Comparing Diffusion Methods for Image Generation

Neha Vinjapuri

Abstract—Diffusion models have emerged as powerful generative models capable of producing high-quality images. These models work by iteratively denoising a randomly sampled noise image to generate realistic outputs. The ability to leverage diffusion models extends beyond generation and into solving inverse problems such as image inpainting and deconvolution. This poster explores how diffusion models can be used in these contexts and evaluates different methods for solving these tasks.

Index Terms—Diffusion Models, Image Generation, Inverse Problems, Image Inpainting, Deconvolution

1 Motivation

Diffusion models have emerged as powerful generative models capable of producing high-quality images. These models work by iteratively denoising a randomly sampled noise image to generate realistic outputs. One of the key strengths of diffusion models is their flexibility, which allows them to be generalized to a wide range of tasks beyond simple image generation, including image inpainting, deconvolution, super-resolution, and more. Compared to traditional methods, diffusion models offer faster and more efficient sampling processes, as they require fewer steps to reach high-quality results. Additionally, they can be applied to diverse domains, such as medical imaging, computer vision, and even artistic image generation, making them suitable for a wide array of real-world applications. This generalization potential allows diffusion models to scale to more complex, high-dimensional tasks, making them more versatile than many previous generative models like GANs or VAEs.

The impact of performing these image tasks with diffusion models is significant. In image inpainting, for example, diffusion models can recover missing parts of an image in a way that preserves both local structure and global coherence, improving the quality of reconstructed images. In deconvolution tasks, diffusion models can reverse the effects of image blurring, leading to sharper, more detailed reconstructions. These advances have important applications in fields ranging from medical diagnostics (where high-quality reconstructions of incomplete or noisy data are crucial) to image editing and restoration, opening the door to more effective and efficient solutions for image reconstruction and enhancement.

2 Related Work

Diffusion models build on the foundations laid by earlier generative models such as Generative Adversarial Networks (GANs) [2], Variational Autoencoders (VAEs)[5], and autoregressive models like PixelCNN[7]. These models have each contributed to the advancement of image generation, but they each come with their own strengths and limitations.

• *Stanford University, nehavin@stanford.edu*

Generative Adversarial Networks (GANs) consist of two neural networks, a generator and a discriminator, that are trained in opposition. The generator tries to create realistic images, while the discriminator evaluates their authenticity by distinguishing between real and fake images. GANs have been widely used for high-quality image generation due to their ability to generate sharp and realistic outputs. However, they are often difficult to generalize to other tasks, such as image inpainting or deconvolution. Additionally, training GANs can be unstable, with issues such as mode collapse, where the generator produces a limited variety of images.

Variational Autoencoders (VAEs) work by learning a latent space that captures the distribution of data. They then use this latent space to generate new images by sampling from it and decoding the samples into the image space. VAEs are well-suited for tasks like interpolation and representation learning. However, they tend to produce blurry images because the model emphasizes a regularization term that can lead to smooth reconstructions. VAEs also struggle with generating images that capture fine-grained details, making them less effective for high-resolution image generation tasks.

Autoregressive Models/PixelCNN generate images by modeling the distribution of pixels sequentially, conditioning on the previously generated pixels. This approach ensures high-quality, pixel-wise accurate image generation. While it excels in generating highly detailed images, it suffers from slow sampling speeds because of the sequential nature of pixel generation. This makes it less practical for real-time applications or tasks requiring fast generation.

While these models have contributed to the field of generative image modeling, diffusion models address some of the inherent limitations of these approaches, particularly in terms of flexibility, speed, and the quality of generated images. Unlike GANs, VAEs, and autoregressive models, diffusion models are more robust in various inverse problem settings, such as image inpainting and deblurring, while also generating high-quality images with greater stability and less noise.

3 METHODS

We compare the following diffusion-based methods:

3.1 Baseline DDPM Sampling

3.1.1 Forward Diffusion Process (Noising)

In this work, we adopt the variance-preserving (VP) formulation of diffusion models. The forward diffusion process adds noise to the original data $\mathbf{x}_0$ over T steps, as follows:

$$\mathbf{x}_t = \sqrt{1-\beta_t}\,\mathbf{x}_{t-1} + \sqrt{\beta_t}\,\mathbf{z}_{t-1}, \quad t = 1, 2, \ldots, T,$$

where $\mathbf{z}_{t-1} \sim \mathcal{N}(0, I)$ and β_t is the noise schedule. This can be rewritten in terms of $\mathbf{x}_0$ as:

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

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$.

3.1.2 Reverse Diffusion Process (Denoising)

The reverse diffusion process aims to recover the original data by denoising $\mathbf{x}_t$. This can be written as:

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

or equivalently using the noise prediction network ϵ_θ:

$$\hat{\mathbf{x}}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t + (1-\alpha_t)\,\epsilon_\theta(\mathbf{x}_t, t)\right).$$

This formulation allows for the denoising of the image in each step, progressively recovering the original data from the noise.

