Interpretability of Denoising Diffusion Probabilistic Models for MNIST Digit Generation *

Chhavi Shah, Devan Srinivasan, Saleh Yasin

November 2023

1 Introduction

Diffusion models were inspired by the principle of diffusion in thermodynamics, the process by which particles move from areas of higher concentration to lower concentration to reach equilibrium. Applied to machine learning, this process can be used to diffuse some unknown distribution to a normal distribution, and then, if reversed, one could sample from the original distribution through the (diffused) normal one. This is the motivation behind Denoising Diffusion Probabilistic Models (DDPMs), the focus of this paper. Originally proposed by Ho and his colleagues in 2020, these models posit a distribution over some image space that we wish to sample (to then generate an image). To learn this distribution with sufficient data, one could diffuse the sample images, and then if accurately reversed, approximately sample from this image space. This is a high-level explanation of how DDPMs function. Gaussian sampled noise is repeatedly added to sample images in a Markov chain, and a classifier learns to recognize the amount of noise added to an image. Then, this "noise predictor" inference can be undergone by repeatedly reversing that Markov process and removing noise, using our predictor, to recover an image. Considering this generative process, this paper explores the interpretability of DDPMs and the denoising process for image generation constrained to the MNIST dataset.

MNIST was chosen as the dataset for a few reasons. In literature (including image generation and interpretability), MNIST is a benchmark dataset for evaluating approaches, solutions, and strategies in machine learning. Additionally, it is easily accessible with a considerable amount of data for our purposes. Additionally, we may justify the choice of DDPMs. While plenty of other generative models exist, we chose diffusion models from sole interest. The research team was fascinated by the denoising process (it's analogous to carving a sculpture) so we thus chose to understand it experimentally, which is what this paper overviews.

2 Related Work

Diffusion-based models and generative AI for imagery is a vast field that has significant research done already. Here, we overview the papers we took inspiration from or those that we believe followed a similar line of inquiry to what we were trying to achieve: the pursuit to investigate the interpretability of the generative process. In our case, it's image (digit) generation with diffusion models.

1. **Park, Ji-Hoon and Ju, Yeong-Joon and Lee, Seong-Whan**
Explaining Generative Diffusion Models Via Visual Analysis for Interpretable Decision-Making Process.

This study closely related to our problem statement, investigating the interpretability of decisions made in the denoising process. In particular, it proposed three meaningful questions (denoted R1, R2, R3 in their paper) for us to consider in our work, namely:

*The code we used for our experiments is found in: https://github.com/devan-srinivasan/digit-generation/

R1. *What regions are recovered by the diffusion model in terms of semantic and detail level?*

We didn't investigate semantics with digit generation but did assess the accuracy and detail level generated by the model, not only at the end of the denoising process but also at various time steps throughout the process.

R2. *Which specific concepts are prioritized at each time step to generate images that align with the given conditional prompt?*

We considered this, noting that our work did not involve a conditional prompt. Our experiments evaluated various timesteps in the denoising process to determine what factors were relevant to the noise prediction.

R3. *What visual concept is implied at time-step t?*

Inspired by this question, we sought to not only visualize the noise at various time steps but also investigate patterns through analysis like PCA,

2. **Luis A. Perez Rey and Vlado Menkovski and Jacobus W. Portegies**
Diffusion Variational Autoencoders

In this paper proposing the Diffusion VAE, researchers trained their model on the MNIST digit set (as we did) with various manifolds such as a sphere, tori, and torus, to investigate geometric and topological properties. While this isn't quite what we did, it did inspire us to investigate different latent spaces in PCA in hopes of observing patterns and properties as they did.

3. **Ryen Krusinga, Sohil Shah, Matthias Zwicker, Tom Goldstein, David Jacobs**
Understanding the (un)interpretability of natural image distributions using generative models

This paper focused on the interpretability of density functions over the image space, in generative models (GAN in this case). While not directly related with our work, the source of inquiry is akin to ours.

3 Problem

While diffusion models for image generation are quite successful, generating realistic state-of-the-art graphics and the related (denoising) process feels unnatural compared to humans. We create imagery building upwards from a blank canvas, whereas DDPMs do so by removing from a noisy image. This was the precise motivation for investigating these models. Namely, we explored what interpretable features the model learns and how (if at all) it relates to visual features humans use to generate images. Intuitively, this paper sets out to unravel the denoising process, but our formal objective can be stated as follows:

Research Objective:
We seek to investigate the interpretability of diffusion models and the denoising process in digit generation.

To investigate and draw insights within our time constraints, we began on a more constrained domain. A smaller domain would mean simpler and more comprehensible patterns could be observed, and confined to a single dataset also offered significant savings in training time. For those reasons, our research pertains to digit (rather than arbitrary image) generation.

4 Methods

We employed three main methods in our investigations with a series of experiments. Given our project's motivation was founded on understanding diffusion models, we naturally organized our investigations as such. Namely, we considered the model overall and its hyperparameters, the U-Net noise predictor, and the

denoising process. All experiments (except the first) used a DDPM trained on the MNIST dataset with 1000 denoising steps, and a β range as a linear space (evenly spaced sequence) from 0.0001 to 0.002.

1. **Model: Hyperparameters**: We tested the model with different hyperparameters (varying the β values and denoising steps) to see patterns in the images generated from a constant noise stream.

