Solving Inverse Problems With Diffusion Models and Methods

Benjia Zhang and Sylvia Chin

Abstract—There is often a strong need for clear, high resolution image reconstruction from a corrupt source. To achieve a cleaned image from degradation without using unique trained models for each, we can use a universal, pretrained diffusion model prior. We explore three different techniques with the diffusion model: SDEdit (Score-Distillation Editing), ScoreALD, and DPS (Diffusion Posterior Sampling). Both qualitative and quantitative analysis (PSNR and Learned Perceptual Image Patch Similarity (LPIPS)) are critical to evaluating the success of these widely applicable techniques. Our findings indicate that Diffusion Posterior Sampling produce the most successful PSNR (high) and LPIPS (low) rates, demonstrating the effectiveness of the using the predictive image results with the pretrained diffusion model to effectively recover an image from denegration.

Index Terms—Computational Imaging, Diffusion Model

1 Introduction

Computational imaging addresses the challenge of recovering images from various types of image denigration. The representations of these corruptions can be formalized as linear inverse problems, such as image inpainting to recover missing pixels or deconvolution to remove blur. Former work on image recovery has involved training completely separate, task-specific neural networks for every distinct type of degradation. This is an inefficient practice across different forms of corruption, and unreasonably expensive at scale. Diffusion models are a nascent solution to the same issues but with much less computation. They use generative priors that eliminate the need for task-specific training. Because a diffusion model naturally learns the through a forward noising process, we can guide its reverse denoising process using our known measurement models. Therefore, we can shape image generation toward a widely applicable solution that is reasonably successful in image reconstruction.

In this project, we evaluate and compare three increasingly rigorous posterior sampling techniques that leverage a single pre-trained unconditional diffusion model: Score-Distillation Editing (SDEdit) [1], Score Approximate Langevin Dynamics (ScoreALD) [2], and Diffusion Posterior Sampling (DPS) [3]. While baseline methods like SDEdit act as unconstrained image-to-image translators, and ScoreALD introduces high-frequency artifacts by evaluating measurements on highly noisy intermediate states, we demonstrate that DPS resolves these limitations. By calculating normalized likelihood gradients on the network's prediction of the clean image, DPS enforces strict measurement consistency. Through quantitative and qualitative analysis across inpainting and deconvolution tasks, we highlight the effectiveness of DPS in producing seamless, state-of-the-art reconstructions.

2 Related Work

Diffusion models have recently become a powerful framework for generative modeling and image reconstruction tasks. A foundational approach is the Denoising Diffusion Probabilistic Model (DDPM) proposed by [4], which learns to generate images by reversing a gradual Gaussian noising process. During training, the model learns to predict and remove noise at each timestep, enabling high-quality image synthesis by iteratively denoising random noise samples. Beyond unconditional generation, DDPM provides the core mechanism underlying many diffusion-based approaches to image restoration and inverse problems.

Several works have extended diffusion models to incorporate measurement information for solving inverse problems. [2] demonstrated that deep generative priors, including score-based models, can effectively regularize ill-posed reconstruction tasks such as compressed sensing MRI. Their work highlights the potential of learned generative priors to replace traditional hand-crafted regularizers in imaging problems with incomplete or noisy observations.

SDEdit, introduced by [1], adapts diffusion models for controlled image editing and restoration. Instead of starting from pure noise, SDEdit perturbs a given input image with noise and then applies the reverse diffusion process to generate a realistic sample that remains consistent with the original input. This strategy enables guided synthesis and has been applied to tasks such as inpainting and image refinement.

More recent methods explicitly incorporate measurement consistency during the diffusion sampling process. Score Approximate Langevin Dynamics (ScoreALD) combines the score function learned by the diffusion model with gradients derived from the measurement likelihood, guiding the sampling process toward solutions that satisfy both the learned data prior and the observation model. Building on this idea, Diffusion Posterior Sampling (DPS) [3] formulates reconstruction as approximate posterior sampling under a diffusion prior. DPS integrates likelihood-based conditioning into the reverse diffusion process, allowing the model to handle noisy measurements and produce reconstructions that better satisfy the forward measurement model.

