Diffusion-Based Priors for Inverse Imaging Tasks

Timothy Jacques

Abstract—Inverse imaging is considered a ill-posed problem due to the nearly infinite number of possible solutions that are present for a given noisy or otherwise perturbed input image. Diffusion models have been shown to be a promising new methodology to solve these problems while retaining realistic and faithful images. In this project, we approach forward and reverse diffusion, and implement and evaluate several reverse diffusion conditioning methodologies on several intractable test tasks. Starting from basic image denoising and unconditional image generation, we then explore three different methods of conditioning diffusion: SDEdit, ScoreALD, and DPS. These are then evaluated on a model trained on the FFHQ dataset to quantitatively and qualitatively evaluate the effectiveness of each method. In general, diffusion models show significant promise to solve inverse imaging problems.

Index Terms—Computational Photography, Diffusion Models, Inverse Imaging, Denoising, Inpainting

1 Introduction

In modern imaging systems, one of the most common issues with realworld sensors is the presence of noise, from the sensor to physical light itself, noise manifests within the output image, regardless of the efforts to prevent it. Therefore, a strong research effort has been in place to not just reduce noise, but to remove it entirely from already noisy images. Unfortunately, noise, blur, and other common imaging errors are not a simple problem to solve, and are often impossible to directly solve for due to a nearly infinite number of possible valid solutions.

A promising domain to solve these inverse imaging problems is generative diffusion models, which can be used to iteratively denoise an image until no noise is present. This method uses learned features (from faces, objects, etc.) to project a noisy input image onto a realistic image domain, often creating completely novel images in the process. One particular application for this is generative image generation, where text conditioning allows for realistic images and videos to be generated from a single text prompt. However, if we are able to condition the generative diffusion process on an input noisy image, we may be able to use this preexisting model to faithfully recreate a non-noisy version of the image.

This project focuses upon several inverse imaging problems: deconvolution / deblurring, box inpainting (infill of an input image), and random inpainting (useful for sensor issues). Additionally, we focus upon several diffusion-based methods to solve these issues, specifically SDEdit [1], ScoreALD [2], and DPS [3].

2 Related Work

Diffusion models are far from the only method that has been recently tried to solve inverse imaging. Some popular optimization-based approaches include Alternating Direction Methods of Multipliers (ADMM) [4] and Half-Quadratic Splitting (HQS). However, these methods can often take a long time to converge, if at all. Additionally, some neural-network based methods have been proposed, such as Untrained Neural Network Priors (UNNP) [5], however they have longstanding performance issues and large computational overhead.

Diffusion methods aim to provide a data-based hybrid computational approach that produce acceptable results with reasonable computational efforts. For this reason, they have been promoted heavily for solving inverse problems.

3 Denoising Diffusion Models

To begin, we discuss the basic functional theory behind diffusion models and how new samples can be generated given a trained model. In the next section, we discuss how diffusion models can be used to solve inverse imaging problems.

3.1 Theoretical Overview

Diffusion can be separated into two main processes: forward and reverse operations. The forward process gradually adds noise to the image given a specific noise schedule until the image is fully Gaussian noise, and the reverse process gradually reduces noise until a visually clear image is reached. For this project, we use the variance-preserving (VP) version, as opposed to the variance-exploding version. The forward process can be written as

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

where x_0 is the original input image, x_t is the noisy image at timestep $t \in [0, T]$, β_t is the predefined noise schedule, and $z_t \sim \mathcal{N}(0, I)$. We can rewrite this to be dependent solely on the original image in the following statement:

$$x_t = \sqrt{\bar{a}_t} \times x_0 + \sqrt{1-\bar{a}_t} \times z \tag{2}$$

where $a_t = 1 - \beta_t$, $\bar{a}_t = \Pi_{i=1}^{t} a_i$, and $z \sim \mathcal{N}(0, I)$.

