Scaling Inference Compute for Denoising Diffusion Probabilistic Models

Niel Ok
Stanford University
nielok@stanford.edu

Hemal Arora
Stanford University
hemal1@stanford.edu

Siya Goel
Stanford University
siyagoel@stanford.edu

Abstract

In this paper we investigate inference time scaling for Denoising Diffusion Probabilistic Models (DDPMs) as an alternative to training optimizations. We use various guided generation techniques during inference by selectively choosing noise samples, using two scoring methods including an MSE-based technique and a Mixture-Based Approach with a few labeled MNIST samples. Furthermore, we employ Top Half Search and Search Over Paths algorithms. These, when used in combination with our scoring methods, allow us to steer the backward diffusion process towards class-specific images. The experimental results indicate that our methods offer a significant enhancement over the unconditioned baseline model for generating class-specific images, with the classification accuracy of an MNIST classifier on generated images for specific classes being 71–77% with Search Over Paths and up to 85% with Top Half Search. We also found that classification accuracy and FID scores did not differ much with verifier size. These findings indicate that inference time scaling can serve as an effective alternative to training optimizations by leveraging guided sampling techniques to improve generation quality and class specificity without modifying the underlying model parameters. In the future, these algorithms should be extended to larger datasets and metrics like inception score should be investigated.

1. Introduction

Previous research in generative AI has predominantly focused on enhancing models at training-time through increasingly large datasets and model architectures. However, as training continues to scale for large-scale generative models in both language and computer vision, it is evident that focusing only on training-time improvements brings diminishing returns with increased computational costs [3]. Advancements in the past year have demonstrated that inference-time computation presents a viable alternative for scaling, as evidenced by OpenAI's o1 and o3 models and DeepSeek's R1 model in the language domain [2]. In this work, we investigate this hypothesis in the computer vision domain and specifically for Denoising Diffusion Probabilistic Models (DDPMs), which are computationally expensive and time-consuming to train.

As diffusion models continue to be applied across a range of domains, including image generation, video synthesis, and robotic motion planning, the efficient scaling of inference is essential for improving their practicality and real-world deployment. In this study, we utilize the MNIST dataset alongside a standard DDPM architecture to investigate guided generation techniques and search algorithms aimed at enhancing image generation quality. Through this approach, we introduce an inference-time scaling framework incorporating a novel scoring function, reducing reliance on computationally expensive training while increasing generation quality.

2. Related Work

Inference time adjustments have been used for a long time to improve the performance of diffusion models. Initial approaches, such as time-step scaling, increased the number of diffusion steps in DDPMs for better sample quality, while techniques like classifier guidance [1] employed an external classifier to direct the denoising. Afterwards, classifier free guidance was introduced, which relied on the diffusion model itself [4]. However, these approaches primarily focused on deterministic trajectory steering rather than explicitly exploring the denoising trajectory space.

Recent research casts inference scaling as a search problem. For example, Google DeepMind introduced a verifier guided noise search framework that samples multiple noises and selects the best from them, specifically designed for deterministic ODE solver-based diffu-

sion models. [5]. Tang et al. proposed Direct Noise Optimization (DNO), which learns the optimal initial noise from a reward function [6]. Other methods change noise trajectories during the denoising process; for instance, Yeh et al. proposed Sampling Demons, which gradually perturbs noise towards high reward regions [7], and Yoon et al. employed Monte Carlo Tree Search (MCTS) to perform online evaluation and pruning of denoising trajectories [8].

Our approach takes a different direction. Instead of relying on external verifiers or learned reward functions, we score and prune noise candidates using a small set of labeled training examples and their known forward diffusion distributions. Our method operates entirely within the standard DDPM framework, preserving its inherent stochasticity while seeking to enhance sample quality—unlike prior work which modifies deterministic ODE solvers. By leveraging test-time conditioning through noise pruning, we introduce an inference-time denoising trajectory space search method that remains faithful to the probabilistic diffusion process while improving performance. A key distinction from DeepMind's approach is that our source of variability extends beyond the initial noise samples. Because we operate within a stochastic diffusion model, it is more accurate to say we are searching over denoising trajectories rather than simply exploring a noise space. This broader exploration enables richer, data-consistent refinements during inference while preserving the generative diversity of the model.

3. Dataset and Model Training

We trained our models using the MNIST dataset. This dataset consists of 60,000 images for training and 10,000 for testing, where each image represents a black and white handwritten digit in a 28×28 pixel format. We chose to focus on the MNIST dataset as it is simple, has a large yet manageable size, and is popular.