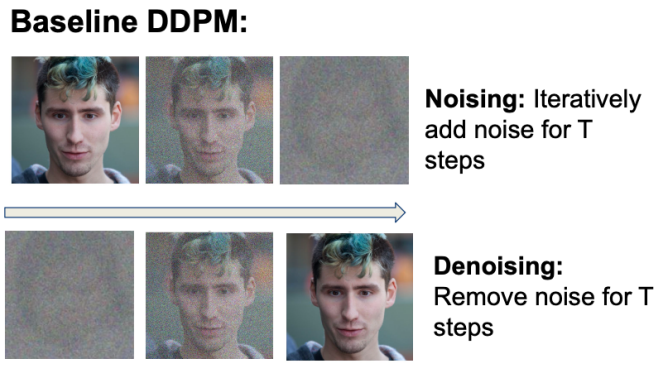

Fig. 1. Baseline DDPM.

3.2 SDEdit

SDEdit[6] is a powerful framework for image synthesis. The motivation behind SDEdit is to offer more flexible control over image generation and modification by leveraging the continuous noise perturbation process. It allows for fine-grained control over the generation and editing process, enabling tasks such as inpainting, super-resolution, and even style transfer.

SDEdit starts with an image task. In our case, we looked at inpainting and deconvolution. We first pass our ground truth image through a measurement model, that adds the perturbation to the image, along with some measurement noise.

$$y = A\mathbf{x} + \epsilon$$

Where A is the measurement matrix (e.g., a downsampling operator or mask), and ϵ is the noise term. We then add noise to this image for a partial number of timesteps, and learn the denoising process similar to DDPM.

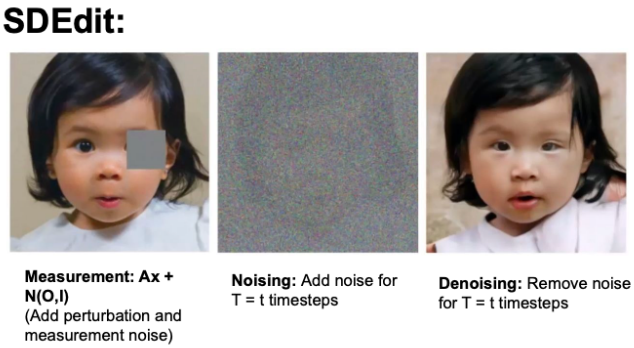

Fig. 2. SDEdit Process.

3.3 ScoreALD

ScoreALD[4] builds off of unconditionally generating images to now adding a constraint using an annealing strategy for more stable reconstructions. Specifically, we make use the following process with a naive assumption.

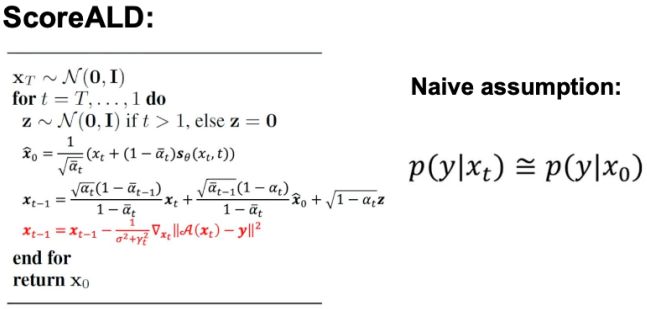

Fig. 3. ScoreALD Process.

This reduces the difference from the original image to the image denoised produced at the previous timestep.

3.4 DPS[1] (Diffusion Posterior Sampling)

The DPS method is similar to ScoreALD in that it introduces a constraint on unconditional generation. Specifically, it reduces the difference between measurement and this time, the denoised out of the current timestep. This is to potentially smooth the updates.

4 DATASET

The dataset used for experiments is the Flickr-Faces-HQ Dataset (FFHQ).

Diffusion Posterior Sampling:

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

Improved assumption:

$p(y|x_t) = \int p(y|x_0)p(x_0|x_t)dx_0$

Fig. 4. DPS Process.

- **Age and Ethnicity:** The dataset includes faces from individuals of various ages, ranging from infants to elderly adults. It also has a wide representation of different ethnicities, contributing to its diversity and making it suitable for training generative models that can generalize across diverse demographics.
- **Image Backgrounds:** FFHQ images feature a variety of backgrounds, ranging from indoor settings to outdoor environments. This variation helps models trained on this dataset to learn to synthesize faces in different contexts, with different lighting conditions and background structures.
- **Accessories:** The dataset contains a wide range of accessories, such as eyeglasses, sunglasses, hats, and more. These features are important for models focused on fine-grained facial feature generation and manipulation, allowing them to learn to handle and generate images with these diverse attributes.

Fig. 5. FFHQ Dataset.