Using Tweedie's formula, we can perform a single step denoising estimate from any timestep t:

$$\hat{x}_0 = \frac{1}{\sqrt{\bar{a}_t}} \left[x_t + (1-\bar{a}_t) \nabla_{x_t} \log(p_t(x_t)) \right] \tag{3}$$

Where $\nabla_{x_t} \log(p_t(x_t))$ is learned by the trained diffusion model, hereby referred to as the score function $s_\theta(x_t, t)$.

Fig. 1. From left to right: Ground truth, Noisy input image, Denoised Image

Fig. 2. Samples of unconditional generation from the FFHQ-trained diffusion model.

Using this pretrained model, the single-step denoising operation then becomes:

$$\hat{x}_0 = \frac{1}{\sqrt{\bar{a}_t}}\left[x_t + (1-\bar{a}_t) \times s_\theta(x_t, t)\right] \tag{4}$$

Expanding this back out to a single timestep, we end with the following:

$$x_{t-1} = \frac{1}{\sqrt{a_t}}\left[x_t + (1-a_t) \times s_\theta(x_t, t)\right] + \sigma z \tag{5}$$

This final step is used within the remainder of the project.

3.1.1 Noise Prediction Network

Instead of a score prediction network $s_\theta(x_t, t)$, a noise prediction network

$$\epsilon_\phi(x_t, t) = -s_\theta(x_t, t) \times \sqrt{1-\bar{a}_t} \tag{6}$$

can be equivalently learned instead. This would then make equation 5 become the following:

$$x_{t-1} = \frac{1}{\sqrt{a_t}}\left(x_t - \frac{1-a_t}{\sqrt{1-\bar{a}_t}}\epsilon_\phi(x_t, t)\right) \tag{7}$$

3.2 Single Step Denoising

Given we have determined a method to jump directly from a noisy image to a clean image in Equation 4, we can use this to directly denoise noisy images as-is. An example of this method is shown in Figure 1. The model's performance in denoising is fairly high quality, although the single-step denoising method often smooths details that are lost in the noise.

3.3 Unconditional Sampling

With denoising models, we can also perform unconditional sampling to acquire a novel image from the model. To do this, we simply start with a fully noisy image $x_T \sim \mathcal{N}(0, I)$, and then iteratively decrease the amount of noise [6]. The specific algorithm is described in Algorithm 1. Several sample outputs from the FFHQ model utilized in this project are shown in Figure 2

Algorithm 1 Unconditional Reverse Diffusion

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

4 Conditional Diffusion Methods

As previously seen, diffusion can be used to unconditionally sample novel points in the trained distribution. While this is useful to obtain semi-realistic novel images, a more interesting problem is *conditioning* the output to guide and control image generation. Given a noisy or otherwise perturbed measurement input, can we use a reverse-diffusion model to produce a plausible clean measurement?

Instead of solving for $\nabla_{x_t} \log p_t(x_t)$, we condition the reverse diffusion process by using $\nabla_{x_t} \log p_t(x_t|y)$ where y is an input noisy measurement. By Baye's rule:

$$\begin{aligned} \nabla_{x_t} \log p_t\left(x_t|y_t\right) &= \nabla_{x_t} \log p_t(x_t) + \nabla_{x_t} \log p_t\left(y|x_t\right) \\ \nabla_{x_t} \log p_t\left(x_t|y_t\right) &= s_\theta(x_t, t) + \nabla_{x_t} \log p_t\left(y|x_t\right) \end{aligned} \tag{8}$$

Where $\nabla_{x_t} \log p_t\left(y|x_t\right)$ is denoted as the "posterior". Because the posterior is intractable, providing a good estimate to replace this value is a widely researched subject. The rest of this project focuses on comparing the previous work done to estimate for this value. To amend Algorithm 1, we can simply add another step that adds in this posterior correction term:

Algorithm 2 Conditional Reverse Diffusion

$x_T \sim \mathcal{N}(0, I)$
for $t = T$ **to** 1 **do**
  $z \sim \mathcal{N}(0, I)$ if $t > 1$, else $z = 0$
  $x'_{t-1} = \frac{1}{\sqrt{a_t}}\left[x_t + (1-a_t) \times s_\theta(x_t, t)\right] + \sigma z$
  $x_{t-1} = x'_{t-1} + \zeta_t \nabla_{x_t} \log p_t\left(y|x_t\right)$
end for
return x_0

where ζ_t is some unknown linear scaling term. As the rest of the operations are linear, we can perform this additional operations after the original estimate of x'_{t-1}.