2. **Denoising: Principal Component Analysis**: We used principal components to see important factors in latent space to give insight into the generative process. So at varying timesteps, we executed PCA on varying states of the denoising process, especially those near the end:

   - Given a sample of N images to generate, we provide PCA at any state during the denoising process across those N samples.
   - We visualize the eigenvectors in latent space, as done in the lecture, to view "eigendigits" that constitute the resulting images (for that state).

3. **U-Net: Captum Analysis**: We used the `captum` package [Kokhlikyan], a suite of interpretability tools for PyTorch models. They offered numerous techniques and methods, all implementations of literature, to evaluate classification, regression, and other machine-learning tasks. According to what was done in [Park, Ji-Hoon, et al.], we used three of these to set up experiments with our model. For all three methods, we had a "constant" and "live" version that allowed us to evaluate at a timestep with *live* noise or with the denoising timestep using *constant* random noise. For the "live" noise this meant that we ran the evaluation while the model was denoising, as opposed to just calling the network predictor on some fixed noise (constant).

   - Grad-CAM: Proposed by Reiff, this method uses gradient analysis at convolution layers concerning the output to return a localized weighted mapping that can then be used to indicate influence to the final result - in this case, the predicted noise. In our case (repeated below), we evaluate the last convolution layer; however, Grad-CAM is traditionally used for classification or scalar regression (as implemented in Captum), so we adapted it for our purpose. We execute the analysis per-pixel (of the 28x28) to return the matrix of attributions for each, and then sum these attributions to then get a weighted attribution map of the last convolution layer concerning the output (predicted noise). We use this analysis at various time steps to see change over time.
   - Saliency Maps: Saliency maps [Simonyan] return the attributions of each input feature (pixels in our case) to the predicted class or regression value through the gradient of the output with respect to the input. Using the summed attribution technique above, we can generate these maps for the input suite and relate them to the predicted noise at varying time steps.
   - Integrated Gradients: Integrated Gradients [Sundararajan] offer an attribution score similar to saliency maps, but differ in approach: they integrate the gradients along some path from the input to output of the model in concern, relative to some baseline (from a baseline input). They then use this to estimate the relevance of each input feature (pixel in our case) as it is carried through the network.

It is important to note that in many analyses and examples, actual digits aren't generated. This is because we chose a very simple DDPM implementation with little to no modifications and optimizations.

5 Results

5.1 Hyperparameter Testing

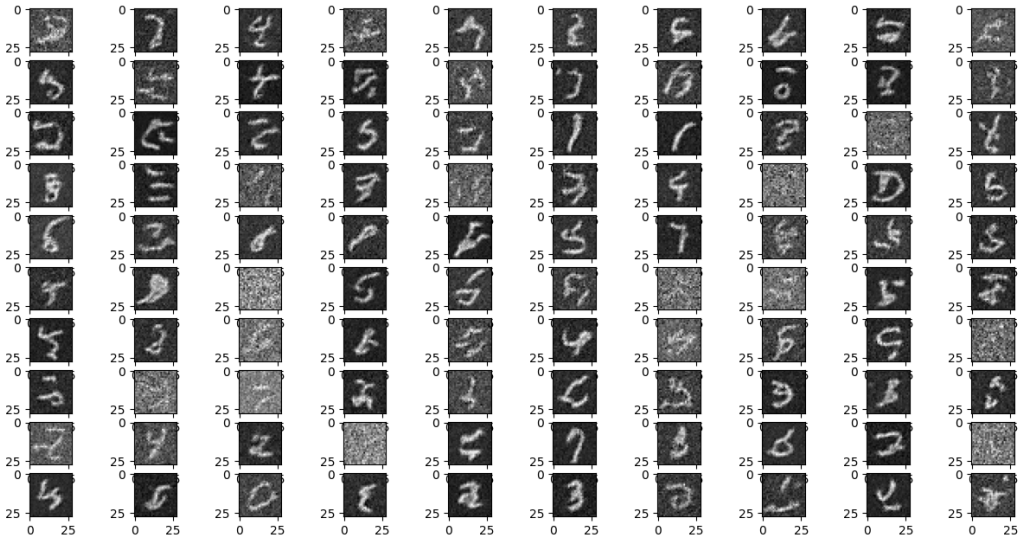
The focus of the hyperparameter testing was on varying the denoising steps and the β range. These experiments aimed to understand how changes in these hyperparameters influence the denoising process and consequently, the quality of the generated images.

5.1.1 Denoising Steps

We conducted experiments by adjusting the number of denoising steps while keeping other hyper parameters constant. We ran experiments over 100 denoising steps, 500 denoising steps, 1000 denoising steps and 5000 denoising steps, and made the following observations:

- **Increasing denoising steps**: By increasing the denoising steps, the generated images exhibited finer details and much smoother transitions during the denoising process.
- **Decreasing denoising steps**: By decreasing the denoising steps, the images were generated much more quickly. However, this led to image quality taking a significant drop and finer details not being captured.

Upon further inspection, we understood that by incrementing the denoising steps, it allowed the model to iteratively remove more "noise" from the input images. By providing the model with more opportunities to refine the estimate of the input image's underlying structure at each denoising step, the thorough denoising process allows the model to capture more intricate details.
However, this process comes with a heavy computational cost. At a higher number of denoising steps, we greatly increase the complexity of the model. As such, the model takes a much longer time to train and, consequently, generate images. This suggests a trade-off between the highly detailed generated images and the efficiency of the denoising process. A higher denoising step count contributes towards a finer image generated, but at a significant computational cost.