To evaluate DDPM inference scalability with our custom noise-scoring method at a small scale, we trained a non-conditioned DDPM from scratch, drawing inspiration from (this Github). During training, MNIST images went through forward diffusion at random timesteps. Specifically, Gaussian noise was added using a fixed 1,000-timestep variance schedule. A simple UNet predicted the noise using MSE loss, optimized with AdamW (initial learning rate 0.001 decayed by a OneCycleLR scheduler over 100 epochs). We used a batch size of 128 and computed EMA weights (decay factor 0.995, updated every 10 steps) for stable inference with the EMA model.

4. Methods

4.1. Guided Generation Techniques

Our approach exploits the known forward-diffusion noise distributions of the training data by aggregating these distributions for examples belonging to a specific class of digit to construct an approximate 'class' distribution. This distribution serves as a reference for evaluating candidate noise samples, allowing us to score noise samples based on their likelihood of producing label-consistent images through reverse diffusion. Unlike traditional guidance methods, our approach conditions the model at test time by optimizing for denoising trajectories consistent with a subset of trajectories from the training distribution, rather than directly modifying the model. To evaluate the effectiveness of this test-time guidance approach, we conduct a proof-of-concept experiment on MNIST, aiming to specialize an unconditioned DDPM to generate specific digits using only a small set of labeled examples.

4.1.1 Mean Square Error (MSE) Based Approach

Since the mean of Gaussian-distributed random variables remains Gaussian, averaging the forward-diffused versions of training images for a digit results in a unimodal Gaussian distribution centered at the 'expected noised image'. This provides an approximation of the class distribution at timestep t.

To compute this, we first take the mean of the clean training images:

$$\bar{x}_0 = \frac{1}{N} \sum_{i=1}^{N} x_0^{(i)},$$

where $x_0^{(i)}$ denotes a training sample and N is the number of images used. Under the forward diffusion process, this mean propagates to timestep t as:

$$\mu_t = \sqrt{\bar{\alpha}_t}\, \bar{x}_0.$$

Instead of computing exact likelihoods, we score a partially denoised sample x_t based on its negative mean-squared error relative to the expected noised image μ_t:

$$s_{\text{MSE}}(x_t) = -\|x_t - \mu_t\|^2.$$

4.1.2 Mixture Based Approach

To account for the variability in the MNIST digits, including differences in stroke thickness, curvature, and shape, we constructed the mixture approach. This approach models each training sample as an individual

Gaussian, preserving the multimodal nature of the distribution:

$$p(x_t) = \frac{1}{N}\sum_{i=1}^{N}\mathcal{N}\left(x_t \mid \sqrt{\bar{\alpha}_t}x_0^{(i)}, (1-\bar{\alpha}_t)I\right).$$

The log-likelihood gives the corresponding score for a candidate sample:

$$s_{\text{Mixture}}(x_t) = \log\left(\frac{1}{N}\sum_{i=1}^{N}\exp\left(-\frac{\|x_t - \sqrt{\bar{\alpha}_t}x_0^{(i)}\|^2}{2(1-\bar{\alpha}_t)}\right)\right).$$

Unlike the MSE-based approach, this model does not allow for mode collapse. As a result, it allows for more informative guidance signaling for reverse diffusion. Thus, the score helps predict how well the candidate image aligns with the target distribution.

4.2. Search Algorithms

Each scoring method was tested with search algorithms to explore the denoising trajectory space more effectively than standard reverse diffusion and find optimal noise candidates.

4.2.1 Top Half Search

Top Half Search iteratively prunes candidate noise samples during reverse diffusion, retaining only the most promising ones. It begins with n noisy samples at $t = T - 1$ and updates them at each step $(t \to t - 1)$ using the model's reverse diffusion function.

At predefined timestep checkpoints, candidates are scored, with the top half retained and the rest discarded. This process continues until $t = 0$, where the highest-ranked candidate is selected as the final image. By progressively filtering lower-scoring samples, Top Half Search focuses computation on the most promising candidates.

In this study, the Top Half Search method was evaluated using 128 initial candidates, which were progressively pruned at specific timestep checkpoints: [100, 200, 300, 400, 500, 700, 900]. We tested this method across different verifier data sizes, where each size represents the number of training examples used to construct each class distribution. The verifier data sizes tested were [50, 100, 200, 600, 1000, 1400, 1800].

4.2.2 Search Over Paths

Unlike the greedy Top Half method, the Search Over Paths algorithm takes an exploratory approach by evaluating multiple denoising trajectories. It starts with n initial noise samples which undergo reverse diffusion for Δ_b steps before being scored and pruned. After denoising for Δ_b steps, the n candidates are duplicated n times (yielding n^2 candidates).