Now, we will present an overview of some of the methods used to perform conditional reverse diffusion.

4.1 SDEdit

The conditioning methodology presented in the SDEdit paper is a naive method to condition the reverse diffusion process [1]. Instead of attempting to estimate the posterior term $\nabla_{x_t} \log p_t\left(y|x_t\right)$, the method presented chooses to start the reverse process at a midpoint timestep τ with a modified measurement:

$$x_\tau = \sqrt{\bar{a}_\tau} \times y + \sqrt{1-\bar{a}_\tau} \times z \tag{9}$$

Where $z \sim \mathcal{N}(0, I)$ After this, the reverse process continues unconditioned as in algorithm 1. The only parameter to be tuned is the starting timestep τ, where increasing τ results in a more realistic, however less faithful input to the

original image. This tradeoff is discussed in more detail in the evaluation in section 5.

4.2 ScoreALD

The approach taken in the ScoreALD paper [2] estimates the posterior $\nabla_{x_t} \log p_t(y|x_t)$ using the normalized difference between the measurement and the current x_t though annealed *Langevin Dynamics* [7]:

$$\zeta_t \nabla_{x_t} \log p_t(y|x_t) \approx -\frac{1}{\sigma^2 + \gamma_t^2} \nabla_{x_t} \|A(x_t) - y\|_2^2 \quad (10)$$

Because the above approximation is not valid for $t \neq 0$, the authors propose an annealing term γ_t to reduce ζ_t when the approximation is poor. However, in the implementation used in this project, the final equation is as follows:

$$\zeta_t \nabla_{x_t} \log p_t(y|x_t) \approx -\frac{1}{2 \times \gamma_t} \nabla_{x_t} \|A(x_t) - y\|_2^2 \quad (11)$$

as this was found to empirically perform better than including the σ^2 term. This essentially ensures that the approximation is always scaled similarly, regardless of the noise level, while still retaining the overall reduction at high timestep values. γ_t is adjusted for each problem as an individual hyperparameter.

4.3 DPS

Another possible approach is Diffusion Posterior Sampling (DPS), which is proposed by the authors of [3]. DPS takes a similar approach to ScoreALD, however the x_t term in the log likelihood estimation is replaced by the original image estimation $\hat{x}_0$ (see equation 4), and a different ζ_t term is adopted:

$$\zeta_t \nabla_{x_t} \log p_t(y|x_t) \approx \zeta_t \nabla_{x_t} \|A(\hat{x}_0) - y\|_2^2 \quad (12)$$

$$\zeta_t = \frac{-\zeta}{\|\nabla_{x_t} \|y - A(\hat{x}_0)\|_2^2\|} \quad (13)$$

Fig. 3. Evaluation input images. From top-left: ground truth, deconvolution task, box inpainting task, random inpainting task.

where $\zeta \in [0.1, 1]$ is a hyperparameter that is tuned per inverse imaging problem.

5 Evaluation and Results

To evaluate the various inverse imaging methods, we devise three separate tasks: deconvolution (deblurring), box inpainting, and random inpainting. Each of these are common inverse imaging problems that require some form of image generation to solve properly. Throughout the evaluation, will refer to a sample input image from the validation set of the Flicker-Faces-HQ (FFHQ) dataset [8]. The three input images, plus the ground truth, are shown in Figure 3.