Low denoising step count of 100, with a β range of [0.0001, 0.002]

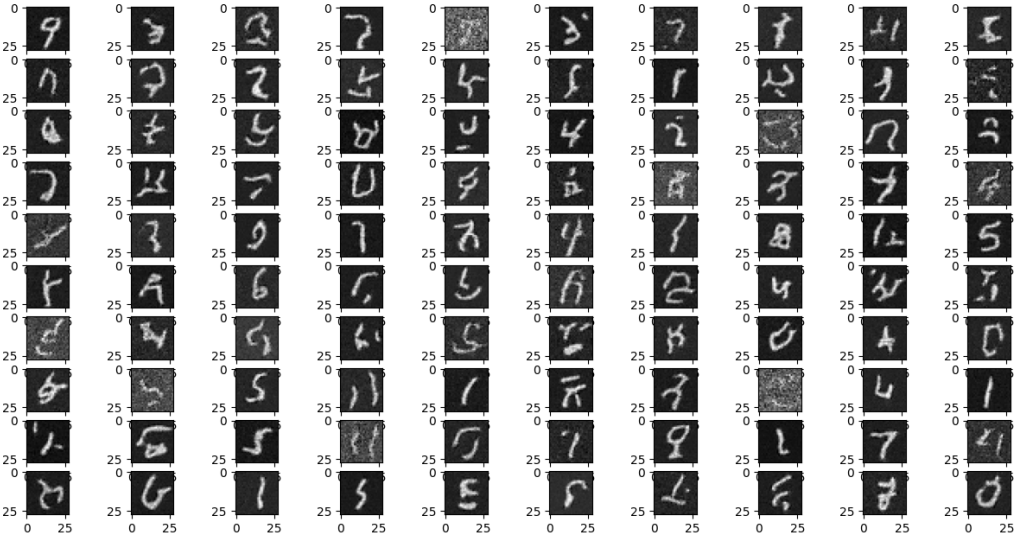
5.1.2 β Range

We redid the previous experiments but this time, we kept other hyperparameters constant and used a variety of β ranges. We ran experiments on β ranges that started off as low as 0.0001 and went as high as 0.2. The beta range was carefully selected to cover a spectrum from minimal denoising to more aggressive noise removal. We made the following observations:

- **Small Beta values**: We observed that the images formed had minimal denoising applied. Subtle variations in the images generated were retained and finer details were preserved, such as texture.
- **Gradually Increasing Beta values**: We saw a gradual increase in the denoising aggressiveness. The noise removal and feature preservation were balanced, coupled with a gradual smoothing of the rough edges that did not sacrifice any significant details of the image.

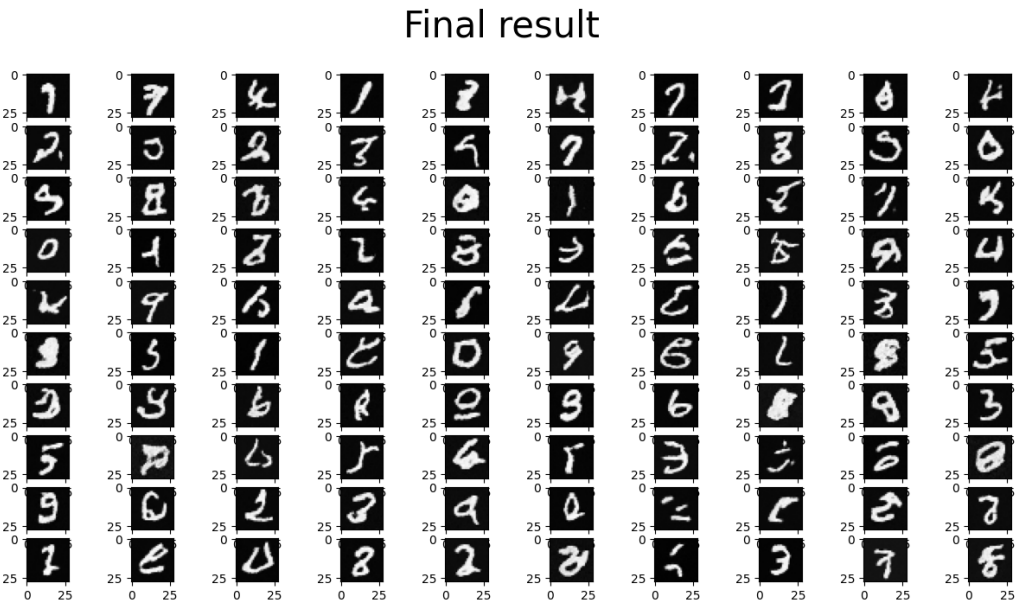
- **Larger Beta values**: We observed aggressive denoising with loss in fine detail. The generated image had clearly undergone oversmoothing, as we saw a cleaner but much less detailed output.

Further inspection on the relationship between the β ranges and the generated image provided us with valuable insight. The choice of β had a very significant influence on the trade-off between the aggression of the noise removal and the retention of fine details. Achieving an optimal compromise ensures that the model must produce visually appealing results without sacrificing any essential features.



Smaller β range of [0.0001, 0.0002] and denoising step count of 1000

Keeping these findings in mind, we chose to keep our β range from 0.0001 to 0.002, and we trained the MNIST dataset with 1000 denoising steps. This achieved a balance between trade-offs associated with the denoising steps as well as the β values.

