DENOISING SCORE MATCHING FOR DIFFUSION MODELS

REPORT

Manon Gouttefangeas
ENS Paris-Saclay
Manon.gouttefangeas@ens-paris-saclay.fr

Mohammed Raki
ENS Paris-Saclay
Mohammed.Raki@ens-paris-saclay.fr

Lucas Versini
École polytechnique
lucas.versini@polytechnique.edu

ABSTRACT

This report delves into generative models, with a particular focus on score-based models. Much of our analysis is drawn from the works of Song and Ermon [2019] and Vincent [2011].

We begin by introducing score-based generative models and comparing them to traditional models that directly estimate the underlying probability distribution. We also introduce key techniques such as Sliced Score Matching and Langevin dynamics.

Next, building on Song and Ermon [2019], we explore the key challenges in approximating the score of a distribution and examine the solutions proposed to address these challenges, including Denoising Score Matching.

Throughout the report, we illustrate these concepts using both synthetic (toy) datasets and real-world datasets, highlighting the practical applications and effectiveness of score-based models.

1 Introduction

Generative models are a class of machine learning models designed to learn a probability distribution from observed samples. The main goal of these models is to estimate the probability distribution $p(x)$ over the data, in order to generate new samples that resemble the real data. In other words, given a set of training examples $D_n = \{x_1, \ldots, x_n\}$ assumed to be drawn from an unknown density p, the aim of generative models is to learn $p(x)$, or more generally the structure of the data, so they can then generate new data points that are similar to D_n.

Several approaches to model the unknown probability distribution exist. Some of them include explicit density estimation, where the model attempts to directly estimate $p(x)$ using some parametric models $p(x; \theta)$ Montúfar [2018]Van Den Oord et al. [2016], while other methods directly generate samples without needing an explicit formulation of the probability distribution, such as Generative Adversarial Networks (GANs) Goodfellow et al. [2014]. However, the task of estimating an unknown distribution is challenging, especially for complex distributions or in the high-dimensional case.

A common issue is the presence of unknown normalization constants. Many generative models require computing a probability distribution that includes a normalization term to ensure that the probabilities sum to 1: the distribution p may not be normalized. This term, denoted as Z, is the integral over the entire space:

$$Z = \int p(x)dx.$$

In high-dimensional spaces, this computation becomes complicated, if not impossible, making it difficult to normalize the distribution correctly.

Several approaches have been proposed to circumvent the issue of normalization constants. For example, score-based generative models do not explicitly model the probability distribution $p(x)$, but instead focus on learning the score function $\nabla_x \log p(x)$, which is the gradient of the log probability density with respect to the data x, and does not depend on Z.

2 Backgrounds

2.1 Generalities on diffusion models

The papers we worked on explore diffusion models, which are inspired by diffusion processes Weilbach et al. [2023]. The core idea is to transform an initial sample x_0 into a chain $x_{0:T}$ by progressively adding noise at each stage, governed by a transition distribution:

$$q(x_t \mid x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I)$$

where $\beta_{1:T}$ determines the noise level at each step. Then, given some noise x_T, the model can estimate x_0.

Diffusion models leverage this framework by employing a neural network to approximate the reverse process. Specifically, the network predicts the reverse transition distribution:

$$p_\theta(x_{t-1}|x_t) \approx q(x_{t-1}|x_t).$$

These models rely on score matching techniques to enable tractable solutions for training Song et al. [2020a] Song and Ermon [2019]. However, alternative approaches, such as Annealed Importance Sampling Sohl-Dickstein et al. [2015], have also been explored in the literature to approximate the reverse Markov transitions effectively.

2.2 The advantages of score matching

As explained in the introduction, generative models aim to learn a probability distribution from a given dataset, so that it is possible to sample from the learned distribution afterwards.

A possibility is to use deep generative models that leverage deep neural networks to estimate the distribution $p_\theta(x)$, where θ are the parameters of the model.

However, as the outputs of neural networks, denoted as f_θ, do not inherently represent probabilities, one strategy is to transform these outputs into positive values by applying the exponential function e^{f_θ} to have a positive output. Despite this transformation, normalizing the output to yield a valid probability distribution requires the computation of the integral $\displaystyle\int e^{f_\theta(x)}\,\mathrm{d}x$, which is often untractable in practice.

Several studies have proposed methods for approximating the normalizing constant $\displaystyle\int e^{f_\theta(x)}\,\mathrm{d}x$ LeCun et al. [2006]Fabius and Van Amersfoort [2014] Rezende and Mohamed [2015]. Finally, Generative Adversarial Networks (GANs) give another strategy, where two models are trained : a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G Goodfellow et al. [2014]. However, these models can not evaluate explicit probabilities.

An alternative approach that combines flexibility in model architecture with accurate probability estimation involves utilizing the (Stein) score function $\nabla_x \log p(x)$, derived from a given probability $p(x)$. Here, the neural network model estimates $\nabla_x \log p(x)$, free from the constraints associated with direct probability estimation. This approach also provides a more robust methodology for generating samples from complex data distributions.

Remark : Intuitively, a score function gives the direction where the density function grows most quickly.

2.3 Score Matching

Given a dataset $D_n = \{x^{(1)}, x^{(2)}, .., x^{(n)}\}$, the objective function $J(\theta)$ which must be minimized with respect to the parameters θ to estimate the (Stein) score function can be expressed as follows :

$$J(\theta) = \frac{1}{2}\mathbb{E}_{p_{data}}\left[\|\psi(x,\theta) - \nabla_x \log p_{data}(x)\|_2^2\right], \quad (1)$$

where p_{data} corresponds to the true unknown probability distribution, and $\psi(x,\theta) = \nabla_x \log q(x,\theta)$, with q being the density model with parameters θ. As p_{data} is unknown, $J(\theta)$ can not be minimized yet.

