Denoising Score Matching

Victorin Turnel & Thomas Winninger
Master MVA, ENS Paris-Saclay

1 Introduction

Many recent approaches like DALL-E, Midjourney, or Stable Diffusion have shown outstanding performance using denoising processes, establishing these methods as a standard for data generation. Among these methods, denoising score matching is considered an efficient method to address challenges raised by likelihood-based methods and GANs for image generation.

In this project, we will focus on score matching techniques and, more precisely, on their application in generative models through the study of the works of (Vincent, 2011) and (Song and Ermon, 2020a).

First, we present a brief overview of the main contributions from the two articles. Then, we reproduce the main experiments of (Song and Ermon, 2020a) by implementing everything from scratch. Finally, we propose some improvements to enhance the data generation quality.

2 Score matching

Energy-Based Models (EBMs) represent a standard framework to address the challenge of learning the parameters θ of probability density models $p_\theta(x)$. The core idea is to associate each configuration with an energy score representing its plausibility: a low energy corresponds to a plausible configuration, and vice versa. The model defines the density via the Boltzmann distribution: $p_\theta(x) = \frac{\exp(-E_\theta(x))}{Z_\theta}$, where E_θ is the energy function, and Z_θ is the normalization constant, or the partition function.

While the energy function can be easily approximated by a neural network, calculating the partition function remains challenging for high-dimensional data like images, a challenge which has long limited their practical use. Score matching (Hyvärinen, 2005) circumvents the partition function by working with the **score function**, the gradient of the log-density:

$$\psi(x,\theta) = \nabla_x \log p(x,\theta) = \nabla_x \log \frac{\exp(-E_\theta(x))}{Z_\theta} = -\nabla_x E_\theta(x) \tag{1}$$

Crucially, the partition Z_θ vanishes when taking gradients since it doesn't depend on x. Following this principle, θ is learned so that $\psi(x,\theta)$ best matches the score of the true distribution, denoted $q(x)$: $\nabla_x \log q(x)$. Then, the problem can be formalized as the minimization of the following objective function:

$$J(\theta) = \mathbb{E}_{q(x)}\left(\frac{1}{2}\|\psi(x,\theta) - \nabla_x \log q(x)\|^2\right) \tag{2}$$

Although this expression does not imply Z_θ, it involves knowing the explicit expression of $\nabla_x \log q(x)$, which is not available in practice. However, (Hyvärinen, 2005) proved that the previous objective function can be written without the explicit score target:

$$J(\theta) = \mathbb{E}_{q(x)}\left(\frac{1}{2}\|\psi(x,\theta) - \nabla_x \log q(x)\|^2\right) = \mathbb{E}_{q(x)}\left(\frac{1}{2}\|\psi(x,\theta)\|^2 + \sum_{i=1}^{d}\frac{\partial \psi_i(x,\theta)}{\partial x_i}\right) + C \tag{3}$$

Where $\psi_i(x,\theta) = \frac{\partial \log p(x,\theta)}{\partial x_i}$ and C is a constant independent of θ. This new expression defines an equivalent optimization objective and enables learning θ in real-world scenarios.

In practice, we sample $x \sim q(x)$ from our dataset, compute $\psi(x, \theta)$ via automatic differentiation, and optimize θ via gradient descent. The partition function never appears in computation.

3 Efficient Score Matching Techniques

As seen in the previous section, the implicit score matching objective involves calculating the trace of a Jacobian matrix, a $d \times d$ matrix for d-dimensional data. This computational cost $O(d^2)$ makes it prohibitive for high-dimensional input such as images. To overcome this limitation, two major efficient variants have been proposed: Denoising Score Matching and Sliced Score Matching.

3.1 Denoising Score Matching and the Autoencoder Connection

The first method is **Denoising Score Matching (DSM)** proposed by (Vincent, 2011). It aims to bypass the computation of the Hessian trace by corrupting the data with a noise distribution, and learning the denoising process instead. Consider corrupting the data with Gaussian noise:

$$\tilde{x} = x + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2 I) \tag{4}$$

This defines a noise-conditional distribution $q_\sigma(\tilde{x} \mid x) = \mathcal{N}(\tilde{x}; x, \sigma^2 I)$ and a marginal $q_\sigma(\tilde{x}) = \int q_\sigma(\tilde{x} \mid x) q(x)$ dx.