5 Experimental Results

5.1 Method Comparison

Method	PSNR	LPIPS
Baseline DDPM	32.4588	0.0.0783
SDEdit	20.3665	0.21037
ScoreALD	29.26346	0.21039
DPS	27.4746	0.04443

TABLE 1
Best PSNR and LPIPS comparison for different methods.

5.2 Visual Comparisons and Ananlysis of Each Method

- **Baseline DDPM**: The Baseline DDPM tends to smooth out images, often resulting in the loss of fine details. While it is effective at retaining the main components of an image, it struggles to preserve intricate textures and small features, leading to a more generalized and less detailed output. You can see this in Figure 6, where the details of DiCaprio's beard or the red panda's fur are lost.
- **SDEdit**: SDEdit is capable of retaining all components of the image, but it sometimes introduces uncanny features, particularly in the background details. This may occur when the model faces difficulty aligning the background with the rest of the image, leading to inconsistencies or artifacts in the final result.
- **ScoreALD**: ScoreALD improves image sharpness by utilizing an annealing factor that adds stability to the denoising process. This allows the model to produce sharper, more detailed images compared to the baseline, without the blurring often seen in simpler diffusion methods. We can see the rest of the image is fairly preserved, which is different from SDEdit. Adjusting the annealing factor also allowed for better results - for the deconvolution task, a value of 15-20 and 10-15 for inpainting.
- **DPS**: DPS excels at removing blurry or blocked portions of the image while still preserving important details. The scaling factor employed by DPS enhances feature retention, leading to more accurate and detailed reconstructions, especially in areas that would typically suffer from blurring in other methods. I tested different values of the scale factor and found that a higher scale factor worked better for inpainting tasks (around 1.0), and a lower one for deconvolution (around 0.3). I also experimented with a new technique, adding a sampling rate, which averages a number of samples at each time step. I found that an increased number of samples produced better results.

6 Overall Analysis and Conclusion

DPS outperforms other methods in both PSNR and Learned Perceptual Image Patch Similarity (LPIPS). ScoreALD balances PSNR and LPIPS well with annealing strategies. SDEdit is effective for controlled modifications but struggles with background inconsistencies. However, there are a diverse set of tasks, unrelated to image accuracy like style transfer, that would be well-suited for SDEdit.

7 Future Work

- Improving computational efficiency for real-time applications.
- Exploring hybrid approaches combining diffusion with transformer models.
- Explore latent diffusion models.
- Extending experiments to more diverse datasets beyond FFHQ.

Fig. 6. DDPM Baseline Results with different no. of timesteps.

Fig. 7. Sample results from unconditional generation.

Fig. 8. Inpainting and deconvolution results from SDEdit.

Fig. 9. Inpainting and deconvolution results from ScoreALD.

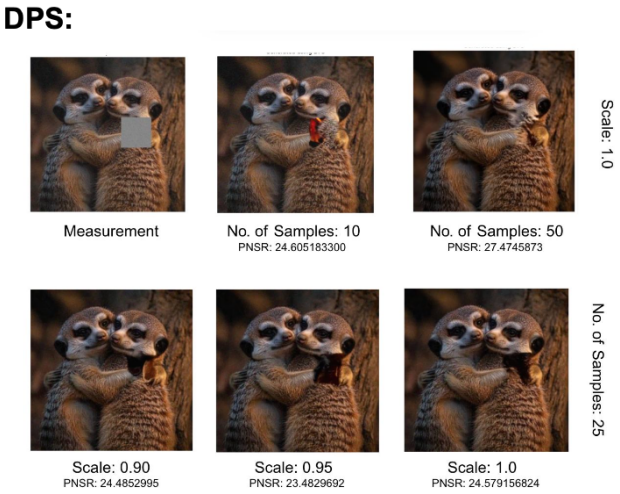

Fig. 10. DPS results with variable scale factor and sampling rate.

8 Default Project

Task 1 Derivations:
https://drive.google.com/file/d/1Yg1ry2f7lrV-l_-E9CM2P-Dv6ltKqwIs/view?usp=sharing

9 Acknowledgments

This research was conducted as part of the CS448I course at Stanford University. Special thanks to the faculty and peers for their valuable feedback and discussions!

10 References

1) Chung, H., et al. (2023). "Diffusion Posterior Sampling for General Noisy Inverse Problems."
2) Goodfellow, I., et al. (2014). "Generative Adversarial Networks."
3) Ho, J., et al. (2020). "Denoising Diffusion Probabilistic Models."

4) Jalal, A., et al. (2021). ”Robust Blind Inverse Problems in Imaging with Score-based Generative Models.”
5) Kingma, D. P., & Welling, M. (2013). “Auto-Encoding Variational Bayes.”
6) Meng, C., et al. (2022). ”SDEdit: Image Synthesis and Editing with Stochastic Differential Equations.”
7) van den Oord, A., et al. (2016). “Conditional Image Generation with PixelCNN Decoders.”