2.4 Sampling with Langevin dynamics

Sampling from a probability distribution p knowing only its score $\nabla_x \log p(x)$ can be carried out using Langevin dynamics. Given a step size $\varepsilon > 0$, and an initial vector $\tilde{x}_0$, the Euler-Maruyama discretization of Langevin dynamics reads:

$$\tilde{x}_{t+1} = \tilde{x}_t + \frac{\varepsilon}{2}\nabla_x \log p(\tilde{x}_t) + \sqrt{\varepsilon}z_t, \quad (2)$$

where $z_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I)$. Theoretically, $\tilde{x}_T$ has density p when $T \to +\infty$ and $\varepsilon \to 0$. In practice, when $T < \infty$ and $\varepsilon > 0$, a Metropolis-Hastings correction is required Chen et al. [2014]. However, even without this correction, convergence bounds can be established Dalalyan [2017].

The above dynamics is the discrete version of the continuous Langevin dynamics :

$$\dot{X}_t = \frac{1}{2} \nabla_x \log p(X_t) + W_t,$$

with W a Brownian motion. This continuous approach is studied in Song et al. [2020a].

3 Denoising Score Matching

3.1 Limitations of Score Matching

Score Matching has its drawbacks, including:

- The need to compute $\nabla_x \log p_{data}(x)$ in (1).
  In fact, it is shown in Vincent [2011] that minimizing $J(\theta)$ in (1) is equivalent, up to some constants and under some assumptions, to minimizing
  $$\mathbb{E}_{p_{data}} \left[\text{Tr}(\nabla_x \psi(x, \theta)) + \frac{1}{2} \left\| \psi(x, \theta) \right\|^2 \right].$$
  However, computing $\text{Tr}(\nabla_x \psi(x, \theta))$ can be costly in high dimension (as it is the case with images). As a solution, Song et al. [2020b] introduced Sliced Score Matching, which consists in minimizing
  $$\mathbb{E}_{p_v} \mathbb{E}_{p_{data}} \left[v^T \nabla_x \psi(x, \theta) v + \frac{1}{2} \left\| \psi(x, \theta) \right\|^2 \right]$$
  for some distribution p_v such as the multivariate standard normal for example. It is shown in the paper that the computation of this quantity is scalable and can be optimized more efficiently than the previous one.
- The manifold hypothesis. In real-world datasets, data usually concentrates on low-dimensional manifolds and is not defined over the entire space. If $p_{data}(x)$ is non-zero only on such a manifold, then $\nabla_x \log p_{data}(x)$ will not be properly defined, making it difficult to use score-based methods. Moreover, the theoretical conditions for Score Matching in low-density regions (where p_{data} is non-zero) require the support of the data distribution to be the whole space. Without it, Score Matching yields rather poor results because only a few samples are present in these regions, as can be seen in figure 4 in the appendix.

These issues (and other ones as well) are detailed in Song and Ermon [2019].

3.2 A solution: Denoising Score Matching

Denoising Score Matching Song and Ermon [2019] is a method that tackles the issue of the manifold hypothesis.

Instead of approximating $\nabla_x \log p_{data}(x)$ by $\psi(x, \theta)$ as in (1), the goal is to find $\psi(x, \theta)$ which approximates the score of a perturbed version of p_{data}. More precisely, given $\sigma > 0$, define $q_\sigma(\tilde{x} \mid x) = \frac{1}{(2\pi\sigma^2)^{d/2}} e^{-\frac{1}{2\sigma^2} \|\tilde{x} - x\|^2}$, and $q_\sigma(\tilde{x}) := \displaystyle\int q_\sigma(\tilde{x} \mid x) p_{data}(x)\, dx$: q_σ is simply the density p_{data} perturbed by some Gaussian noise.

Then, Denoising Score Matching consists in minimizing the quantity

$$J_{\text{ESM}_{q_\sigma}}(\theta) = \mathbb{E}_{q_\sigma(\tilde{x})} \left[\frac{1}{2} \left\| \psi(\tilde{x}, \theta) - \nabla_x \log q_\sigma(\tilde{x}) \right\|_2^2 \right].$$

It is shown in Vincent [2011] that minimizing $J_{\text{ESM}_{q_\sigma}}(\theta)$ is equivalent to minimizing

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

or, in the case of Gaussian noise,

$$J_{\text{DSM}_{q_\sigma}}(\theta) = \mathbb{E}_{q_\sigma(x, \tilde{x})} \left[\frac{1}{2} \left\| \psi(\tilde{x}, \theta) - \frac{1}{\sigma^2}(x - \tilde{x}) \right\|_2^2 \right]. \tag{3}$$

This new objective is easier to minimize, because adding a Gaussian noise (supported in the entire space) solves the issue raised by the manifold hypothesis.

Also note that the empirical version of (3) can easily be computed and optimized.

In Appendix A.2, we give details on the link between diffusion models we mentionned in 2.1 and denoising score matching.

3.3 Annealed Langevin dynamics

Annealed Langevin dynamics can be employed to adjust the parameter σ. There are two key considerations when selecting the optimal value for σ:

- Large noise facilitates the optimization process: given the need to validate the manifold hypothesis, a larger value of σ reduces the occurrence of low-density regions, ensuring that the score estimation remains more accurate.
- Excessive noise may deviate from the target distribution: if the noise is too large, the model may diverge from the desired target distribution. As the noise decreases, the model is more likely to converge toward the true distribution. Using the notation from section 3.2, $q_\sigma(\widetilde{x}|x) \to 1 \quad \text{as} \quad \sigma \to 0$, which implies that $q_\sigma(x, \tilde{x}) \to p_{data}$.