A denoising autoencoder learns to predict the clean x from corrupted $\tilde{x}$. Vincent showed that minimizing the denoising objective:

$$J_{\text{DSM}}(\theta) = \frac{1}{2}\mathbb{E}_{q_{\sigma(\tilde{x},x)}}\left[\frac{1}{2}\left\|\psi(\tilde{x}, \theta) - \nabla_{\tilde{x}} \log q_{\sigma(\tilde{x} \mid x)}\right\|_2^2\right] \tag{5}$$

is equivalent to score matching on the noise-perturbed distribution $q_\sigma(\tilde{x})$. Specifically, it shows that the optimal score is proportional to the reconstruction error of an autoencoder trained to denoise $\tilde{x}$ back to x: $\psi(\tilde{x}, \theta) \approx \frac{r(\tilde{x}) - \tilde{x}}{\sigma^2}$. Where $r(\tilde{x})$ is the reconstruction provided by the autoencoder. This connection provides a powerful intuition: learning the score of the data is equivalent to learning to project noisy data points back onto the data manifold.

To understand this equivalence, it is beneficial to look at the derivation provided by (Vincent, 2011). The authors define a specific energy-based model where the energy function $E(x; \theta)$ corresponds to a neural network with a single hidden layer and tied weights:

$$E(x; \theta) = -\frac{\langle c, x\rangle - \frac{1}{2}\|x\|^2 + \sum_{j=1}^{d_h} \text{softplus}\left(\langle W_j, x\rangle + b_j\right)}{\sigma^2} \tag{6}$$

From this energy function, we can compute the model's score $\psi(x, \theta) = -\nabla_x E(x; \theta)$. By deriving the expression with respect to x, we obtain:

$$\psi(x, \theta) = \frac{1}{\sigma^2}(W^T \text{ sigmoid}(Wx + b) + c - x) \tag{7}$$

We recognize the term W^T sigmoid$(Wx + b) + c$ as the reconstruction function $r(x)$ of a standard autoencoder. Thus, the score is simply a scaled residual of the reconstruction: $\psi(x, \theta) = \frac{1}{\sigma^2}(r(x) - x)$. Finally, by injecting this expression and the explicit score of the Gaussian noise kernel ($\nabla_{\tilde{x}} \log q_{\sigma(\tilde{x}|x)} = \frac{x - \tilde{x}}{\sigma^2}$) into the DSM objective, we obtain:

$$J_{\text{DSM}}(\theta) = \mathbb{E}\left[\frac{1}{2}\left\|\frac{1}{\sigma^2}(r(\tilde{x}) - \tilde{x}) - \frac{1}{\sigma^2}(x - \tilde{x})\right\|^2\right] = \frac{1}{2\sigma^4}\mathbb{E}[\|r(\tilde{x}) - x\|^2] \tag{8}$$

This demonstrates that minimizing the Denoising Score Matching objective is strictly equivalent (up to a constant factor) to minimizing the Euclidean reconstruction error of a Denoising Autoencoder.

3.2 Sliced Score Matching

Another efficient alternative is **Sliced Score Matching (SSM)**. Instead of matching the score of perturbed data, SSM reduces the computational complexity of the implicit score matching objective by projecting the score onto random vectors v sampled from a distribution p_v.

The objective function uses these random projections to estimate the expensive trace term, reducing the cost to $O(d)$:

$$J_{\text{SSM}}(\theta) = \mathbb{E}_{p_v}\left[\mathbb{E}_{q(x)}\left[\frac{1}{2}\|\psi(x,\theta)\|^2 + v^T(\nabla_x \psi(x,\theta))v\right]\right] \tag{9}$$

By averaging this objective over random directions v, it can efficiently approximate the true score matching objective.

4 Sampling with Annealed Langevin Dynamics

Once we have trained a score network $\nabla_x \log p_\sigma(x) \approx \nabla_x \log q_\sigma(x)$, we can generate samples with a method named Langevin dynamics, which aims to sample a random point, and walk toward high-density regions to improve the sample step by step.