These n^2 candidates are re-noised for Δ_f steps, reinjecting noise. These n^2 are then denoised for Δ_b steps again before they are pruned back to the top n candidates in accordance with the noise sample scores. This cycle—reverse diffusing for Δ_b steps, duplicating candidates, re-noising for Δ_f steps, reverse diffusing again for Δ_b steps, and pruning back to n candidates—continues until $t = 0$, where the highest-scoring candidate is selected. By broadening the search space, this method can compare and explore denoising pathways compared to a strictly greedy approach.

In this study, Search Over Paths used 5 initial candidates, with a forward step size Δ_f of 100 and a backward step size Δ_b of 200. Verifier data sizes tested were [50, 100, 200, 600, 1000, 1400, 1800].

4.3. Experimentation

4.3.1 Baseline

We establish a baseline for our unconditioned model by generating 500 images through 1000 steps of reverse diffusion. A pretrained HuggingFace MNIST classifier then categorizes these images (which has a 99.2% accuracy), providing class frequencies without test-time conditioning.

4.3.2 Methodology

This experiment explored whether a non-label-conditioned diffusion model could be guided toward class-specific generation at test time despite the small amount of labeled training examples. Instead of using class-conditioning during training, we tested whether a model could achieve significant specialization with only a small fraction of labeled examples, far fewer than MNIST's 60,000. Success in this approach could benefit domains like biology or robotics, where labeled data is limited.

In each experiment, we generated 50 samples targeting a specific digit, varying the size of the verifier dataset (the subset of labeled training data used to compute noise scoring metrics). We then evaluated classification accuracy, defined as the proportion of generated samples correctly matching the intended digit, across different combinations of noise scoring and denoising trajectory space search methods. Classification accuracy tells us whether a pretrained classifier is able to qualitatively recognize the generated digits based on their structural quality.

We also analyzed the Frechet Inception Distance (FID) to quantitatively compare the feature embeddings of generated images with those of real MNIST images. The FID was computed by training our own CNN classifier on MNIST using a simple architecture—with three convolutional layers, ReLU activations, pooling, and dropout. We removed the final linear layer to extract penultimate feature embeddings from 500 images from the MNIST test set and 500 generated images (accumulated across all digits for a given verifier data size) for the FID calculation. The FID is defined as

$$\|\mu_r - \mu_g\|^2 + \mathrm{Tr}\left(\Sigma_r + \Sigma_g - 2\left(\Sigma_r\Sigma_g\right)^{\frac{1}{2}}\right)$$

where μ_r and μ_g are the mean feature vectors and Σ_r and Σ_g are the covariance matrices of the embeddings of real and generated images. A lower score indicates closer similarity and better image generation. By analyzing classification accuracy and FID scores over varying verifier dataset sizes, we aimed to determine the minimum amount of labeled data required to effectively condition the model at test time.

5. Results and Discussion

5.1. Baseline

In Figure 1, the baseline classification accuracy is approximately 10%, as expected. Since the non-label-conditioned diffusion model lacks guidance, it generates images without targeting a specific digit class, resulting in roughly uniformly distributed outputs across the classes.

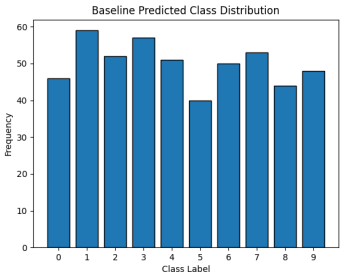
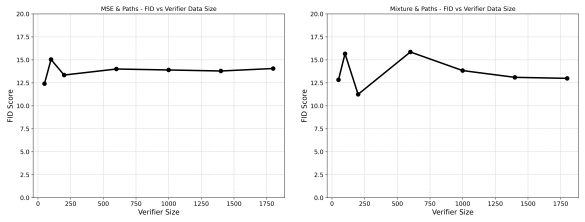


Figure 1: Histogram of Baseline Results

5.2. Comparison Between Generation Techniques for Search Over Paths

As shown in Figure 2 and Figure 3, classification accuracy and FID scores are fairly consistent with respect to the verifier size. Both MSE and Mixture guided generation achieve accuracies of 0.71–0.77 and FID scores of 12-15 for MSE and 11-16 for Mixture, respectively. Both methods reach above 70% accuracy (far better than the around 10% baseline) by leveraging search over paths with our custom verifier guidance.