To additionally provide quantitative evaluation values, we calculate two separate metrics: Peak Signal-To-Noise Ratio (PSNR) and Learned Perceptual Image Patch Similarity (LPIPS). While PSNR is commonly used in the image processing community, it is often criticized for not fully representing human perception, as many images with "good" PSNR may appear very visually different to an observer. Therefore, LPIPS is additionally used. LPIPS uses a small neural network to better emulate the feature extraction performed by human perception, enabling a better quality metric than normal image processing statistics alone [9]. All of the output images discussed for the remainder of this section are presented in Figure 4

5.1 SDEdit

For the start timestamp hyperparameter τ, we chose to test $\tau = [300, 400, 500]$, as these are closest to the usable range of values, rather than extremes. As τ increases, the visual similarity between the ground truth and the final output image decreases. At $\tau = 300$, the output for the deconvolution task is fairly faithful, however still a bit blurry. Both inpainting tasks still have significant perturbations, and are completely unrealistic. At $\tau = 400$, the deconvolved image is clear and realistic, but not particularly faithful to the ground truth. Both inpainting tasks remain unrealistic. At $\tau = 500$, all images are fairly realistic, however the output has devolved significantly from the ground truth. Both quantitatively and qualitatively, SDEdit does not offer good performance in inverse imaging problems. Therefore, it is best focused on generating controllable outputs, as discussed in the original paper [1].

5.2 ScoreALD

ScoreALD is tested using various ζ hyperparameter values tuned to each individual problem. At $\zeta_t \in [10, 15]$, we can see that the performance of this method for the deconvolution task is quite faithful, with little blurring and a realistic look. There is still noticeable differences from the ground truth, however they are minimal and unnoticeable at a glance. The random inpainting also looks very faithful to the original, with a very low PSNR and LPIPS. Box inpainting does not perform well, with very prominant artifacting present. At $\zeta_t \in [28, 33]$, we see that the deconvolution task begins to suffer from some artifacting, as the face, especially the mouth, become more distorted. However, the random inpainting and box inpainting tasks both produce very realistic results at this schedule, with the lowest LPIPS

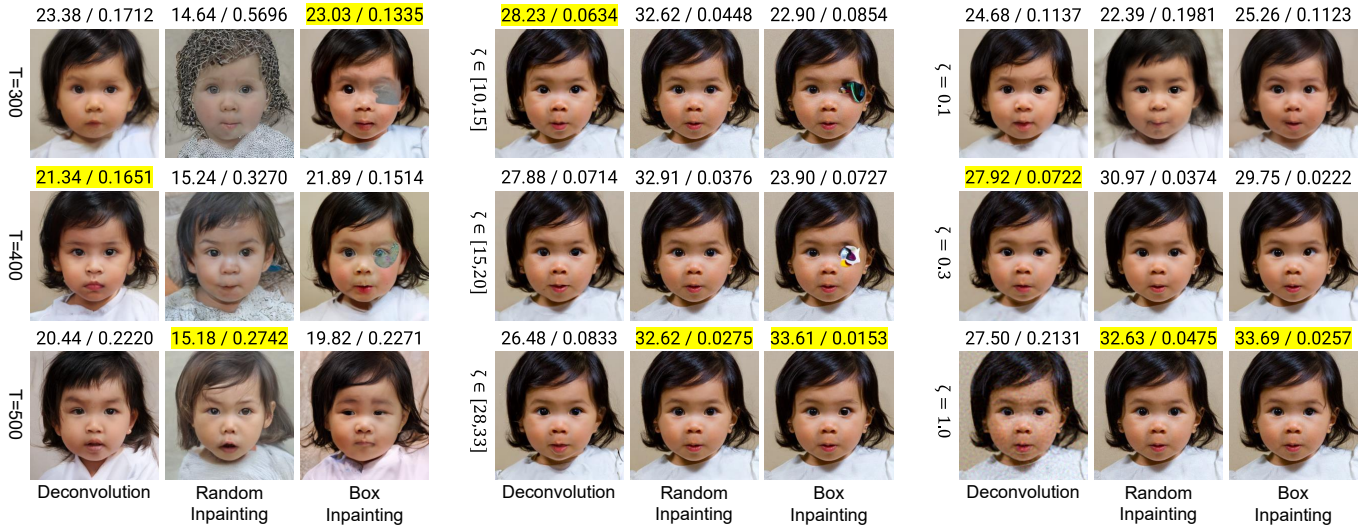

Fig. 4. Visual results from SDEdit, ScoreALD, and DPS respectively. Highlighted values show best performance in column. PSNR/LPIPS.