4.1 Langevin Monte Carlo

(Song and Ermon, 2020a) describes a recursive Langevin, where one starts with $\tilde{x}_0 \sim \mathcal{N}(0, I)$, and iterate:

$$\tilde{x}_t = \tilde{x}_{t-1} + \frac{\varepsilon}{2}\nabla_x \log p(\tilde{x}_{t-1}) + \sqrt{\varepsilon} z_t, \quad z_t \sim \mathcal{N}(0, I) \tag{10}$$

where $\varepsilon > 0$ is a step size. The core idea is to iteratively move towards high-density areas, following the direction indicated by the gradient, while the noise z_t ensures proper exploration. As $\varepsilon \to 0$ and $T \to \infty$, the distribution $\tilde{x}_T$ converges to $q(x)$.

However, this naive approach faces two significant obstacles. First, data of interest typically reside on low-dimensional manifolds embedded in a high-dimensional space (e.g., the space of all images). Consequently, the score is particularly hard to estimate in low-density regions, leading to major convergence issues. Second, when multiple modes are separated by low-density regions, the standard Langevin method struggles to recover the correct relative weights of these modes in the distribution.

4.2 Annealed Langevin Dynamics

To overcome these limitations, the proposed solution is to use **Annealed Langevin Dynamics**: corrupt the data with a sequence of noise levels that gradually decrease as the sampling process approaches the data manifold (Song and Ermon, 2020a). Indeed, data perturbed by large Gaussian noise populates the ambient space rather than being confined to a low-dimensional manifold resulting in a well-defined score function everywhere.

Starting with large noise σ_1, the score is reliable everywhere. As we anneal to smaller noise levels, we refine samples toward the true data manifold.

Algorithm 1: Annealed Langevin Dynamics

```
1  $x_0 \sim \mathcal{N}(0, \sigma_1^2 I)$
2  for $i = 1$ to $L$ do
3      $\alpha_i \leftarrow \varepsilon \frac{\sigma_i^2}{\sigma_L^2}$
4      for $t = 1$ to $T$ do
5          $z_t \sim \mathcal{N}(0, I)$
6          $\tilde{x}_t \leftarrow \tilde{x}_{t-1} + \frac{\varepsilon}{2}\nabla_x \log(p(\tilde{x}_{t-1})) + \sqrt{\varepsilon} z_t$
7      end
8      $\tilde{x}_0 \leftarrow \tilde{x}_T$
9  end
10 return $\tilde{x}_T$
```

This multi-scale approach ensures both good mixing (large σ) and high-quality samples (small σ). The temperature-like annealing schedule $(\sigma_1, \sigma_2, ..., \sigma_L)$ is typically geometric: $\frac{\sigma_i}{\sigma_{i+1}} = \gamma$ for $\gamma > 1$.

5 Experiments

In this section, we will go through the different experiments carried out in this project. The goal of these experiments it to reproduce the main results of (Song and Ermon, 2020a) in order to visualize the notion of score, sampling and the applications on data generation tasks with a simplified architectures.

5.1 Experiment 1 : Toy examples for score and sampling visualization

The goal of this experiment is to visualize the score associated to a 2D-distribution as a vector field. Secondly, we aim at emphasizing the difference between the classical Langevin Dynamics and the annealed Langevin dynamics.

Setup. To do that, we first implemented a data generator able to generate two types of distributions : the one used in (Song and Ermon, 2020a) $p_{\text{data}} = \frac{1}{5}\mathcal{N}((5,5), I) + \frac{4}{5}\mathcal{N}((-5,-5), I)$ and a new "circle" distribution. To approximate the score of these distribution we implemented the same MLP, made of 3 linear layers with a softplus activation function, as the original example. Finally, in order to check the relevancy of the annealed Langevin dynamic, two sampling methods were implemented and compared.

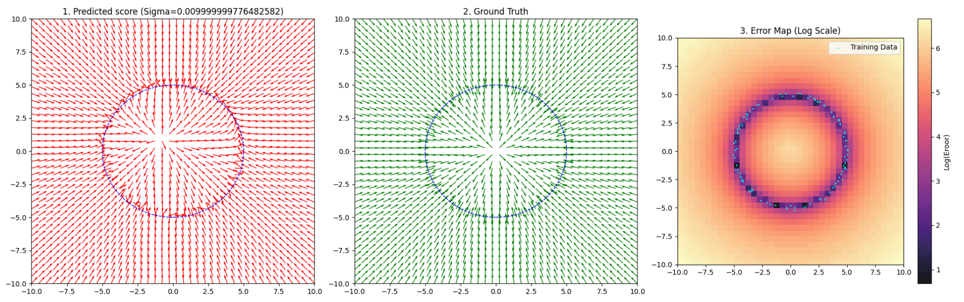


Figure 1: (1) Predicted score for the circle distribution (2) Ground truth score of the circle distribution (3) Error between the predicted score and the ground truth