We observed that the verifier size (number of labeled training examples) does not affect performance greatly, as both accuracy and FID scores remain constant across verifier sizes. This may be because the distribution used by our custom verifiers may converge in as few as 50 samples due to the simplicity and low-dimensionality of the MNIST dataset. Theoretically, this makes sense since the MSE and Mixture based scoring functions should level out and show only minor improvements once additional samples do not change the verifier distribution significantly.



(a) MSE Approach (b) Mixture Based Approach

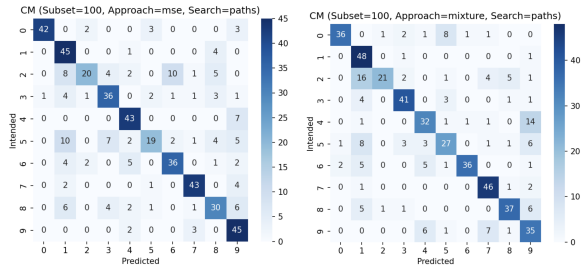
Figure 2: FID Scores Across Different Verifier Sizes for Search Over Paths

Verifier Size	MSE Accuracy	MSE FID	Mixture Accuracy	Mixture FID
50	0.740	12.4160	0.726	12.8323
100	0.718	15.0442	0.718	15.6678
200	0.756	13.3500	0.760	11.2510
600	0.726	14.0067	0.740	15.8588
1000	0.752	13.9030	0.766	13.8378
1400	0.744	13.7918	0.766	13.1011
1800	0.758	14.0624	0.752	12.9923

Figure 3: Comparison of MSE and Mixture Approaches Across Different Verifier Sizes for Search Over Paths

From Figure 3, we can see that classification accuracy and FID scores do not differ much in the MSE approach vs. Mixture approach. This may be because both approaches converge to similar high-density regions of the target distribution due to MNIST's low complexity and the robust nature of the reverse diffusion process. Thus classification accuracy and FID scores between MSE and Mixture is similar.

Figure 4 shows that digits like "2" and "5" are challenging to generate for both models. Specifically, many samples generated with the intended label of "5" are classifier by our classifier as "1" or "6," and with the



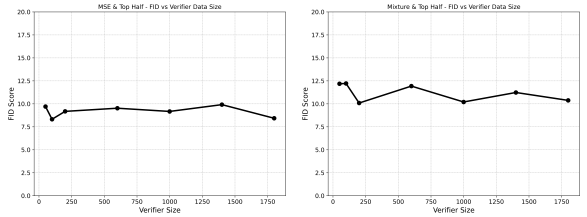
(a) MSE Approach (b) Mixture Based Approach

Figure 4: Confusion Matrices for Search Over Paths (Verifier Size 100)

mixture-based approach, many intended ”4s” are predicted as ”9s.” This occurs because the diffusion process smooths images by removing fine details, sometimes making them resemble simpler digits like ”1.” Additionally, the mixture-based approach may blend similar modes, resulting in guidance signals that transform generated ”4s” into ”9s.”

5.3. Comparison Between Generation Techniques for Top Half Search

From Figure 5 and Figure 6 we can see that the range of accuracy for MSE was 0.75-0.80 and Mixture was 0.79-0.85. Additionally, we saw that the range of FID scores for MSE was 8-10 and for Mixture was 10-12.5. Therefore, for similar reasons, such as smoothness in the reverse diffusion process and simplicity of the MNIST dataset, we saw similar accuracy and FID scores across different verifier sizes.



(a) MSE Approach (b) Mixture Based Approach

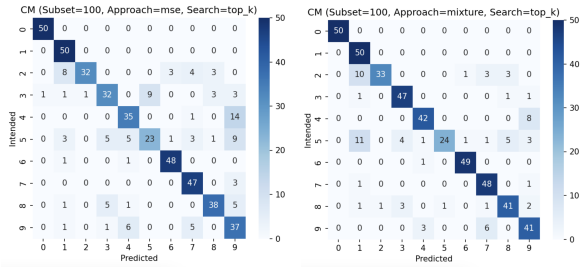
Figure 5: FID Scores Across Different Verifier Sizes for Top Half

From Figures 5 and 6, we see that MSE produces lower FID scores than the Mixture approach, suggesting that its images are statistically closer to overall MNIST distribution than the others. This may happen because the MSE pulls the generated images towards the mean of the training examples thus ensuring that there is a good match with global statistics. However, from Figure 7, we see that the Mixture approach outperforms the other in terms of classification accuracy. We think this improvement is achieved because, although the mixture method leads to a slightly higher FID, it preserves more of the selective/important details specific of each digit for accurate semantic recognition. In conclusion, although MSE gives a better quantitative fit to the overall distribution, the mixture approach improves recognizability by preserving important nuances.