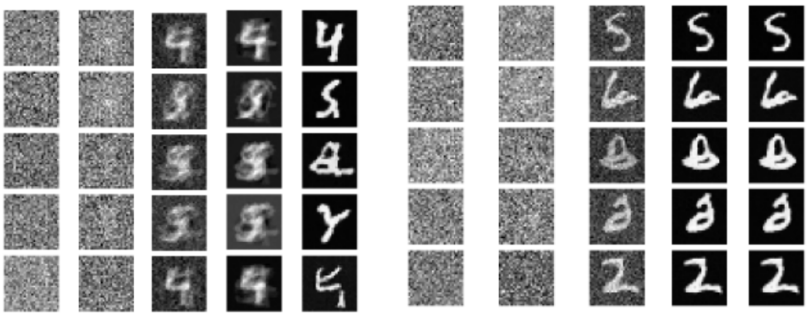


Balanced output, with denoising step count of 1000 and a β range of [0.0001, 0.02]

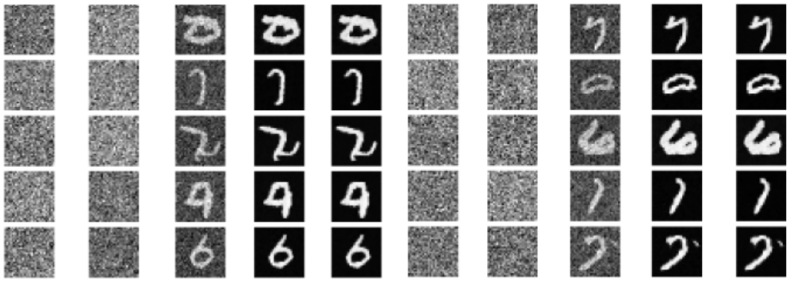
5.2 Principal Component Analysis (PCA)

The goal of running PCA on the output of our DDPM was to gain insights into the model learnings and how the denoising process works. We ran PCA over 1000 timesteps and visualised the reconstructed outputs at various timesteps in that interval, to see how the denoising was occuring.
We used sample sizes of 15 images to have a sampling size that was easy to parse.
k represents the number of principal components we run our experiments on.

Shown below are the outputs of running the model with k = 3, 50, 400, 784, recording each of the denoising outputs at timesteps = 999, 500, 100, 50, 1, along with the original images from left to right. Through these timesteps, we hoped to see the progression of the denoising. These number of principal components were chosen to be able to see how the denoising changes with k-values spread across its valid range of [1, 784]. We show 5 out of 15 of our images for brevity.



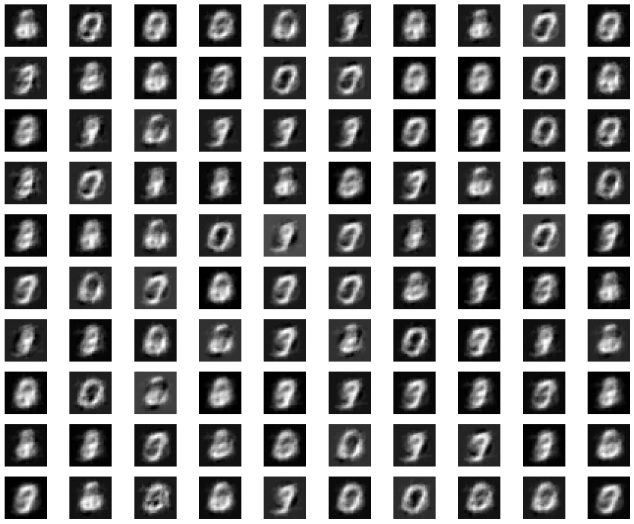
PCA: *k = 3* *k = 50*



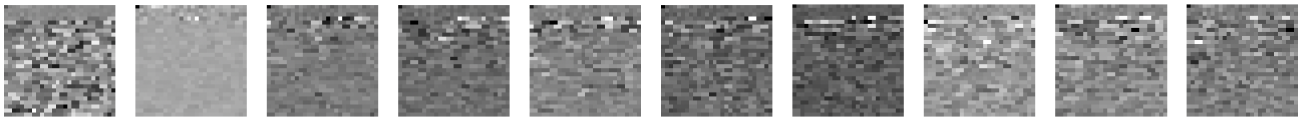
PCA: *k = 500* *k = 784*

Looking at these images, one can see that when $k > 15$ the reconstructions are just the actual denoised images at the timestep seemingly enough as opposed to k=3 when we see more blur and reminiscent of other digits. This is likely because with so many components, more than the number of samples, the entire sample of 15 images is captured in the latent basis ($784 \times k$).

As such, we decided it was worth investigating low values of k with more sample images (5 shown for brevity) to see patterns in reconstructions, as well as visualizing the eigendigits that compose the basis for latent (reduced) representation. Below we ran it with $k = 3$ and $n = 100$ and examined the latent states of $t = 100, 50, 5, 1$. Below we attached the visualization for all 100 at timestep 1, as the previous timesteps were essentially just noisier versions of this. That is, the reconstructions don't change through the process, which was expected, as from the beginning to the end, the same digit is being generated through denoising (also shown just from pure denoising outputs), and this is affirmed just by looking at the actual current state of the model at each step in diffusion (omitted).



An interesting pattern we observed, evident in this image, is that the reconstructed images all resemble zeroes or eights despite not being denoised to those digits. It suggests the figure and curves were very prominent in latent representations, which aligns well with our intuition of digits, as curves and straight lines almost uniquely identify them. We then investigated the "eigendigits" when decomposing the 100 images right before denoising, again with $k = 3$. No noticeable pattern was observed; results are shown below (10 out of 784).