Together, these methods represent a progression from unconditional diffusion generation (DDPM) to increasingly measurement-aware reconstruction approaches (SDEdit, ScoreALD, and DPS), motivating comparative evaluation of their effectiveness on inverse imaging tasks.

3 Proposed Method

3.1 Baseline DDPM Sampling

Before applying diffusion models to reconstruct specific corrupted measurements, it's important to understand how they generate images from pure noise. This baseline procedure, known as unconditional image generation, follows the standard Denoising Diffusion Probabilistic Model (DDPM) framework [4]. DDPM sampling performs iterative denoising. The process starts with a completely random image composed of Gaussian noise x_t. Over a sequence of discrete timesteps (from T down to 1), a pre-trained neural network processes this noisy input. Rather than directly predicting the final image, the network estimates a score: a vector field pointing toward the natural image manifold.

Using this score and a statistical relation known as Tweedie's formula, the model infers the estimated clean image $\hat{x}_0$ at each step. It then removes a small, controlled portion of noise to produce a slightly denoised intermediate sample x_{t-1}. Repeating this refinement over hundreds of steps gradually transforms pure noise into a high-quality image.

Since this process has no conditioning information, we call it "unconditional": the model generates a random but realistic image, such as a plausible human face. However, this iterative denoising loop forms the basis for all reconstruction methods studied in this work. By intervening in specific steps of the denoising process, we can guide the generation to align with our measured data, turning a random image synthesizer into a powerful framework for computational imaging.

3.2 SDEdit

To evaluate the effectiveness of the SDEdit method, we conduct experiments on images with a resolution of 256×256. Our goal is to assess the ability of SDEdit to perform both *inpainting* and *deconvolution* under different types of image corruption.

Before passing an image into the SDEdit pipeline, we apply one of three corruption operations to the original image:

1) **Square masking:** A 50×50 square region of the image is masked.
2) **Random blotch masking:** Approximately 50% of the image is masked using randomly shaped blotches.
3) **Gaussian blurring:** The image is blurred using a Gaussian kernel to simulate degradation from convolutional distortion.

After applying one of these corruption processes, we add Gaussian noise corresponding to diffusion step t, as shown in (1). The noisy image is then passed through the SDEdit procedure, which iteratively denoises the sample to reconstruct a clean image. This experimental setup allows us to evaluate how effectively SDEdit can recover the underlying image structure from partially missing or degraded inputs.

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

3.3 ScoreALD

Although SDEdit works reasonably well, it is mostly an unconstrained generative method and does not strictly enforce agreement with the observed measurements. To address this, we consider Score Approximate Langevin Dynamics (ScoreALD) [2], which directly injects measurement information into the reverse diffusion process.

ScoreALD alternates between a standard unconditional DDPM denoising step and a gradient descent update based on the measurement likelihood. At each timestep t, it applies the forward degradation operator $\mathcal{A}$ to the current *noisy* state x_t, then computes the gradient of the squared L_2 distance between this degraded version and the observed measurement y.

The modified update rule is:

$$x_{t-1} = x'_{t-1} - \frac{1}{2(\sigma^2 + \gamma_t^2)} \nabla_{x_t} \left\| \mathcal{A}(x_t) - y \right\|^2,$$

where x'_{t-1} is the unconditional DDPM update, σ is the measurement noise standard deviation, and γ_t is the diffusion noise level at timestep t. The parameter σ models the assumed noise level in the degraded observation and controls how strongly the gradient update enforces measurement consistency. Intuitively, a larger σ reduces the effective step size, encouraging the model to trust its learned prior instead of overfitting to unreliable, heavily corrupted pixels. We set the step size using a simple linear annealing schedule for stability.