Annealed Langevin dynamics, described in the algorithm on the right, starts with higher noise during the early stages of sampling, gradually decreasing over time.
This approach allows the model to explore the parameter space initially, and then focus more precisely on the true distribution as the noise is reduced.

Algorithm 1 Annealed Langevin Dynamics

```
Input $\{\sigma_i\}_{i=1}^{L}, \epsilon, T, \tilde{x}_0$
for $i \leftarrow 1$ to $L$ do
    $\alpha_i \leftarrow \epsilon \frac{\sigma_i^2}{\sigma_L^2}$
    for $t \leftarrow 1$ to $T$ do
        Draw $z_t \sim \mathcal{N}(0, I)$
        $\tilde{x}_t \leftarrow \tilde{x}_{t-1} + \frac{\alpha_i}{2} \nabla s_\theta(\tilde{x}_{t-1}, \sigma_i) + \sqrt{\alpha_i} z_t$
    end for
    $\tilde{x}_0 \leftarrow \tilde{x}_T$
end for
return $\tilde{x}_T$
```

In the experimental section, we explore alternative sigma sequences that adhere to the criterion of being large initially and decreasing over time. Despite satisfying this condition, these configurations do not always generate high-quality images, thereby demonstrating that this condition is insufficient for ensuring proper model behavior.

4 Experiments

4.1 Model

As in Song and Ermon [2019], the model we used is a Noise Conditional Score Network (NCSN). This model takes as input both a vector x and a noise level σ, and aims at predicting the score of the perturbed distribution: $s_\theta(x; \theta) \approx \nabla_x \log q_\sigma(x)$. Note that the noise levels have to be fixed at the beginning: the number of such levels will determine the number of parameters of the model.

This model is based on the U-Net architecture (see Ronneberger et al. [2015] for reference). In our implementation, we include both the original U-Net and the Noise Conditional Score Network (NCSN). U-Net offers the advantage of having fewer parameters than NCSN, as its parameter count does not depend on the number of noise levels σ_i. However, the results obtained with U-Net are not as strong as those achieved with NCSN.

Then, we adopt denoising score matching as it is faster than sliced score matching and aligns better with estimating scores of noise-perturbed data distributions. With the notations of section 3.2, we have the denoising score matching objective for a single noise level σ:

$$\ell(\theta; \sigma) = \frac{1}{2} \mathbb{E}_{p_{\text{data}}(\mathbf{x})} \mathbb{E}_{\mathcal{N}(\tilde{\mathbf{x}}|\mathbf{x}, \sigma^2 \mathbf{I})} \left\| s_\theta(\tilde{\mathbf{x}}, \sigma) + \frac{\tilde{\mathbf{x}} - \mathbf{x}}{\sigma^2} \right\|_2^2.$$

We combine the above for all $\sigma \in \{\sigma_i\}_{i=1}^{L}$ into a unified objective:

$$\mathcal{L}(\theta; \{\sigma_i\}_{i=1}^{L}) = \frac{1}{L} \sum_{i=1}^{L} \lambda(\sigma_i) \ell(\theta; \sigma_i),$$

where $\lambda(\sigma_i) > 0$ is a coefficient function and L the number of noise levels. To ensure $\lambda(\sigma)\ell(\theta;\sigma)$ has a consistent order of magnitude, the authors of Song and Ermon [2019] first observed empirically that $\|s_\theta(\tilde{\mathbf{x}},\sigma)\|_2 \propto 1/\sigma$. Therefore using $\lambda(\sigma) = \sigma^2$ approximately makes $\lambda(\sigma)\ell(\theta;\sigma)$ independent of σ.

4.2 Choice of σ

In the Annealed Langevin dynamics, the noise σ is progressively reduced. A question is to know how to reduce it: in Song and Ermon [2019], the authors use a geometric sequence. In their subsequent work Song and Ermon [2020], they provide a heuristic explanation supporting why a geometric sequence is a reasonable choice, which we explain in A.4.

We decided to try and experiment on this choice, by testing other possibilities on the CIFAR-10 dataset. The results were compared using Fréchet Inception Distance (FID), as explained in A.3 in the appendix.

We only considered sequences starting from 1 and ending with 0.01. Specifically:

- As in Song and Ermon [2019], a geometric sequence $\sigma_i = q^i$ with q such that $q^{200} = 0.01$.
- An affine sequence, $\sigma_i = ai + b$, with a and b such that $\sigma_1 = 1$ and $\sigma_{200} = 0.01$.
- A cosine sequence, $\sigma_i = \cos(ai + b)$, with a and b such that $\sigma_1 = 1$ and $\sigma_{200} = 0.01$.
- A sigmoidal sequence, $\sigma_i = \dfrac{m}{1 + \exp(-(ai + b))} + p$, with a, b, m, p chosen so that $\sigma_1 = 1, \sigma_{200} = 0.01$, and to prevent the decrease from being too brutal.

The described sequences can be seen in Figure 5, and the results on CIFAR-10 in Figure 1.

σ sequence	**Cosine**	**Geometric**	**Linear**	**Sigmoid**
FID	108.77	16.88	80.34	16.71

Figure 1: FID for different σ sequences on CIFAR-10

We see that a geometric sequence and a sigmoid sequence lead to similar results, while a linear sequence and a cosine sequence lead to rather poor results.

Samples for these different sequences can be seen in 6. We observe that for the linear and the cosine sequences, the images are noisy, whereas they are very clean for the geometric and sigmoid sequences.