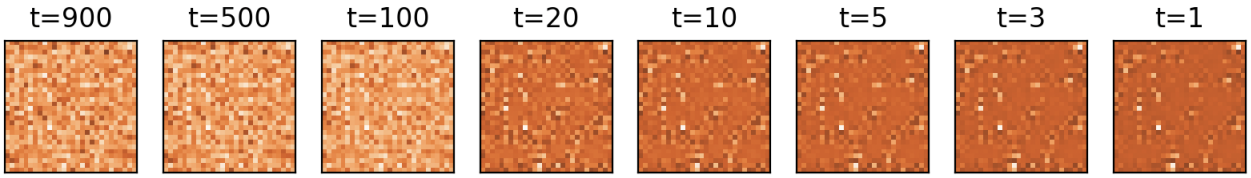


5.3 Captum Analysis

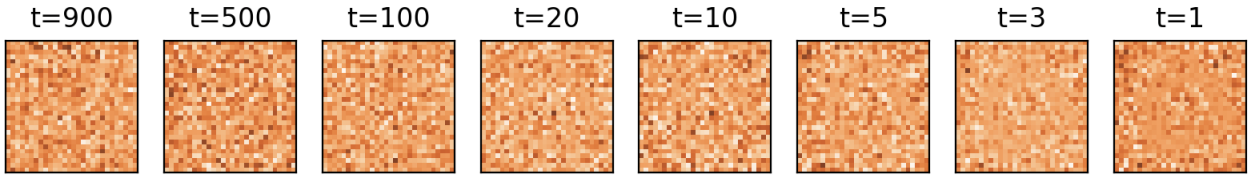
For all visualizations below, darker pixels indicate higher attribution.

5.3.1 Grad-CAM

Below are the results of Grad-CAM on the last convolution layer of U-Net, with a constant noise stream (isolating the timestep) so we can directly see the effects of timestep on the denoising process. As one may expect, earlier stages exhibit less attention as the attributions are scattered. Proceeding to the latter stages, we see the curved shapes more centred in the image (not on the border), indicating that these regions are of more concern to the model for predicting noise. Given the constant noise (and thereby lack of Markhov process), the generated image is nonsensical, but our ideas are corroborated in the live experiment.

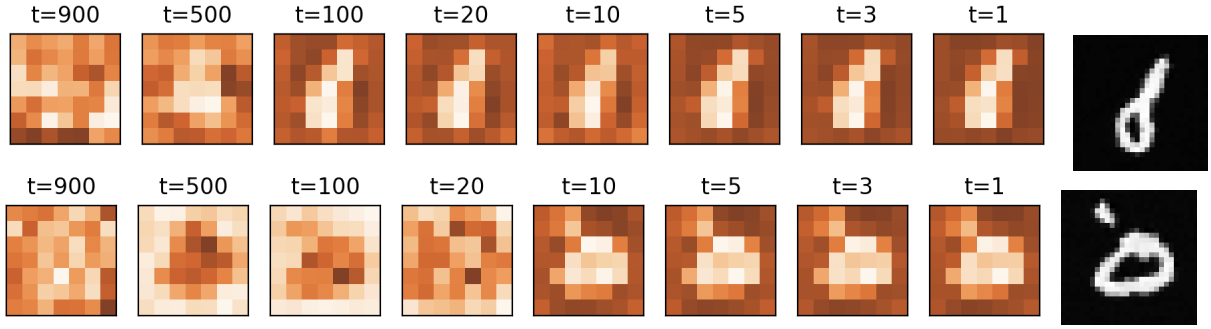


Grad-CAM - Last Convolution Layer - Constant

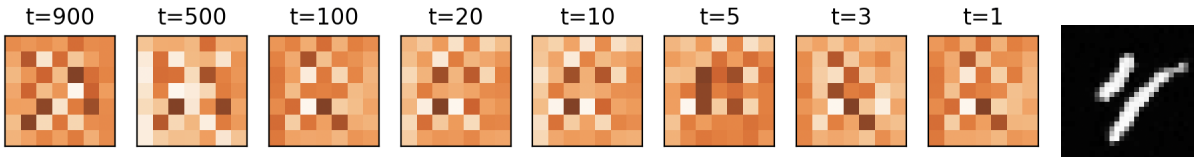


Grad-CAM - Last Convolution Layer - Live
Following this, we then sought to repeat this experiment with the output of deeper convolution layers within U-Net during the live denoising process. Note that the U-Net is composed of a down sampling process (with three main phases), the bottleneck section, and then the upsampling process (with three main phases). With that in mind, the results of these experiments are shown below, with the generated image at the end.

Grad-CAM - End of Down Layers - Live (2 examples)



Grad-CAM - Beginning of Up Layers - Live

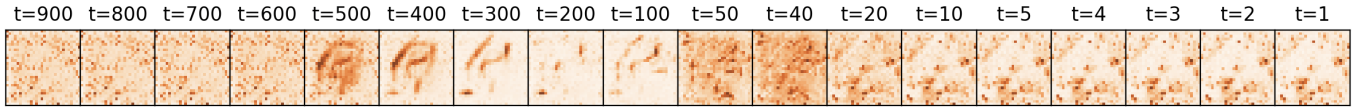


These show very interesting results. The down layers occur early during the image segmentation process and, in both scenarios, closely resemble the image generated. Furthermore, we see in the second case (that generated zero), that an inverse of relevance occurs surrounding the zero shape. In the upper layers, however, we see no discernible pattern. We may then hypothesize important denoising decisions, at every phase, occur during the down sampling of the U-Net architecture.