This extra gradient step nudges the diffusion trajectory toward states that match the measurements. However, a key drawback is that $\mathcal{A}$ is applied to the very noisy state x_t instead of a clean image. For measurements with sharp boundaries (ex. the square masks in our inpainting task), this can push the U-Net away from the natural image manifold and produce high-frequency artifacts, such as diagonal banding or colored speckle, as it tries to satisfy the hard measurement constraint.

3.4 DPS

To reconstruct clean images from degraded observations, we employ DPS, a diffusion-based framework for solving inverse problems. DPS formulates image reconstruction as sampling from the posterior distribution of clean images conditioned on the observed corrupted image. In this setting, the corruption process is modeled as a measurement operator applied to the clean image, and the goal is to recover samples that are both consistent with the observation and lie on the natural image manifold learned by the diffusion model. DPS achieves this by modifying the reverse diffusion process. At each denoising step, the diffusion model generates a proposal for a cleaner image, while an additional gradient term derived from the likelihood of measurement encourages consistency with the observed degraded image, shown in (2). This guidance allows the

sampling process to balance two objectives: enforcing realism through the learned diffusion prior and satisfying the constraints imposed by the observation.

$$x_{t-1} = x_{t-1} - \zeta_t \nabla_{x_t} \|\mathcal{A}(x_t) - y\|^2 \tag{2}$$

4 Experimental Results

4.1 Task 1

Below we present the derivations of the forward noising process, the reverse denoising process, and the training objective used in DDPM to learn a noise-predicting network ϵ_θ. More detailed handwritten calculations can be found in the appendix at the end of the document.

4.1.0.1 1. Forward Noise Modeling:

$$x_t = \sqrt{1-\beta_t}\, x_{t-1} + \sqrt{\beta_t}\, z_{t-1}, \quad t = 1, 2, \ldots, T$$

$$z_{t-1} \sim \mathcal{N}(0, I)$$

We want to prove

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

where

$$\alpha_t = 1 - \beta_t, \quad \bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i, \quad z \sim \mathcal{N}(0, I).$$

Expanding recursively,

$$x_1 = \sqrt{1-\beta_1} x_0 + \sqrt{\beta_1} z_0$$

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

$$= \sqrt{1-\beta_2}\sqrt{1-\beta_1} x_0 + \sqrt{1-\beta_2}\sqrt{\beta_1} z_0 + \sqrt{\beta_2} z_1$$

Continuing this expansion gives

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sum_{j=1}^{t} \sqrt{\beta_j} \left(\prod_{i=j+1}^{t} \sqrt{\alpha_i} \right) z_{j-1}.$$

Since z_i are i.i.d., the second term can be written as a single Gaussian:

$$\sum_{j=1}^{t} c_j z_{j-1} \sim \sqrt{\sum_{j=1}^{t} c_j^2}\, z$$

where

$$c_j^2 = \beta_j \prod_{i=j+1}^{t} \alpha_i.$$

Then

$$\sum_{j=1}^{t} c_j^2 = \sum_{j=1}^{t} \beta_j \prod_{i=j+1}^{t} \alpha_i = 1 - \bar{\alpha}_t.$$

Thus

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

4.1.0.2 2. Reverse Process:

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

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

Substituting $\hat{x}_0$ gives

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

4.1.0.3 3. Score Prediction: DDPM trains a network $\epsilon_\theta(x_t, t)$ to predict the noise:

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t} z$$

$$x_0 = \frac{1}{\sqrt{\bar{\alpha}_t}} \left(x_t - \sqrt{1-\bar{\alpha}_t} z \right)$$

Since

$$\epsilon_\theta(x_t, t) \approx z,$$

we obtain

$$\hat{x}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}} \left(x_t - \sqrt{1-\bar{\alpha}_t}\, \epsilon_\theta(x_t, t) \right).$$

Matching this with the score-based formulation gives

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

Thus the update becomes

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

4.2 Task 2

Before using the diffusion model to solve complex inverse problems, we first tested its fundamental ability to remove noise from an image [4]. In this task, we took a clean reference image (x_0) and artificially corrupted it by adding Gaussian noise to simulate a specific intermediate diffusion timestep (x_t).