4.3 On the importance of σ_1

We decided to analyze the impact of σ_1 on the quality of generated images. Specifically, we expect to get rather bad results when σ_1 is too large, because then the resulting final distribution may deviate significantly from the target, producing irrelevant samples. Conversely, setting σ_1 too small may prevent the model from exploring properly the distribution, potentially leading to suboptimal outcomes.

For this investigation, we conducted experiments on the MNIST dataset using a geometric sequence $(\sigma_i)_i$, with $\sigma_1 \in \{1, 5, 10, 20, 30\}$. For each value of σ_1, we generated 10,000 samples and computed the Fréchet Inception Distance (FID) between these samples and the MNIST dataset. The resulting FID scores are presented in Table 1.

σ_1	1	5	10	20	30	100
FID	15.60	3.38	3.67	3.92	4.15	5.17

Table 1: FID score for various values of σ_1 on the MNIST dataset.

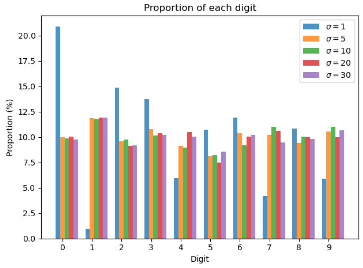


Figure 2: Class proportions in generated samples for different σ_1 values.

From the results, we observe that smaller values of σ_1 result in relatively high FID scores. However, a visual inspection of the generated images reveals that they closely resemble those in the MNIST dataset, despite the poor FID metrics.

To further investigate, we analyzed the class distribution of the generated samples. This was achieved using a classifier trained on MNIST, which achieves over 99% accuracy. The class proportions for different σ_1 values are shown in Figure 2.

The results indicate that for $\sigma_1 = 1$, the class proportions vary significantly, ranging from 0.93% to 20.92%, whereas for the other values of σ_1, the proportions vary between 7% and 12%. This imbalance suggests that overly small σ_1 values may lead to biases in the sampled data, even if the individual samples appear visually accurate.

Therefore σ_1 plays a role in the exploration and exploitation trade-off during sampling. For small values of σ_1, the model exploits specific regions of the distribution too early, resulting in biases in the generated data, as seen in the imbalanced class proportions for $\sigma_1 = 1$. Conversely, larger values of σ_1 promote greater exploration, which mitigates biases but can cause the generated samples to deviate from the target distribution, as was observed in practice. Thus, achieving an optimal σ_1 is essential to ensure a proper balance between exploration and exploitation, leading to the generation of diverse yet accurate samples.

4.4 Unbalanced data

In this experiment, we investigated the impact of unbalanced class distributions in the training dataset on the model's ability to generate samples. Specifically, we examined how reducing the number of samples for a single class corresponding to the digit 1 in the MNIST dataset might influence the model's performance: would the model generate fewer samples "1", and would the quality of these samples be reduced ?

Class	0	1	2	3	4	5	6	7	8	9
Proportion (%)	11.02	4.36	10.89	10.48	9.98	8.96	10.54	10.77	11.13	11.87

Figure 3: Proportions of each class in the generated samples when using an imbalanced training dataset.

As expected, the proportion of samples for the digit 1 in the generated data is significantly lower compared to the other classes. This behavior aligns with the fact that reducing the number of "1" images in the training dataset effectively changes the target density. Our observations are consistent with prior findings that annealed Langevin dynamics can correctly estimate the weights of different components in the target density.

Then, we computed the FID between true "1" images from the MNIST dataset, and generated "1" images.

When using only 35% of the "1" images to train the model, the FID is 14.30; when using all "1" images to train the model, the FID is 10.24. So using less samples leads to a higher FID for the corresponding class. This was of course expected: since the model is trained on less images, it cannot perform as well as when all images are used.

5 Conclusion

Denoising score matching provides an effective method for reducing the challenges associated with approximating complex data distributions. As previously discussed, the non-zero values of the data distribution are typically concentrated within a small manifold, rendering the score function ill-defined across the entire space. Additionally, computing the score function in high-dimensional settings remains computationally expensive.

To address these issues, denoising score matching perturbs the initial data distribution by adding noise, thereby expanding the support of the distribution and making it well-defined. The objective then shifts from estimating the exact score function to approximating a noise-augmented version. Stochastic simulation methods, such as Annealed Langevin Dynamics, are particularly effective for generating new samples, as explained in the original paper.

In this research, we first examined the hyperparameters proposed in the original study and explored alternative approaches. In the experimental section, we discuss the impact of different values for the noise levels $(\sigma_i)_i$ and its initial value σ_1.

Our findings indicate that simply using a decreasing sequence of $(\sigma_i)_i$ values is insufficient to achieve optimal image generation. Furthermore, while the initial value of σ_1 does not disrupt the class proportions within the dataset, smaller initial values result in generated images that deviate significantly from the original dataset, MNIST in the experiment. Higher initial values (greater than 1) yield better results.

We also conducted experiments to evaluate how unbalanced datasets influence the class proportions of generated images. As anticipated, these experiments clearly demonstrate the impact of dataset imbalance on the generated samples.

Finally, we extended our experiments to the Oxford-IIIT Pet dataset to assess its impact on sample generation. The results in Fig. 9 highlight that the number of iterations required to produce clean, high-quality images is highly dataset-dependent.

