Diffusion Models as Plug-and-Play Priors: Conditional Generation on MNIST

Master of Science Student
Merve Pakcan Tufenk
National University of Science and Technology Politehnica Bucharest
Faculty of Automatic Control and Computers (Romania)

March 2025

1 Introduction

Denoising diffusion probabilistic models (DDPMs) have emerged as powerful generative models capable of modeling complex data distributions. Traditionally, DDPMs are retrained as stand-alone generators for each task. To overcome this limitation, a plug-and-play framework using pretrained DDPMs as conditional priors was introduced [1], enabling controllable generation through auxiliary differentiable constraints without modifying model weights.

This report reproduces a subset of the original experiments on the MNIST dataset. A U-Net-based DDPM is trained, and inference is guided by a handcrafted horizontal dissimilarity constraint. The input noise vector is optimized using gradients from both the model and auxiliary loss, while model weights remain fixed.

The results show that pretrained DDPMs can be effectively steered using simple constraints, even in low-dimensional settings. These findings confirm the feasibility of plug-and-play inference and support the original claims in a simplified setup, demonstrating practicality for controllable generation in low-data or task-specific scenarios without retraining.

2 Method

The plug-and-play diffusion framework was adopted in this study as introduced in [1]. A DDPM is implemented with a U-Net denoiser, using 1 input/output channel and a 32×32 resolution. The architecture includes 2 residual blocks per resolution, attention at spatial scales 8 and 4, channel multipliers (1, 2, 3, 4), and 4 attention heads.

Training is performed on the MNIST dataset. Grayscale 28×28 images are padded to 32×32 and normalized to [-1, 1]. A batch size of 128 is used. The forward diffusion process follows a linear 1000-step noise schedule. The model is trained to predict added noise using mean squared error (MSE) loss.

During inference, a plug-and-play approach optimizes only a latent noise vector while DDPM weights stay fixed. The loss merges denoising with a handcrafted horizontal dissimilarity term, promoting vertical asymmetry, and gradients update the noise vector without altering model weights.

All experiments were run on Google Colab. Training used a T4 GPU, while inference was performed on CPU due to limited GPU availability, which was challenge that slowed the process.

3 Experiments

The controllability of pretrained DDPMs is evaluated through inference experiments on the MNIST dataset. The model is trained as described in Section 2, using a 1000-step linear schedule and 32×32 padded inputs.

Instead of unconditional sampling, a handcrafted horizontal dissimilarity constraint is introduced to encourage structural asymmetry across the vertical axis. This auxiliary loss is combined with the denoising objective, and the total loss is minimized with respect to the input noise vector using gradient-based optimization.

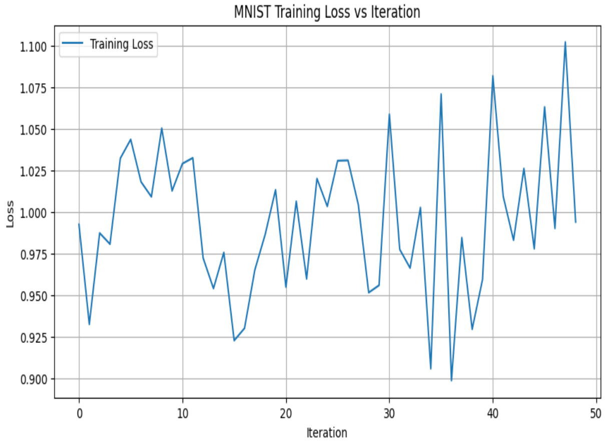
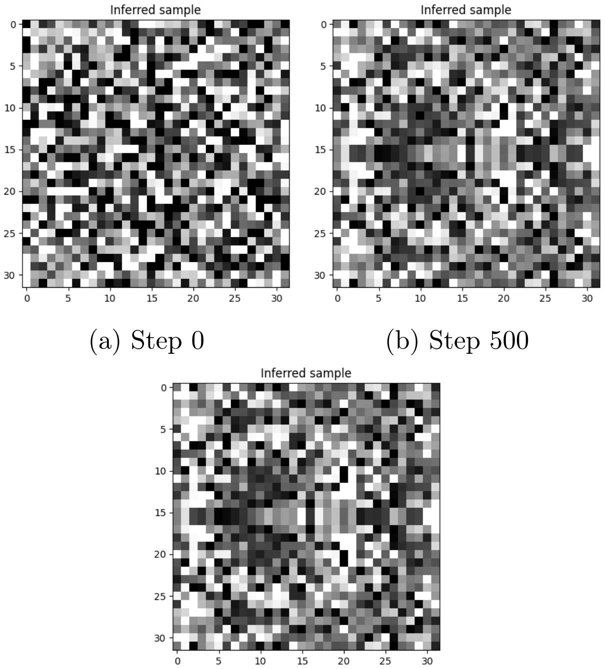


Figure 1: MNIST training loss vs. iterations.

As shown in Figure 1, training loss remains within a stable range despite fluctuations caused by the interaction of diffusion and auxiliary gradients. This stability indicates convergence toward a structured solution space.

As shown in Figure 2, the samples evolve from unstructured noise to asymmetric patterns, reflecting the influence of the imposed constraint. This suggests that pretrained diffusion models can be steered toward target structures using simple auxiliary losses, without retraining.



(a) Step 0

(b) Step 500

(c) Step 1000

Figure 2: Inferred samples at optimization steps.

Code and results can be accessed: Colab link.

4 Conclusion

This study shows that pretrained DDPMs can be guided with simple constraints for controllable generation without retraining, validated in low-data settings like MNIST.

For future work, this approach could be extended to more complex or high-dimensional datasets used in the original study, such as FFHQ or TSP or applied to real-world scenarios requiring structure-aware sampling in limited-data environments.

References

[1] Graikos, A., Malkin, N., Jojic, N., & Samaras, D. (2022). *Diffusion models as plug-and-play priors*. Advances in Neural Information Processing Systems, 35, 14715–14728.