We then passed this noisy image through our pre-trained network. Because the model is trained to predict the "score" (a vector pointing toward the manifold of clean data), we can use Tweedie's formula to estimate the original, uncorrupted image in a single mathematical step.

As shown in Figure 1, the model successfully strips away the heavy static to reveal a clear and accurate prediction of the original face. This confirms that the network has a strong baseline understanding of natural image structures, which is required before we can attempt to guide it with complex measurements.

Fig. 1. Single-Step Denoising Result

4.3 Task 3

Building on the single-step denoising from the Task 2, we now use the diffusion model to generate entirely new images from scratch. Instead of starting from a corrupted photograph, unconditional generation begins from a tensor of pure Gaussian noise (x_T). From timestep T down to 0, we repeatedly pass this noisy state through the U-Net. At each step, the model predicts a score that estimates the clean image, we move a small step toward this estimate, and we re-inject a carefully scaled amount of noise to preserve the correct data distribution. After hundreds of these reverse diffusion steps, the model transforms random static into a coherent image. As shown in Figure 2, this process allows the network to synthesize highly realistic and diverse human faces. Because there is no measurement constraint, these faces are not tied to any real input; they are purely hallucinated from the model's learned prior. This illustrates the model's raw generative capability and provides a baseline before we introduce constraints for inverse problem tasks, covered in later Tasks.

Fig. 2. Unconditional Image Generation Result

4.4 Task 4

As shown in Figure 3, SDEdit produces visually coherent reconstructions and is able to perform both inpainting and deconvolution effectively. However, the reconstructed images often differ noticeably from the ground truth. For example, in the image containing the young girl, attributes such as hair color, facial features, background colors, and clothing are altered in the reconstruction. This illustrates that while sampling from a strong diffusion prior generates realistic images, it does not explicitly enforce consistency with the original measurement. Among the corruption types we evaluated, the random mask inpainting case performed the worst both qualitatively and quantitatively. This is expected, as the masking procedure removes approximately 50% of the image, leaving the model with significantly less information to guide the reconstruction.