A Appendix

The code we used can be found in the following GitHub repository:

https://github.com/lucas-versini/Denoising-score-matching-for-diffusion-models

A.1 Contribution statement

- Manon Gouttefangeas: conducted background research by reviewing relevant articles and summarizing key concepts. Ran several experiments on toy datasets in the notebooks to explore and demonstrate core methods. Improved parts of the codebase to enhance functionality. Wrote the majority of the introductory and related work sections of the report, wrote part of the *Denoising Score Matching* section and the conclusion.
- Mohammed Raki: conducted research on Sliced Score Matching and Langevin Dynamics sampling. Ran hyperparameter experiments on toy datasets for Langevin Dynamics. Wrote the *Model* subsection of *Experiments* and part of *Sampling with Langevin Dynamics* and *Limitations of Score Matching* subsections.
- Lucas Versini: wrote the initial versions of the notebooks for Score Matching and Denoising Score Matching. Trained models on MNIST and CIFAR-10 datasets. Investigated the impact of the choice of the noise sequence $(\sigma_i)_i$, the initial noise σ_1 and the number of images per class on model performance. Wrote part of the *Denoising Score Matching* section, and the majority of the *Experiments* section.

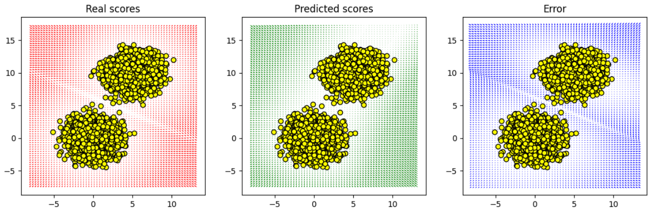


Figure 4: Poor estimation in low density regions.

A.2 Link between diffusion models and denoising score matching

In diffusion models Ho et al. [2020], we begin by considering a sample $x_0 \sim q$. The forward diffusion process is defined recursively as:

$$x_t = \sqrt{1-\beta_t}\, x_{t-1} + \sqrt{\beta_t}\, \varepsilon_t,$$

where $\beta_t \in (0,1)$ and $\varepsilon_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I)$. This can be equivalently expressed as a conditional Gaussian distribution:

$$x_t \mid x_{t-1} \sim \mathcal{N}(\sqrt{1-\beta_t}\, x_{t-1}, \beta_t I).$$

It can be shown that the marginal distribution of x_t given x_0 is $x_t \mid x_0 \sim \mathcal{N}(\sqrt{\overline{\alpha}_t}\, x_0, (1-\overline{\alpha}_t)I)$, where $\overline{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$. As $t \to \infty$, the distribution of x_t converges to a standard Gaussian $\mathcal{N}(0, I)$.

The core idea behind diffusion models is to reverse the forward diffusion process. Starting from $x_T \sim \mathcal{N}(0, I)$ for a sufficiently large T, we aim to reconstruct x_0 by estimating x_{t-1} from x_t using a learned model of the form:

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(x_{t-1} \mid \mu_\theta(x_t, t), \beta_t I),$$

where $\mu_\theta(x_t, t)$ is a learned mean function. The objective is to learn the parameter θ.

To maximize the likelihood $\log p_\theta(x_0)$, we derive a variational lower bound using the following inequalities:

$$\begin{aligned}\log p_\theta(x_0) &= \log \mathbb{E}_{(x_0,\ldots,x_T)\sim q}\left[\frac{p_\theta(x_0,\ldots,x_T)}{q(x_1,\ldots,x_T \mid x_0)}\right] \\ &\geq \mathbb{E}_{(x_0,\ldots,x_T)\sim q}\left[\log \frac{p_\theta(x_0,\ldots,x_T)}{q(x_1,\ldots,x_T \mid x_0)}\right] \quad \text{(Jensen's inequality)} \\ &=: \mathcal{L}(\theta).\end{aligned}$$

Thus, instead of maximizing $\log p_\theta(x_0)$, we maximize the lower bound $\mathcal{L}(\theta)$. By reparametrizing the model to predict noise $\varepsilon(x_t, t)$, where $x_t = \sqrt{\overline{\alpha}_t}\, x_0 + \sqrt{1-\overline{\alpha}_t}\, \varepsilon(x_t, t)$, maximizing $\mathcal{L}(\theta)$ becomes equivalent to minimizing the following denoising objective:

$$\sum_{t=1}^{T} \frac{\beta_t^2}{2\alpha_t \sigma_t^2 \zeta_t^2} \mathbb{E}_{x_0 \sim q, \varepsilon \sim \mathcal{N}(0,I)} \left[\|\varepsilon_\theta(x_t, t) - \varepsilon\|_2^2 \right]. \tag{4}$$

We can compare this objective to the Denoising Score Matching (DSM) objective:

$$\frac{1}{2L} \sum_{i=1}^{L} \lambda(\sigma_i) \mathbb{E}_{x_0, \varepsilon} \left[\|s_\theta(\tilde{x}, \sigma_i) - \nabla_{\tilde{x}} \log q_{\sigma_i}(\tilde{x} \mid x_0)\|_2^2 \right]. \tag{5}$$

To draw a direct parallel, we rewrite (4) by noting that $x_t \mid x_0 \sim \mathcal{N}(\sqrt{\overline{\alpha}_t}\, x_0, (1-\overline{\alpha}_t)I)$. The log-density is then $\log q(x_t \mid x_0) = -\frac{1}{2(1-\overline{\alpha}_t)} \|x_t - \sqrt{\overline{\alpha}_t}\, x_0\|_2^2 +$ constant. Thus, the score function is $\nabla_{x_t} \log q(x_t \mid x_0) = -\frac{1}{1-\overline{\alpha}_t}(x_t - \sqrt{\overline{\alpha}_t}\, x_0)$.

Since $\varepsilon_\theta(x_t, t)$ approximates the noise $\varepsilon(x_t, t)$, we have

$$\nabla_{x_t} \log q(x_t \mid x_0) = -\frac{\varepsilon_\theta(x_t, t)}{\sqrt{1-\overline{\alpha}_t}},$$

which shows that (4) and (5) have the same structure.