scores out of all three methods presented. If nitpicking, there are still visible differences in the eyes, however they would be unnoticeable at a glance. ScoreALD performs remarkably well, given proper hyperparameter tuning.

5.3 DPS

DPS is also tested using various ζ hyperparameter values, optimized for each task. We try three values: $\zeta \in [0.1, 0.3, 1.0]$. At $\zeta = 0.1$, the images are oddly smoothed of detail, and faces are fairly faithful, although not particularly realistic. The quantitative PSNR and LPIPS values support these takeaways. At $\zeta = 0.3$, all three tasks become more realistic, with some detail returning. At this ζ, the PSNR and LPIPS for the deconvolution task are best, however the output is still lesser than the ScoreALD result. At $\zeta = 1.0$, the deconvolution task starts to become blurred, and low amplitude chrominance noise is present. For the other two tasks, this noise is not present, and both look very faithful to the original image. These are the best performers for DPS, however ScoreALD still beats them quantitatively. Qualitatively, the box inpainting result from DPS is more realistic due to the generation of the eye, however this could be due to this single result, rather than being representative of the method as a whole. In general DPS performs well across these tasks, however the artifacting from high / low ζ values is very noticeable.

6 Discussion and Conclusions

Through this project, several methods were explored to solve intractable inverse imaging problems, such as deconvolution, random inpainting, and box inpainting. Each of these methods find a different way to estimate the posterior of the image, $\nabla_{x_t} \log p_t(y|x_t)$. All methods were tested on the same human face, and evaluated in both qualitative and quantitative methods.

Given the proper tuning, the best contender between the three methods presented was modified ScoreALD, both quantitatively and qualitatively. However, without tuning, DPS appears to be the best method presented, as the results were more acceptable across a wide range of hyperparameter inputs. Notably, both ScoreALD and DPS require much more computational effort than SDEdit, as an additional forward process and gradient calculation is necessary to estimate the posterior in both cases.

Overall, diffusion models seem very promising to reliably and faithfully recreate source images without significant disturbances, given a well-trained model that maps well to the image subject.

Acknowledgments

The authors would like to thank the EE367 instruction team for their support and a great quarter of learning!

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," 2022. [Online]. Available: https://arxiv.org/abs/2108.01073

[2] A. Jalal, M. Arvinte, G. Daras, E. Price, A. G. Dimakis, and J. I. Tamir, "Robust compressed sensing mri with deep generative priors," 2021. [Online]. Available: https://arxiv.org/abs/2108.01368

[3] H. Chung, J. Kim, M. T. Mccann, M. L. Klasky, and J. C. Ye, "Diffusion posterior sampling for general noisy inverse problems," 2024. [Online]. Available: https://arxiv.org/abs/2209.14687

[4] P. Neal, C. Eric, P. Borja, and E. Jonathan, "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.

[5] A. Qayyum, I. Ilahi, F. Shamshad, F. Boussaid, M. Bennamoun, and J. Qadir, "Untrained neural network priors for inverse imaging problems: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 5, pp. 6511–6536, 2023.

[6] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," 2020. [Online]. Available: https://arxiv.org/abs/2006.11239

[7] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," 2020. [Online]. Available: https://arxiv.org/abs/1907.05600

[8] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," 2019. [Online]. Available: https://arxiv.org/abs/1812.04948
[9] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," 2018. [Online]. Available: https://arxiv.org/abs/1801.03924