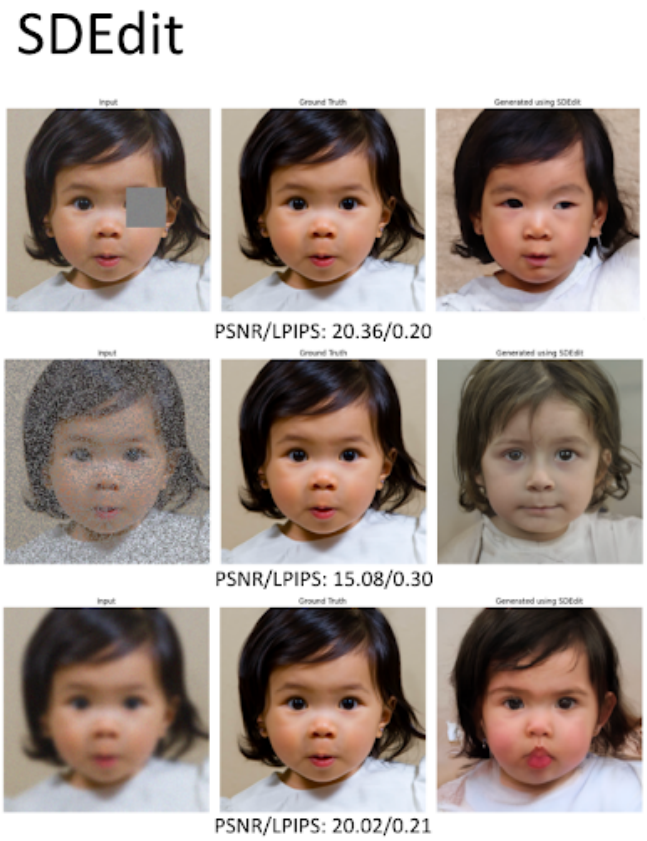

Fig. 3. SDEdit Result

4.5 Task 5

To address the lack of measurement constraints in SDEdit, we apply Score Approximate Langevin Dynamics (ScoreALD) to two inverse problems: image inpainting with a square mask and deconvolution of Gaussian blur. By inserting gradient-based updates into the denoising loop, ScoreALD explicitly pushes the generated image to agree with the known, uncorrupted pixels. As shown in Figure 4, this produces a coherent face that better matches the original background than the SDEdit baseline. The qualitative evaluation aligns as well, with a higher PSNR than the previous method.

A critical challenge in implementing ScoreALD is the extreme sensitivity of the gradient update step. The mathematical step size is governed by the measurement noise parameter, σ, and the diffusion noise level, which we control empirically via an annealing factor. Because the likelihood gradients are computed on unconstrained, noisy intermediate states, the gradients are prone to exploding. To achieve mathematical stability, the annealing factor must be strictly tuned to the specific forward operator $\mathcal{A}$. For deconvolution, we linearly annealed this factor from 10 to 15 across the diffusion timesteps. In contrast, inpainting, which imposes harsh, binary measurement boundaries, required a stronger annealing schedule of 15 to 20 to prevent the pixel values from collapsing.

These results and tuning requirements perform decently, but there are still drawbacks. Since ScoreALD computes its measurement gradients on very noisy intermediate states (x_t), the hard constraints interact poorly with the noise. Even with perfectly tuned annealing schedules, the network responds by hallucinating high-frequency artifacts, clearly visible as squiggly, static-like patterns inside the inpainted box. Although the deconvolution example recovers sharper detail, the instability introduced by enforcing measurements on noisy states perfectly motivates the more robust approach developed in Task 6.

These results, however, also expose the main weakness of ScoreALD. Because ScoreALD computes its measurement gradients on very noisy intermediate states (x_t), the hard constraints interact poorly with the noise. The network responds by hallucinating strong high-frequency artifacts, clearly visible as squiggly, static-like patterns inside the inpainted box. Although the deconvolution example recovers sharper detail, the instability introduced by enforcing measurements on noisy states motivates the more robust approach developed in Task 6.

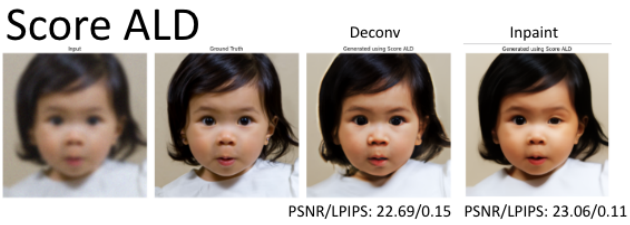

Fig. 4. ScoreALD Results: Box Deconvolution and Inpaint

4.6 Task 6

Compared to the previous task experimenting with ScoreALD, Task 6 uses very similar technique to use a gradient to guide the forward pass image closer and closer to the measurement. However, the difference here is that we forward pass the predicted clean image $\hat{x}_0$ rather than x_t from each iteration. This inputs a cleaner and more realistic "original" image so the overall process does a better job, as shown in Figure 5.

5 Discussion

The experimental results demonstrate a clear progression in reconstruction quality as increasingly measurement-aware diffusion methods are introduced. While the baseline DDPM sampling procedure confirms the model's ability to generate realistic images from noise, it lacks any mechanism to enforce consistency with observed measurements. Methods such as SDEdit partially address this limitation by conditioning the diffusion process on corrupted inputs, producing visually plausible reconstructions. However, because SDEdit primarily relies on the learned generative prior, it does not guarantee that the reconstructed image strictly matches the observed measurements.