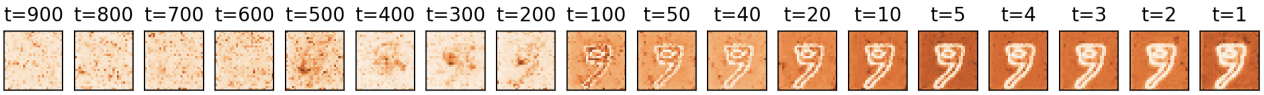
5.3.2 Saliency

We observe some incredible results just by looking at the final layer in U-Net. Holding noise constant, we see given varying timesteps, that the middle stage of denoising has interesting activity. It seems parts of the digit generated are heavily reflected here, which agrees with the live experiment. In the live denoising, we see the middle stages, timesteps 500 to 200; each "part" of the nine shapes generated has particular attentive portions. We can see the righthand outlines at t = 200, the inside portions at 300, and some outer ones at 500. Then, from t=100 onwards, the outside and inside of the nine matter a lot, while the border contributes next to nothing, and this makes perfect sense as we are predicting noise.

Saliency - Final Convolution Layer - Constant

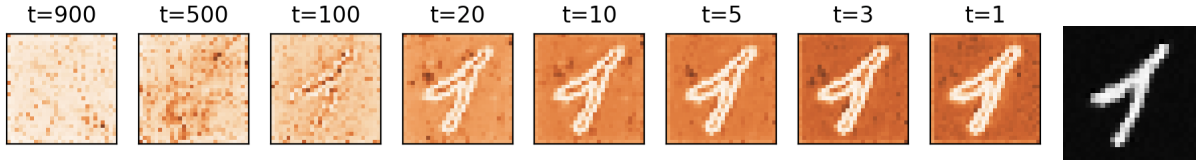


Saliency - Final Convolution Layer - Live



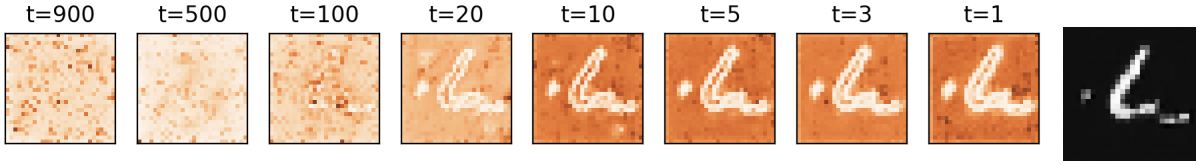
Once again we repeat this live experiment with other convolution layers in the U-Net, evaluating the down, middle, and up portions of the U-Net.

Saliency - End of Down Layers - Live



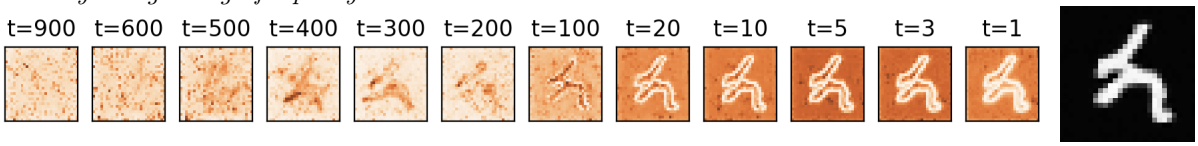
Saliency - End of Bottleneck Layer - Live

Saliency - Beginning of Up Layers - Live



Interestingly enough, we see identical patterns, as observed in the last convolution layer, with various other layers within the U-Net, in the upsampling, bottleneck, and downsampling process. In fact, the input attributions get significantly more confident throughout the process of denoising, as well as information respective to the number generated seems to occur earlier in the middle stage of denoising (indicated at t=500). This is further demonstrated when we expand this middle phase, as done below.

Saliency - Beginning of Up Layers - Live



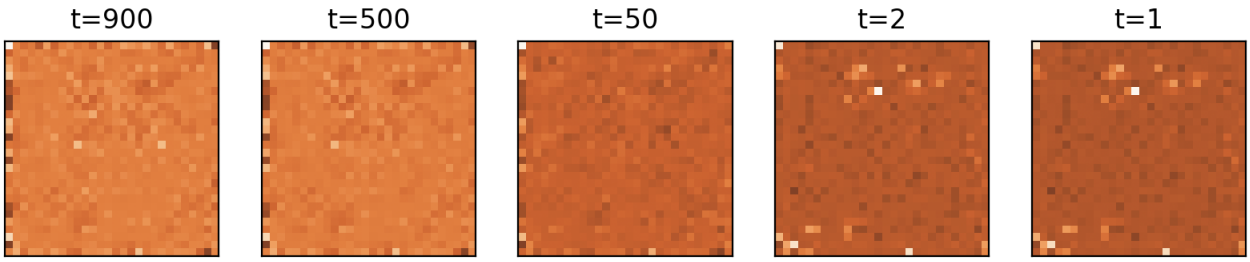
Wee see, once again, various parts of the generated image are done in different timesteps.