Results. Figure 1 shows that, when visualizing the representation as a vector field of the computed score, the vectors point correctly in the direction of the area with the highest data density, revealing the circular shape of the distribution. However, as emphasized in (Song and Ermon, 2020a), the vector field is consistent for points close to the training data (low-dimensional manifold) but does not seem relevant in the whole space. This phenomenon is particularly visible in the middle of the circle and for points far from the circle. These two areas do not contain training data.

The second result (Figure 2) confirms the validity of annealed Langevin dynamics. Indeed, the classical approach is unable to recover the relative density between the two modes. However, the sampling method presented in (Song and Ermon, 2020a) successfully preserves this proportion. In contrast to the experiment carried out in the original article, where sampling used the ground truth, we conducted the sampling directly on the score predicted by our MLP.

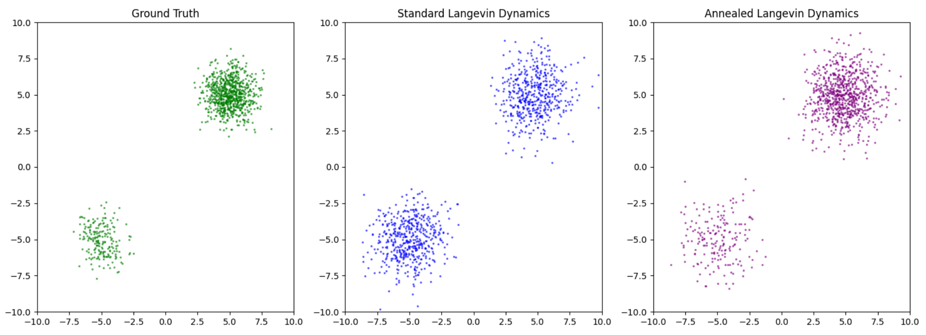


Figure 2: (Left) Ground truth distribution (Middle) Sampling with Langevin Dynamics (Right) Sampling with Annealed Langevin Dynamics

5.2 Experiment 2: Simple U-net for MNIST

For this experiment on MNIST, we implemented a simplified U-Net architecture from scratch, designed to be significantly lighter than the RefineNet used in the original paper. Optimized for fast training, this model deviates from the complexity of the reference paper while retaining the fundamental principle of conditioning on the noise level σ via a dedicated embedding module injected at each convolution step.

To move beyond simple visual inspection, we integrated the calculation of the Fréchet Inception Distance (FID) directly into our training loop. This metric allows us to objectively measure the distance between the distribution of real images and that of images generated by Langevin dynamics.

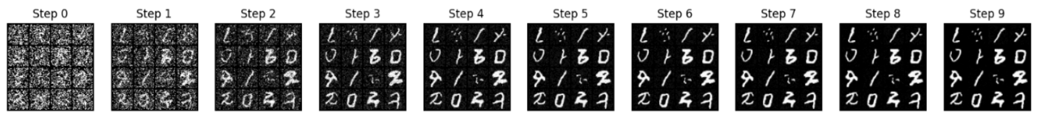


Figure 3: Visualization of the sampling steps. The sampling was performed using the predicted score at epoch 15. While some digits are recognizable, the results lack consistency.

We obtained satisfactory results on the MNIST dataset by training the model for 50 epochs ($\approx$ 6min) with a learning rate of $1e-3$ and a batch size of 128. However, generating consistent samples remains challenging due to the instability of the visual quality. Indeed, this can likely be explained by the simplicity of the architecture. The model consists of a shallow U-Net architecture with skip connections. It relies on standard convolutional layers paired with Group Normalization and SiLU activations.

5.3 Experiment 3 : Improvements on the simple U-net for MNIST

The main advantage of the previous architecture lies in its efficiency, offering a good trade-off between decent results and rapid training times ($\approx$ 7s for one epoch on an RTX 5070). However, we observed the instability phenomenon described in (Song and Ermon, 2020b). While the loss function decreases steadily throughout training, the visual quality of the generated samples remains highly unstable.