ScoreALD improves upon SDEdit by explicitly incorporating measurement gradients into the diffusion process. This allows the model to better satisfy the forward degradation model and generally improves quantitative metrics such as PSNR. However, our experiments reveal a key limitation: because measurement gradients are applied to highly noisy intermediate states, the algorithm can introduce instability and visible high-frequency artifacts. In addition, the scaling coefficient used to control the strength of the measurement gradient plays a critical role in the reconstruction process. Small changes to this coefficient can significantly affect stability and image quality, requiring careful tuning depending on the degradation operator and noise level.

DPS addresses these issues by applying measurement guidance to the predicted clean image rather than the noisy intermediate state. By operating on a cleaner estimate of the image during each denoising step, DPS achieves a better balance between measurement consistency and adherence to the natural image prior. This approach reduces the instability observed in ScoreALD and produces reconstructions that are both visually coherent and quantitatively superior, achieving higher PSNR and lower LPIPS scores across the evaluated tasks.

Overall, these results highlight two important considerations for diffusion-based inverse problem solvers: the stage at which measurement information is incorporated and the scaling of the measurement guidance term. While generative priors provide powerful regularization for ill-posed imaging problems, improper gradient scaling or guidance applied to highly noisy states can degrade performance. Methods such as DPS that guide the diffusion trajectory using cleaner image estimates, together with carefully tuned guidance scaling, provide a more stable and effective framework for computational imaging tasks such as inpainting and deconvolution.

6 Conclusion

Computational imaging needs methods that can recover high-quality images from corrupted measurements without retraining a new, separate model for every task. In this work, we showed that a single pre-trained unconditional diffusion model can act as a universal prior for linear inverse problems such as inpainting and deconvolution. By comparing three sampling strategies, we observed a clear improvement in both reconstruction quality and stability.

SDEdit produced visually appealing images but behaved mostly like an unconstrained generator, often changing important details and ignoring strict measurement fidelity. ScoreALD better enforced the measurements using likelihood gradients, yet its use of very noisy intermediate states made it extremely sensitive to hyperparameters and prone to high-frequency artifacts. In contrast, Diffusion Posterior Sampling (DPS) addressed these issues by applying gradients to the model's clean-image prediction, resulting in higher PSNR and lower LPIPS as well as clear visual improvement. Our qualitative figures and PSNR/LPIPS scores both show that overall, DPS strikes the right balance between the generative prior and measurement consistency, yielding a stable and effective framework for image reconstruction.

Acknowledgments

We want to thank Professor Gordon Wetzstein and Sonia for their precious time, guidance, and help throughout the

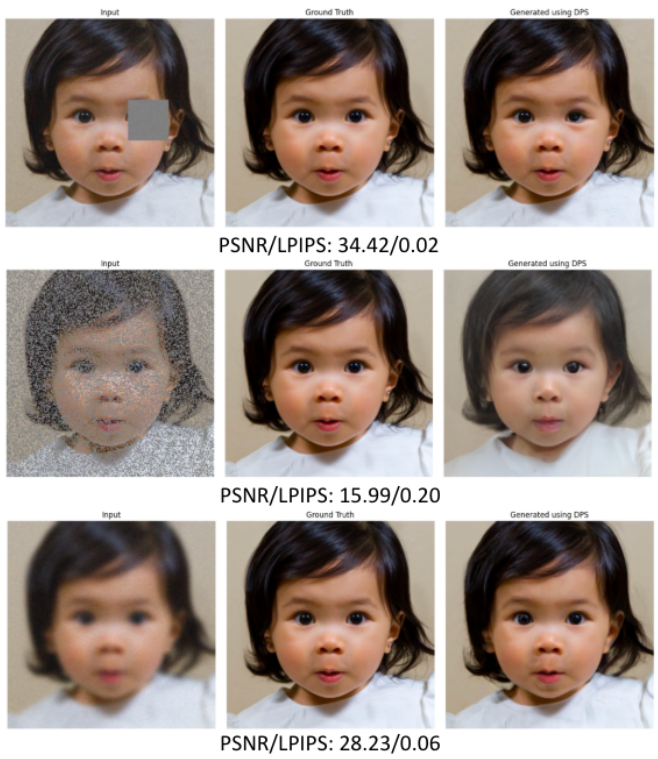

Fig. 5. DPS Result

quarter and during this project.