5.3.3 Integrated Gradients

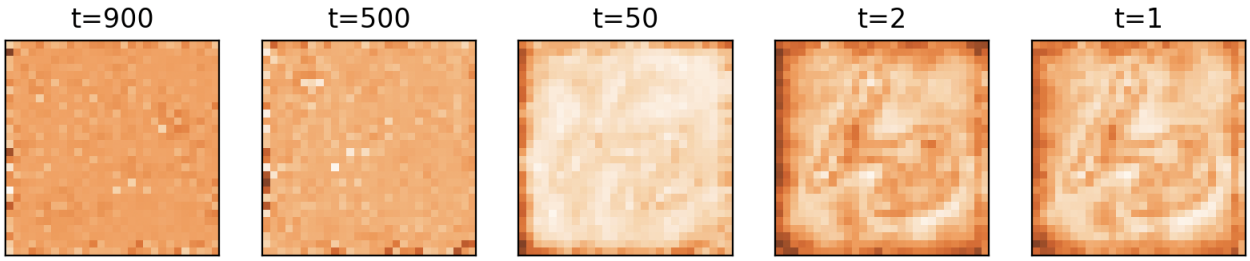
We saw no noticeable patterns integrating gradients with constant noise (i.e. no noticeable correlation with time). However, in the live experiment, we see very interesting results. Note that we used fewer time steps than saliency because integration took a long time.

Contrary to saliency, we see that the middle stages of the live experiment don't show heavy attribution from the pixels surrounding the generated shape, as t=500 and t=900 have almost identical attribution maps. However, from t=50 onwards, we see heavy attention to the window's border and the image's border generated, closing in as t decreases (model denoises). Nonetheless, we see polar attribution around the border and inside of the digit shape, as seen with the Saliency results. Another interesting phenomenon is while Saliency gave sharp differences in attribution, the integrated ones show much more of a gradient, corresponding to gradual differences.

Integrated Gradients - Final Convolution Layer - Constant



Integrated Gradients - Final Convolution Layer - Live



We did not investigate further into the U-Net with Integrated Gradients as Saliency gave similar results (seen above) which is expected as they both measure attribution from input, and the integration process generally takes a long time.

6 Future Work

Participating in this research sparked ideas for furthering this line of work and investigating other attributes of the model. They are overviewed below:

1. Conditional Sampling
   Past just arbitrary generation, there are various techniques for generating sample images from prompts, such as learned joint spaces and conditional sampling. Particularly, conditional sampling techniques alter the model in which the model denoises. Therefore, one could employ techniques and strategies as we did in this paper to see differences in generation and how the conditioning directs digit generation. This would of course fall well within the inquiry of interpreting diffusion generation models.

2. Generation Accuracy
   Another metric worth investigating is how accurate the generative model is, past just observed results. An unconventional idea we had was using state-of-the-art MNIST classifiers to classify the digits generated with an empirically tuned cutoff percent for class confidence. This way we could ideally omit images that don't resemble any digit at all. While this metric isn't objective (as the classifier is bound for error), it will give valuable insight into the quality of our model, relevant to tuning hyperparameters as well.

3. Manifolds and Latent Space
   This paper touched upon latent representations of our data, and it referenced Variational Autoencoders with the application of classifying MNIST on different manifolds [Rey]. From a similar thread, with perhaps more ambition and complexity, one could experiment with different predictors (including U-Net) and latency spaces. In such an endeavour, we could investigate learning patterns in the image generation process through patterns in different latent representations.

4. Multi-output Interpretability Methods
   We summed attributions from Captum's output to create a matching map for visualization, but Captum also has a breadth of other tools and packages, as do many other interpretability libraries. As such, it may be worth exploring other techniques to unveil more about the model and its process.

7 Discussion

The purpose of this project was to better understand how DDPMs generate images from noise by analyzing its interpretability on all fronts and breaking down what goes on within the model. Apart from learning an immense amount when working with the model and creating experiments, our results showed all parts of the model to be quite interpretable.
Please note we do not claim any conclusive explanations, merely extrapolations and ideas for intuition from results observed

PCA reconstructions suggested that curves and arcs were prominent features when reconstructing digits generated. The NxD matrix was decomposed into NxK through the DxK principal component eigenvectors. Then when reconstructed, we see prominence in the curves of the digits. This is analogous to what was done in lecture where prominent facial features like eyes and noses were reconstructed from the reconstructed faces using a similar set of parameters for PCA (notably more samples which explains the higher quality reconstructions).

The captum investigations also gave a suite of interesting results. The train U-Net predicts noise confidently with an unwavering linear path towards a denoised digit. This isn't at all surprising as the math behind this assumes a Markov noising process, but nonetheless, it was nice to see such a theory validated empirically. Live experiments gave really interesting results. With respect to gradients, we saw how the down-sampling process exhibited a rough structure of the digit through gradient attribution, indicating, and given that the down-sampling process precisely reduces dimensionality, such results were very intuitive. As per saliency, we saw a repeating pattern at every step in the U-Net, and it was one that showed the discernible digit's shape when measuring input attribution to predict noise (to be removed). Moreover, such attributions indicated heavy relevance to either the border or inside of the digit's body, indicating heavy emphasis on shape and edges. We also conjecture that the middle section of the denoising process goes through phases for each region of the digits as the results showed at different time steps different portions of the generated image had more "focus" (higher input attributions). Lastly, integrating the gradients produced similar results to saliency but the attribution scores calculated were much more gradual, and only latent sections of the denoising process seemed to show any relevance to the digit generated.