This observation is confirmed quantitatively by the fluctuating FID scores in Table 1, and qualitatively by the presence of characteristic artifacts across samples generated from the same checkpoint.

Table 1: Comparison of the loss and the FID score between the previous customized U-net (**Cust**) and the customized U-net with EMA (**EMA**)

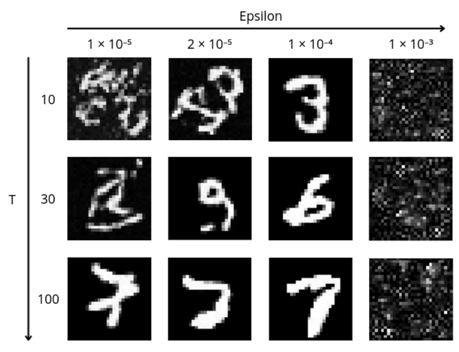
Epoch	FID (Cust) ↓	FID (EMA) ↓	Loss (Cust)	Loss (EMA)
1	-	-	0.4502	0.3068
5	2.9746	0.6112	0.2095	0.1288
10	11.3181	0.3198	0.1609	0.1058
15	0.9717	0.2249	0.1471	0.0971

To address this, we enhanced the stability of the vanilla NCSN by implementing an Exponential Moving Average (EMA) of the weights, as proposed in (Song and Ermon, 2020b). Specifically, let θ_i denote the model parameters at the i-th training iteration and θ' be an independent copy of these parameters. We update θ' using the rule $\theta' \leftarrow m\theta' + (1-m)\theta_i$ after each step, where m is the momentum parameter (typically close to 1). As we can see in Table 1, the FID Score of this new approach decreases steadily.

5.4 Experiment 4: grid search

As noted in the previous section, achieving high quality with NCSN is challenging due to the instability of generated samples. This difficulty is mainly compounded by the sensitivity of the results to the step size ε and the number of steps T. Therefore, we conduct a visual grid search to analyze how variations in these parameters affect the perceptual quality of the samples.

Our experiments reveal two distinct failure cases. First, setting a small ε with an insufficient number of steps T results in noisy images where digits are barely recognizable (top left). In this scenario, the Langevin dynamics are stable but too slow: the process fails to fully traverse the distance from the initial random noise to the data manifold. Conversely, using an excessively large step size ε causes the process to overshoot and oscillate around the manifold. This leads to this "snow-like" texture where no structure is discernible (right column). Therefore, the optimal results are obtained by combining a small ε (to ensure a precise convergence) with a large number of step T to allow the process sufficient time to reach the high-density regions.



This experiment confirms the observation of (Song and Ermon, 2020a), when $\varepsilon \to 0$ and $T \to \infty$ the samples approximate those of the original distribution.

5.5 Experiment 5 : Image generation on Fashion MNIST, and analysis of model collapse

For the second task, we took the model architecture of (Song and Ermon, 2020a) and trained it on the MNIST (LeCun et al., 1998) and Fashion MNIST (Xiao et al., 2017) datasets, as the Fashion MNIST dataset was designed to be harder than MNIST, but still with a complexity in a similar order of magnitude.

Setup. For the following experiments, we used our custom U-Net (*custom U-Net*), the U-Net defined in (Song and Ermon, 2020a) (*original U-Net*), and a slightly larger version with a bigger embedding dimension and dropout layers (*large U-Net*). The hyperparameters are similar, and the number of epochs follows the same order of magnitude (10000-100000). To test the performance of the model, we use the FID score (Heusel et al., 2018).

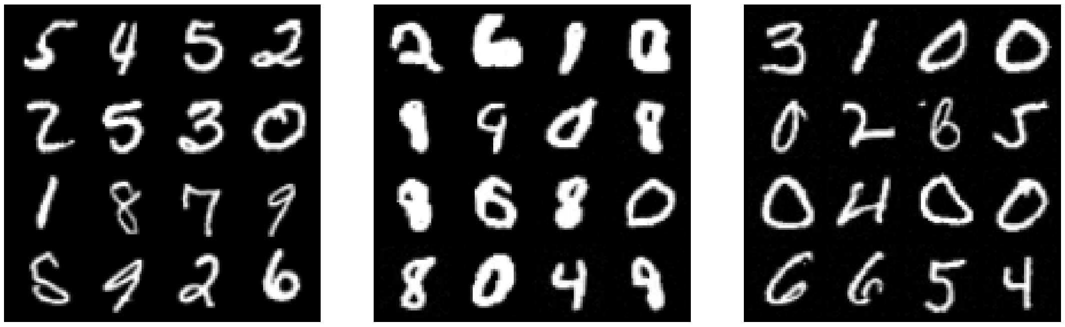


Figure 5: **Left:** Images sampled from the MNIST dataset. **Middle:** Images generated with the model trained for 30000 epochs. **Right:** Images sampled with the model trained for 100000 epochs.