REFERENCES

[1] 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.

[2] 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 Workshop / arXiv preprint*, 2021.

[3] 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.

[4] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., 2020, pp. 6840–6851. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf

Appendix: Handwritten Derivations for Task 1

1.) Forward noise modeling:

$$x_t = \sqrt{1-\beta_t}\, x_{t-1} + \sqrt{\beta_t}\, z_{t-1} \qquad t = 1, 2, \ldots, T$$

$z_{t-1} \sim N(0, I)$

Prove $x_t = \sqrt{1-\beta_t}\, x_{t-1} + \sqrt{\beta_t}\, z_{t-1} = \sqrt{\tilde{\alpha}_t}\, x_0 + \sqrt{1-\tilde{\alpha}_t}\, z$

$$\alpha_t = 1 - \beta_t \;, \quad \tilde{\alpha}_t = \prod_{i=1}^{t} \alpha_i \;, \quad z \sim N(0, I)$$

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

$$x_1 = \sqrt{1-\beta_1}\, x_0 + \sqrt{\beta_1}\, z_0$$

$$x_2 = \sqrt{1-\beta_2}\, x_1 + \sqrt{\beta_2}\, z_1 = \sqrt{1-\beta_1}\sqrt{1-\beta_2}\, x_0 + \sqrt{1-\beta_2}\sqrt{\beta_1}\, z_0 + \sqrt{\beta_2}\, z_1$$

$$x_3 = \sqrt{1-\beta_3}\, x_2 + \sqrt{\beta_3}\, z_2 = \underbrace{\sqrt{1-\beta_1}\sqrt{1-\beta_2}\sqrt{1-\beta_3}}_{\prod_{i=1}^{t} \tilde{\alpha}_i}\, x_0 + \sqrt{1-\beta_3}\sqrt{1-\beta_2}\sqrt{\beta_1}\, z_0 + \sqrt{1-\beta_3}\sqrt{\beta_2}\, z_1 + \sqrt{\beta_3}\, z_2$$

$$= \sqrt{\alpha_1 \alpha_2 \alpha_3}\, x_0 + \sqrt{\alpha_3 \alpha_2 \beta_1}\, z_0 + \sqrt{\alpha_3 \beta_2}\, z_1 + \sqrt{\beta_3}\, z_2$$

$$x_t = \sqrt{\tilde{\alpha}_t}\, x_0 + \sum_{j=1}^{t} \underbrace{\left(\sqrt{\beta_j} \prod_{i=j+1}^{t} \sqrt{\alpha_i} \right)}_{c_j} z_{j-1}$$

Appendix: Handwritten Derivations for Task 1

Since z_i are i.i.d, the second term can be represented by one z with covariance $\left(\sum_{j=1}^{t} c_j^2\right) I$, so the second term becomes $\sqrt{\sum_{j=1}^{t} c_j^2}\, z$

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{\sum_{j=1}^{t} c_j^2}\, z$$

$$c_j^2 = \beta_j \prod_{i=j+1}^{t} \alpha_i$$

$$\sum_{j=1}^{t} c_j^2 = \sum_{j=1}^{t} \beta_j \prod_{i=j+1}^{t} \alpha_i$$

$$= \sum_{j=1}^{t} 1 - \alpha_j \prod_{i=j+1}^{t} \alpha_i$$

$$= (1-\alpha_t) + (1-\alpha_{t-1})\alpha_t + (1-\alpha_{t-2})\alpha_{t-1}\alpha_t + \dots + (1-\alpha_1)\alpha_2\alpha_3\alpha_4 \dots \alpha_t$$

