha_t (1-\bar{\alpha}_{t-1})}}{1-\bar{\alpha}_t} X_t \\ & + \frac{\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)}{\sqrt{\bar{\alpha}_t (1-\bar{\alpha}_t)}} \left( X_t + (1-\bar{\alpha}_t) S_\theta(X_t, t) \right). \end{aligned} \tag{18}$$

Rewriting:

$$\begin{aligned} X_{t-1} = {} & \left( \frac{\sqrt{\alpha_t (1-\bar{\alpha}_{t-1})}}{1-\bar{\alpha}_t} + \frac{\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)}{\sqrt{\bar{\alpha}_t (1-\bar{\alpha}_t)}} \right) X_t \\ & + \frac{\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)}{\sqrt{\bar{\alpha}_t (1-\bar{\alpha}_t)}} S_\theta(X_t, t). \end{aligned} \tag{19}$$

Since:

$$\bar{\alpha}_t = \alpha_t \bar{\alpha}_{t-1}, \tag{20}$$

we substitute:

$$\begin{aligned} X_{t-1} = {} & \left( \frac{\alpha_t (1-\bar{\alpha}_{t-1}) + (1-\alpha_t)}{\sqrt{\alpha_t (1-\bar{\alpha}_t)}} \right) X_t \\ & + \frac{(1-\alpha_t)}{\sqrt{\bar{\alpha}_t}} S_\theta(X_t, t). \end{aligned} \tag{21}$$

Simplifying further:

$$X_{t-1} = \frac{1-\bar{\alpha}_t}{\sqrt{\alpha_t (1-\bar{\alpha}_t)}} X_t + \frac{(1-\alpha_t)}{\alpha_t} S_\theta(X_t, t). \tag{22}$$

Thus, we can express the function:

$$X_{t-1} = \frac{1}{\sqrt{\alpha_t}} X_t + \frac{(1-\alpha_t)}{\alpha_t}. \tag{23}$$

## Part 3: Equivalence of Reverse Diffusion and Algorithm 2 in DDPM

**Proposition 3:** Prove that:

$$X_{t-1} = \frac{1}{\sqrt{\alpha_t}} (X_t + (1-\alpha_t) \delta_\theta(X_t, t))$$

is equivalent to:

$$X_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( X_t - \frac{(1-\alpha_t)}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(X_t, t) \right)$$

**Proof:**

The forward process is given by:

$$X_t = \sqrt{\bar{\alpha}_t} X_0 + \sqrt{1-\bar{\alpha}_t}\, \epsilon(X_t, t). \tag{24}$$

The estimate of $X_0$ is:

$$\hat{X}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}} \left( X_t + (1-\bar{\alpha}_t) S_\theta(X_t, t) \right). \tag{25}$$

Rearranging for $\epsilon(X_t, t)$:

$$\epsilon(X_t, t) = \frac{X_t - \sqrt{\bar{\alpha}_t} X_0}{\sqrt{1-\bar{\alpha}_t}}. \tag{26}$$

Substituting $X_0$ from $\hat{X}_0$:

$$\epsilon(X_t, t) = X_t - \sqrt{\bar{\alpha}_t} \left( \frac{1}{\sqrt{\bar{\alpha}_t}} X_t + \frac{1-\bar{\alpha}_t}{\sqrt{\bar{\alpha}_t}} S_\theta(X_t, t) \right). \tag{27}$$

Expanding:

$$\epsilon(X_t, t) = X_t - \frac{X_t + (1-\bar{\alpha}_t) S_\theta(X_t, t)}{\sqrt{1-\bar{\alpha}_t}}. \tag{28}$$

Simplifying:

$$\epsilon(X_t, t) = -\sqrt{1-\bar{\alpha}_t} S_\theta(X_t, t). \tag{29}$$

Thus, solving for $S_\theta(X_t, t)$:

$$S_\theta(X_t, t) = -\frac{\epsilon(X_t, t)}{\sqrt{1-\bar{\alpha}_t}}. \tag{30}$$