All models, including our *custom U-Net* performed well on the MNIST dataset (Appendix Figure 5), however, neither our *custom U-Net*, nor the *original U-Net* were able to generate proper images. More intriguing, they both generated samples uniquely from one class: shirts. As shown in Figure 7, this looks like an overfitting pattern. This is coherent with the FID score, decreasing at the start of the training, and then increasing after 40000 epochs (Appendix Figure 6).

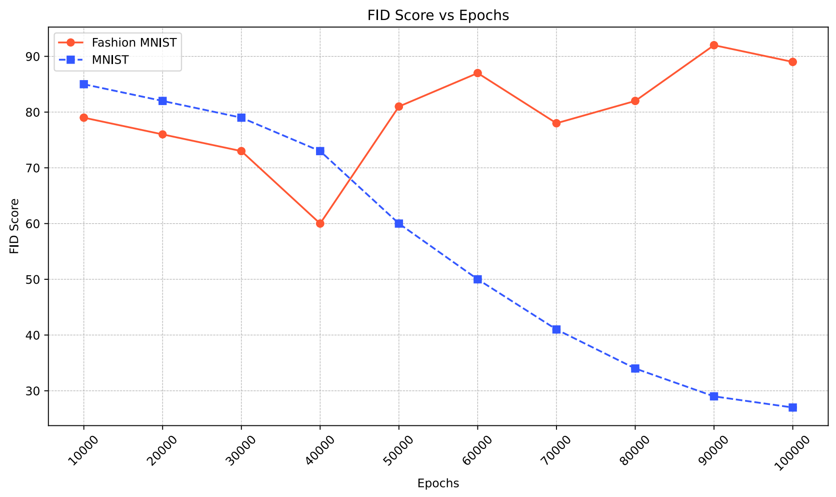


Figure 6: FID scores throughout the training process.

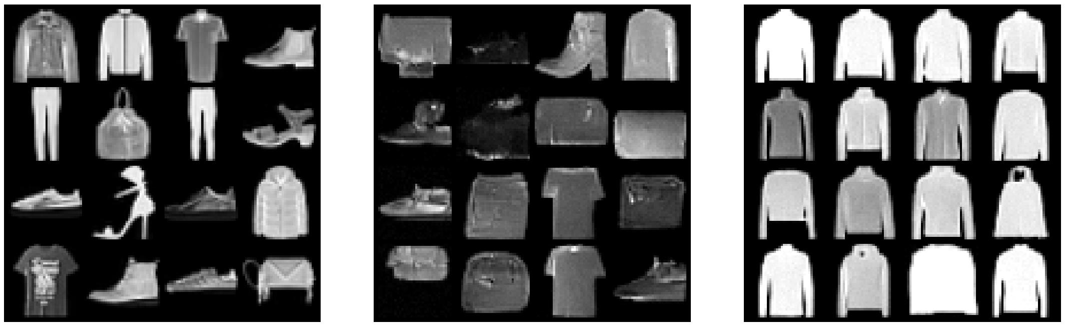


Figure 7: **Left:** Images sampled from the FashionMNIST dataset. **Middle:** Images generated with the original U-Net trained for 10000 epochs. **Right:** Images sampled with the original U-Net trained for 20000 epochs.

To test if the model was overfitting, we increased the number of parameters and added dropout layers to create *large U-Net*. While the results are better (Figure 8), there was still an issue at some point during the training. An explanation is given by (Favero et al., 2025), who showed that, even in overparametrized models, models could reach generalization before collapsing in later stages of the training.