$$= 1 - \alpha_t + \alpha_t - \alpha_{t-1}\alpha_t + \alpha_{t-1}\alpha_t - \alpha_{t-2}\alpha_{t-1}\alpha_t + \dots + \alpha_2\alpha_3\alpha_4 \dots \alpha_t - \alpha_1\alpha_2\alpha_3 \dots \alpha_t$$

$$= 1 - \alpha_1\alpha_2\alpha_3 \dots \alpha_t = 1 - \bar{\alpha}_t$$

$$\therefore\ x_t = \sqrt{\bar{\alpha}_t}\, \beta_0 + \sqrt{1-\bar{\alpha}_t}\, z$$

Appendix: Handwritten Derivations for Task 1

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

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

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

$$= \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)}{\sqrt{\bar{\alpha}_t}(1-\bar{\alpha}_t)}x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}}{\sqrt{\bar{\alpha}_t}}(1-\alpha_t)s_\theta(x_t, t)$$

$$= \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t + \frac{(1-\alpha_t)}{\sqrt{\alpha_t}(1-\bar{\alpha}_t)}x_t + \frac{1}{\sqrt{\alpha_t}}(1-\alpha_t)s_\theta(x_t, t)$$

$$= \frac{\alpha_t(1-\bar{\alpha}_{t-1}) + (1-\alpha_t)}{\sqrt{\alpha_t}(1-\bar{\alpha}_t)}x_t + \frac{1}{\sqrt{\alpha_t}}(1-\alpha_t)s_\theta(x_t, t)$$

$$= \frac{\alpha_t - \bar{\alpha}_t + 1 - \alpha_t}{\sqrt{\alpha_t}(1-\bar{\alpha}_t)}x_t + \frac{1}{\sqrt{\alpha_t}}(1-\alpha_t)s_\theta(x_t, t)$$

$$= \frac{1-\bar{\alpha}_t}{\sqrt{\alpha_t}(1-\bar{\alpha}_t)}x_t + \frac{1}{\sqrt{\alpha_t}}(1-\alpha_t)s_\theta(x_t, t)$$

$$= \frac{1}{\sqrt{\alpha_t}}\left(x_t + (1-\alpha_t)s_\theta(x_t, t)\right)$$

Appendix: Handwritten Derivations for Task 1

3.) score-predicting:

$$\hat{x}_0 = \frac{1}{\sqrt{\hat{\alpha}_t}}\left(x_t + (1-\hat{\alpha}_t)\, s_\theta(x_t, t)\right)$$

DDPM trains $\epsilon_\theta(x_t, t)$ to predict:

$$x_t = \sqrt{\hat{\alpha}_t}\, x_0 + \sqrt{1-\hat{\alpha}_t}\, z$$

$$x_0 = \frac{1}{\sqrt{\hat{\alpha}_t}}\left(x_t - \sqrt{1-\hat{\alpha}_t}\, z\right)$$

$$\epsilon_\theta(x_t, t) \approx z$$

$$\hat{x}_0 = \frac{1}{\sqrt{\hat{\alpha}_t}}\left(x_t - \sqrt{1-\hat{\alpha}_t}\, \epsilon(x_t, t)\right)$$

Set the two $\hat{x}_0$ equal:

$$-\sqrt{1-\hat{\alpha}_t}\, \epsilon(x_t, t) = (1-\hat{\alpha}_t)\, s_\theta(x_t, t)$$

$$s_\theta(x_t, t) = -\frac{1}{\sqrt{1-\hat{\alpha}_t}}\, \epsilon_\theta(x_t, t)$$

$$\therefore \quad x_{t-1} = \frac{1}{\sqrt{\hat{\alpha}_t}}\left(x_t - \frac{(1-\alpha_t)}{\sqrt{1-\hat{\alpha}_t}}\, \epsilon_\theta(x_t, t)\right)$$