In the diffusion process, we have time steps, whereas in the denoising score matching, we have different noise levels. In the first case, T needs to be large enough so that the distribution of x_T is approximately $\mathcal{N}(0, I)$, while in the second case, L needs to be large enough as discussed previously in this report.

A.3 Fréchet Inception Distance

In order to compare several models, and evaluate the impact of a hyperparameter, we needed to find a way to quantify the quality of the generated images. To do so, we relied on the Fréchet Inception Distance (FID), see Heusel et al. [2017].

The idea is relatively simple: given two multivariate Gaussian distributions $\mathcal{N}(\mu_1, \Sigma_1)$ and $\mathcal{N}(\mu_2, \Sigma_2)$, a measure of the distance between these distributions is given by Fréchet distance (or equivalently in this case, Wasserstein-2 distance):

$$d\left(\mathcal{N}(\mu_1, \Sigma_1), \mathcal{N}(\mu_2, \Sigma_2)\right)^2 = \|\mu_1 - \mu_2\|_2^2 + \operatorname{tr}\left(\Sigma_1 + \Sigma_2 - 2\left(\Sigma_1 \Sigma_2\right)^{1/2}\right).$$

Then, given the real dataset S_1 and generated samples S_2, the idea is to consider f a model pretrained on ImageNet (in this case, a variant of the Inception model Szegedy et al. [2015]), to find $\mu_1, \Sigma_1, \mu_2, \Sigma_2$ such that $\mathcal{N}(\mu_1, \Sigma_1)$ fits $f(S_1)$ and $\mathcal{N}(\mu_2, \Sigma_2)$ fits $f(S_2)$, and to compute the distance between the two distributions.

The result is the FID. A small FID indicates that S_1 and S_2 are quite close, whereas a large FID indicates that the performance of our generative models is quite poor.

A.4 Geometric noise sequence

As we mentioned in 4.2, the authors of Song and Ermon [2020] explain why using a geometric sequence $(\sigma_i)_i$ is a natural choice.

Assume the dataset contains a single data point, so that $p_{\sigma_i}(x) = \mathcal{N}(x \mid 0, \sigma_i^2 I)$.

Then it is shown in Proposition 2 of Song and Ermon [2020] that whenever the dimension d is large (which is the case when dealing with images), we have $\left\|\mathcal{N}(0, \sigma_i^2 I)\right\|_2 \approx \mathcal{N}(m_i, s_i^2)$ with $m_i = \sqrt{d}\sigma_i$ and $s_i^2 = \sigma_i^2/2$, while the argument of $\mathcal{N}(0, \sigma_i^2 I)$ is uniformly distributed on $[0, 2\pi[$.

Then, what we want to do is choose σ_i such that samples from p_{σ_i} cover high density regions of $p_{\sigma_{i-1}}$. And we know that most of the mass of $p_{\sigma_{i-1}}$ is contained in $I_{i-1} = [m_i - 3s_i, m_i + 3s_i]$, and that

$$\begin{aligned}\int_{I_{i-1}} p_{\sigma_i}(r)\, dr &= \Phi\left(\frac{m_{i-1} - m_i + 3s_{i-1}}{s_i}\right) - \Phi\left(\frac{m_{i-1} - m_i - 3s_{i-1}}{s_i}\right) \\ &= \Phi\left(\sqrt{2d}\,(\gamma_i - 1) + 3\gamma_i\right) - \Phi\left(\sqrt{2d}\,(\gamma_i - 1) - 3\gamma_i\right),\end{aligned}$$

with $\gamma_i = \dfrac{\sigma_{i-1}}{\sigma_i}$, and Φ the cumulative distribution function of a $\mathcal{N}(0,1)$ random variable.

So if we require $\displaystyle\int_{I_{i-1}} p_{\sigma_i}(r)\,dr = C$ for each $i \in [\![1, L]\!]$ for some constant $C > 0$, then this can be achieved by requiring $\gamma_i = q$ to be constant, that is $\sigma_{i-1} = q\sigma_i$.

So overall, under simplified assumptions, choosing a geometric noise sequence implies that the distribution with noise σ_i overlaps with the distribution with noise σ_{i-1}.