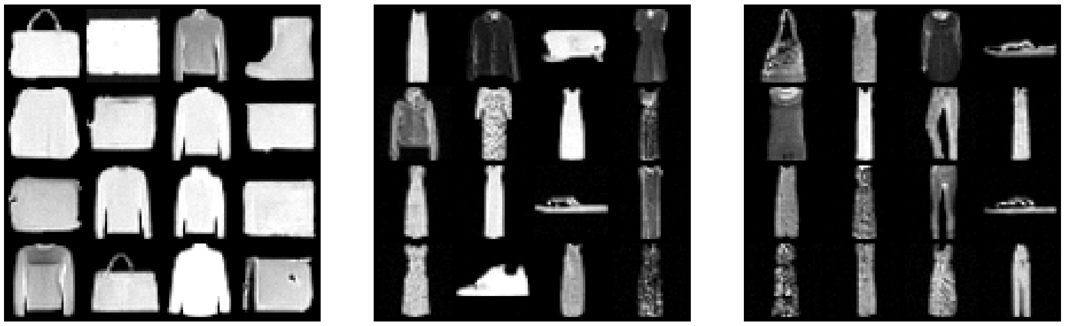


Figure 8: **Left:** *large U-Net* at 40000 epochs. **Middle:** *large U-Net* with larger σs. **Right:** *large U-Net* with larger σs and early stopping.

Also, (Bonnaire et al., 2025) provides an explanation to why the model collapse much faster on Fashion-MNIST – at epoch 40000 for *large U-Net* – while it is still improving at this stage on the MNIST task. They showed that diffusion models had two stages: generalization then memorization, and that datasets with higher complexity reached the memorization stage faster. Hence, a solution is to combine early stopping with larger sigmas to slow down the memorization, to add EMA, or use recent results on entropy-based data selection (Shi et al., 2025).

6 Conclusion

In this project, we reproduced the experiments of (Song and Ermon, 2020a) while investigating two distinct model configurations. We developed a small custom U-Net, optimized with Exponential Moving Average (EMA) to enhance its capabilities, and tested the original paper's architecture modified with dropout layers. This allowed us to explore the role of regularization and model capacity in the diffusion process.

The primary challenge we faced was the hyperparameter tuning, which proved to be both complex and unstable. The important number of sensitive variables, including the noise schedule (σ_{min}, σ_{max}) and sampling parameters (ε, T), often made it difficult to distinguish between model collapse and implementation errors. We mitigated this uncertainty by sampling frequently during training, which allowed us to closely monitor stability.

While we validated the effectiveness of Denoising Score Matching, even on very small models, for the MNIST task, we quickly ran into one of the most significant challenges in scaling these methods to more complex data distributions: model collapse. This aligns with the active recent research on the subject, suggesting that diffusion models eventually shift from generalization to memorization. Hence, just building larger models (like our *large U-Net*) is not a miracle solution. Future work may focus on implementing dynamic regularization techniques, such as entropy-based data selection or specific early-stopping criteria.

References

Bonnaire, T., Urfin, R., Biroli, G., and Mézard, M. *Why Diffusion Models Don't Memorize: The Role of Implicit Dynamical Regularization in Training*, 2025. https://arxiv.org/abs/2505.17638

Favero, A., Sclocchi, A., and Wyart, M. *Bigger Isn't Always Memorizing: Early Stopping Overparameterized Diffusion Models*, 2025. https://arxiv.org/abs/2505.16959

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. *GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium*, 2018. https://arxiv.org/abs/1706.08500

Hyvärinen, A. Estimation of Non-Normalized Statistical Models by Score Matching. *Journal of Machine Learning Research*, *6*(24), 695–709, 2005.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324, 1998. https://doi.org/10.1109/5.726791

Shi, L., Wu, M., Zhang, H., Zhang, Z., Tao, M., and Qu, Q. *A Closer Look at Model Collapse: From a Generalization-to-Memorization Perspective*, 2025. https://arxiv.org/abs/2509.16499

Song, Y., and Ermon, S. *Generative Modeling by Estimating Gradients of the Data Distribution*, 2020a.

Song, Y., and Ermon, S. *Improved Techniques for Training Score-Based Generative Models*, 2020b. https://arxiv.org/abs/2006.09011

Vincent, P. A Connection Between Score Matching and Denoising Autoencoders. *Neural Computation*, *23*(7), 1661–1674, 2011. https://doi.org/10.1162/NECO_a_00142

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *Arxiv Preprint Arxiv:1708.07747*, 2017.