As shown by Figures 3 and 6, the Top Half method outperforms the search-over-paths approach in terms of classification accuracy and FID scores. However, these results are not comparable as we do not know how each approach scales with compute, at least for stochastic diffusion models like our DDPM.

Verifier Size	MSE Accuracy	MSE FID	Mixture Accuracy	Mixture FID
50	0.778	9.7094	0.796	12.1861
100	0.784	8.3248	0.850	12.2174
200	0.802	9.1820	0.798	10.0897
600	0.756	9.5254	0.810	11.9307
1000	0.794	9.1754	0.800	10.2044
1400	0.796	9.9133	0.798	11.2343
1800	0.790	8.4322	0.788	10.3777

Figure 6: Comparison of MSE and Mixture Approaches Across Different Verifier Sizes for Top Half



(a) MSE Approach (b) Mixture Based Approach

Figure 7: Confusion Matrices for Top Half (Verifier Size 100)

6. Conclusion and Future Work

Our results show that incorporating additional guidance through generation and sampling techniques significantly improves classification accuracy to over 70%, with FID scores ranging from 11 to 16— a substantial improvement over the 10% baseline. Interestingly, classification accuracy and FID scores remained largely unchanged across different verifier sizes. Among the methods tested, the MSE approach achieved lower FID

scores, indicating a closer statistical match to MNIST, while the Mixture approach enhanced recognizability, leading to slightly higher classification accuracy.

Moving forward, we aim to collect more data to strengthen the robustness of our claims, as computational constraints limited the number of generated images. While Top Half search outperformed Search Over Paths in our experiments, we hypothesize that Search Over Paths will scale better for DDPMs, similar to its behavior in ODE solvers as observed in the DeepMind paper. Generating more images will also improve the reliability of our FID measurements. Additionally, we plan to benchmark our method's compute-scaling efficiency by investigating Neural Function Evaluations (NFEs) and FLOPs.

Given our method's ability to guide model outputs effectively, we seek to extend our approach to larger datasets such as CIFAR-10 and ImageNet. Expanding our evaluation metrics to include inception score and precision-recall will provide deeper insights into image quality and diversity. We also aim to refine verifier approaches by integrating localized distributions (pixel-level) with our current global approach (full-image analysis).

Beyond image generation, our method of leveraging forward distributions from a small subset of labeled examples holds promise for domains with relatively large amounts of unlabeled data. This is especially true when there isn't enough labeled data to train a label-conditioned model from scratch. A particularly compelling application lies in robotics, where scaling inference compute for diffusion policies could improve motion planning accuracy and adaptability.

We will also do more theory work, analytically determining the size of noise space vs denoising trajectory space and verifying that inference search algorithms are able to adequately direct denoising trajectories in the larger denoising trajectory space.

7. Contributions & Acknowledgements

First and foremost, we gratefully acknowledge compute support provided by researchers at DeepMind. In this project Niel, Hemal, and Siya each played pivotal roles in developing the algorithms in this project. Niel focused on training the models and vectorizing the code to run large scaling studies, as well as the theoretical formalization for trajectory space search. Hemal developed the foundations of the experiment and ran large scale studies. Siya ran small scale studies and constructed baselines. Collectively, we implemented these algorithms, gathered results, and meticulously documented our findings.

References

[1] P. Dhariwal and A. Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[2] Y. Fan and K. Lee. Optimizing ddpm sampling with shortcut fine-tuning. *arXiv preprint arXiv:2301.13362*, 2023.

[3] R. Gozalo-Brizuela and E. C. Garrido-Merchan. Chatgpt is not all you need. a state of the art review of large generative ai models. *arXiv preprint arXiv:2301.04655*, 2023.

[4] J. Ho and T. Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[5] N. Ma, S. Tong, H. Jia, H. Hu, Y.-C. Su, M. Zhang, X. Yang, Y. Li, T. Jaakkola, X. Jia, et al. Inference-time scaling for diffusion models beyond scaling denoising steps. *arXiv preprint arXiv:2501.09732*, 2025.

[6] Z. Tang, J. Peng, J. Tang, M. Hong, F. Wang, and T.-H. Chang. Inference-time alignment of diffusion models with direct noise optimization. *arXiv preprint arXiv:2405.18881*, 2024.

[7] P.-H. Yeh, K.-H. Lee, and J.-C. Chen. Training-free diffusion model alignment with sampling demons. *arXiv preprint arXiv:2410.05760*, 2024.

[8] J. Yoon, H. Cho, D. Baek, Y. Bengio, and S. Ahn. Monte carlo tree diffusion for system 2 planning. *arXiv preprint arXiv:2502.07202*, 2025.