A.5 Images obtained with different noise sequences

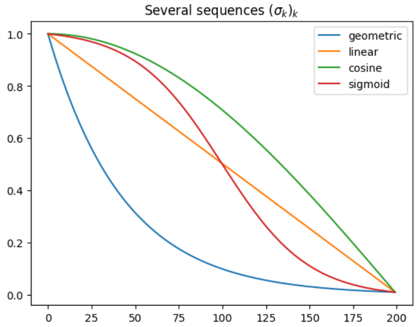
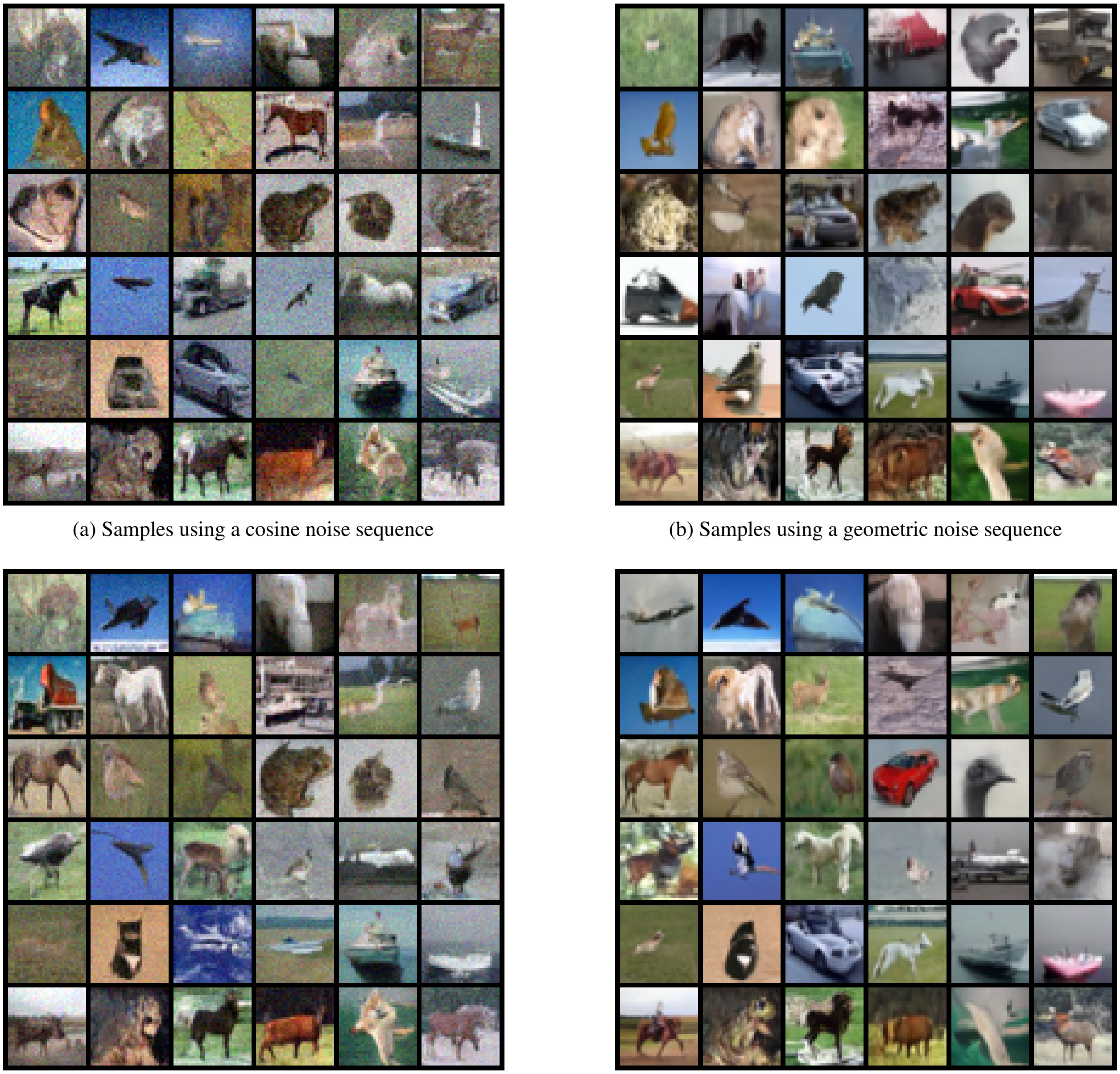


Figure 5: Different σ sequences



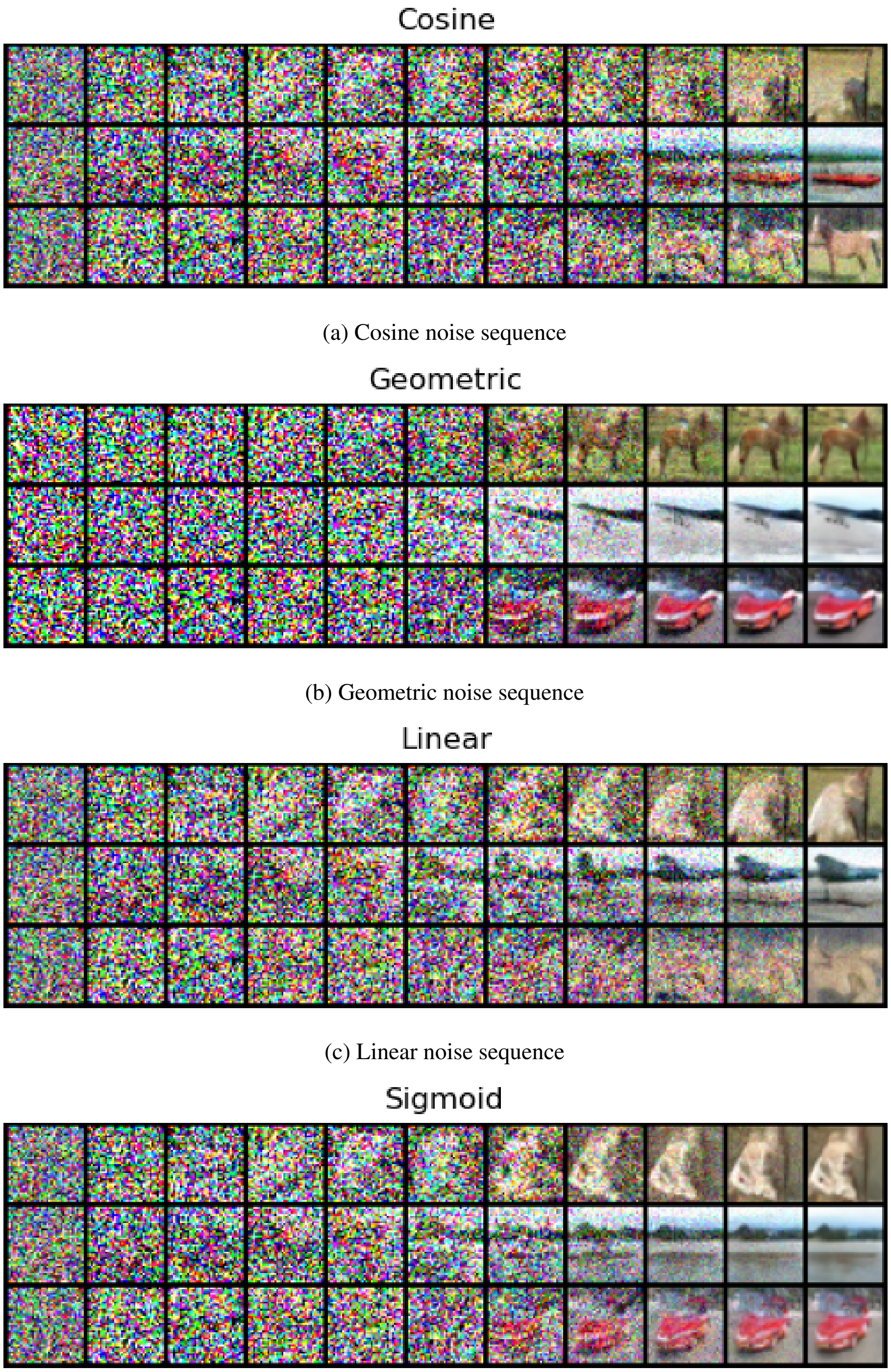
(a) Samples using a cosine noise sequence