8 Conclusion

To conclude, we believe our experiments and results contributed positively to our research objective. The results assessing all parts of the model from various angles showed a very clear resemblance to the digit generated early and later throughout the model, as well as gradual change (and consequently clarity) of the features and overall image in denoising. U-Net segmentation and prediction seemed to coincide very clearly with the shape of the digit as well as different features it had earlier on in denoising and integrating gradients with respect to the input features that supported these ideas.

9 References

1. Brack, Manuel, et al. "The Stable Artist: Steering Semantics in Diffusion Latent Space." arXiv, 31 May 2023, `https://arxiv.org/pdf/2212.06013`. Accessed 21 Nov. 2023.

2. Graikos, Alexandros, et al. "Diffusion Models as Plug-and-Play Priors." arXiv, 8 Jan. 2023, `https://arxiv.org/pdf/2206.09012v3`. Accessed 20 Nov. 2023.

3. Ho, Jonathan, et al. "Denoising Diffusion Probabilistic Models." arXiv, 16 Dec. 2020, `https://arxiv.org/abs/2006.11239`. Accessed 21 Nov. 2023.

4. "How U-Net Works?" ArcGIS Developers, `https://developers.arcgis.com/python/guide/how-unet-works/`. Accessed 20 Nov. 2023.

5. Jain, Aryan. "CS 198-126: Lecture 12 - Diffusion Models." YouTube, 3 Dec. 2022, www.youtube.com/watch?v=687zEGODmHA&ab_channel=MachineLearningatBerkeley. Accessed 20 Nov. 2023.

6. Kennedy, Tom. "Chapter 7 - Markov Chain Background." Monte Carlo Methods - a Special Topics Course, 27 April 2016, https://www.math.arizona.edu/~tgk/mc/book.pdf. Accessed 15 Nov. 2023.

7. Kokhlikyan, Narine, et al. "Captum: A Unified and Generic Model Interpretability Library for Pytorch." Google Scholar, 16 Sept. 2020, https://scholar.google.com/citations?view_op=view_citation&hl=en&user=oZjHXwUAAAAJ&citation_for_view=oZjHXwUAAAAJ:W7OEmFMy1HYC. Accessed 27 Nov. 2023.

8. Krusinga, Ryen, et al. "Understanding the (Un)Interpretability of Natural Image Distributions Using Generative Models." arXiv, 25 Feb. 2019, https://arxiv.org/abs/1901.01499. Accessed 15 Nov. 2023.

9. O'Connor, Ryan. "Introduction to Diffusion Models for Machine Learning." AssemblyAI, 12 May 2022, www.assemblyai.com/blog/diffusion-models-for-machine-learning-introduction/. Accessed 15 Nov. 2023.

10. O'Sullivan, Conor. "A Step-by-Step Introduction to PCA." Towards Data Science, 14 Apr. 2020, https://towardsdatascience.com/a-step-by-step-introduction-to-pca-c0d78e26a0dd. Accessed 15 Nov. 2023.

11. Park, Ji-Hoon, et al. "Explaining Generative Diffusion Models Via Visual Analysis for Interpretable Decision-Making Process." SSRN, 20 Oct. 2023, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4608638. Accessed 20 Nov. 2023.

12. Powell, Victor. "Markov Chains." Setosa, https://setosa.io/ev/markov-chains/. Accessed 15 Nov. 2023.

13. Pulfer, Brian. "Generating Images with DDPMs: A PyTorch Implementation." Medium, 10 July 2022, https://medium.com/mlearning-ai/enerating-images-with-ddpms-a-pytorch-implementation-cef5a2ba8cb1. Accessed 22 Nov. 2023.

14. Reiff, Daniel. "Understand Your Algorithm with Grad-CAM." Towards Data Science, 21 July 2021, https://towardsdatascience.com/understand-your-algorithm-with-grad-cam-d3b62fce353. Accessed 27 Nov. 2023.

15. Rey, Luis A. Pérez, et al. "Diffusion Variational Autoencoders." arXiv, 28 Mar. 2019, arxiv.org/abs/1901.08991. Accessed 28 Nov. 2023.

16. Ross, Andrew, et al. "Evaluating the Interpretability of Generative Models by Interactive Reconstruction." ACM Digital Library, 7 May 2021, https://dl.acm.org/doi/abs/10.1145/3411764.3445296. Accessed 20 Nov. 2023.

17. Selvaraju, Ramprasaath R., et al. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization." arXiv, 3 Dec. 2019, https://arxiv.org/abs/1610.02391. Accessed 27 Nov. 2023.

18. Simonyan, Karen, et al. "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps." arXiv, 19 Apr. 2014, https://arxiv.org/abs/1312.6034. Accessed 1 Dec. 2023.

19. Sohl-Dickstein, Jascha, et al. "Deep Unsupervised Learning Using Nonequilibrium Thermodynamics." arXiv, 18 Nov. 2015, https://arxiv.org/abs/1503.03585. Accessed 28 Nov. 2023.

20. Sundararajan, Mukund, et al. "Axiomatic Attribution for Deep Networks." arXiv, 13 July 2017, https://arxiv.org/pdf/1703.01365.pdf. Accessed 20 Nov. 2023.

21. Weng, Lilian. "What Are Diffusion Models?" Lil'Log, 11 July 2021, https://lilianweng.github.io/posts/2021-07-11-diffusion-models/#nice. Accessed 15 Nov. 2023.