(b) Samples using a geometric noise sequence

(c) Samples using a linear noise sequence

(d) Samples using a sigmoid noise sequence

Figure 6: CIFAR-10 samples generated using different noise sequences $(\sigma_i)_i$



(a) Cosine noise sequence

(b) Geometric noise sequence

(c) Linear noise sequence

(d) Sigmoid noise sequence

Figure 7: Evolution of Langevin dynamics for CIFAR-10 samples.

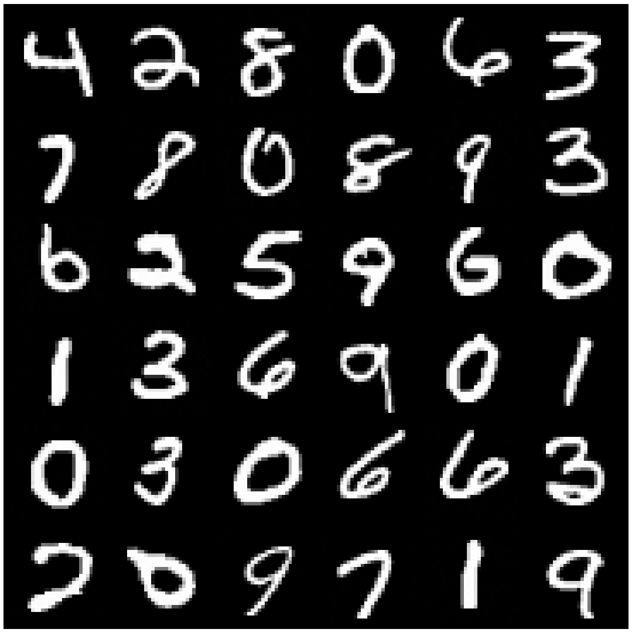


Figure 8: MNIST samples (geometric noise sequence)

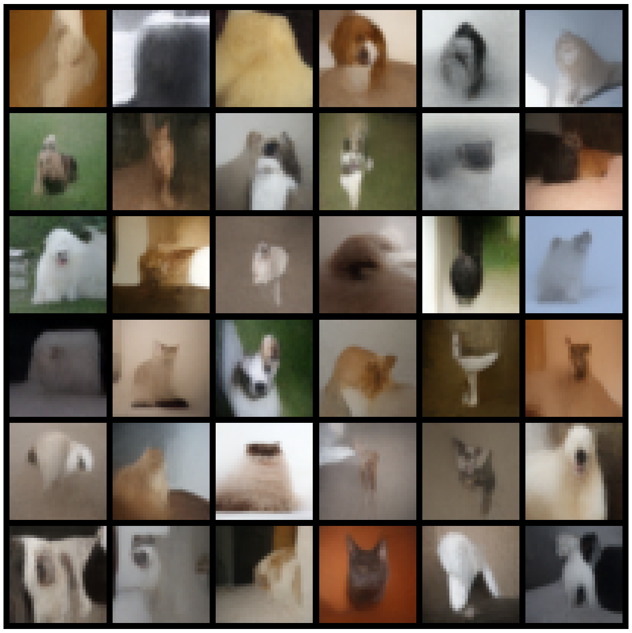


Figure 9: OxfordPet samples (geometric noise sequence). The results are less convincing than for the other datasets. This is likely due to the fact that the dataset contains more classes, with fewer samples per class.

References

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7): 1661–1674, 2011.

Guido Montúfar. Restricted boltzmann machines: Introduction and review. In *Information Geometry and Its Applications: On the Occasion of Shun-ichi Amari's 80th Birthday, IGAIA IV Liblice, Czech Republic, June 2016*, pages 75–115. Springer, 2018.

Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International conference on machine learning*, pages 1747–1756. PMLR, 2016.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Christian Dietrich Weilbach, William Harvey, and Frank Wood. Graphically structured diffusion models. In *International Conference on Machine Learning*, pages 36887–36909. PMLR, 2023.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020a.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, Fujie Huang, et al. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

Otto Fabius and Joost R Van Amersfoort. Variational recurrent auto-encoders. *arXiv preprint arXiv:1412.6581*, 2014.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.

Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo. In *International conference on machine learning*, pages 1683–1691. PMLR, 2014.

Arnak S Dalalyan. Theoretical guarantees for approximate sampling from smooth and log-concave densities. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 79(3):651–676, 2017.

Yang Song, Sahaj Garg, Jiaxin Shi, and Stefano Ermon. Sliced score matching: A scalable approach to density and score estimation. In *Uncertainty in Artificial Intelligence*, pages 574–584. PMLR, 2020b.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33:12438–12448, 2020.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.