Seminar

How imbalanced datasets affect score based generative models from an uncertainty perspective

Department of Statistics
Ludwig-Maximilians-Universität München

Carlo Rondo-Brovetto

Munich, 03 14th, 2025

Supervised by Dr. Mina Rezai

Contents

1 Introduction

This project focuses on exploring the theoretical bounds of score-based generative models (SGMs), particularly through the Wasserstein Uncertainty Propagation (WUP) theorem, as introduced in the referenced paper.
The main goal of this project is to empirically derive the from the paper introduced bound and investigate how dataset imbalance affects these bounds. Specifically, we aim to:

- Implement the theoretical bound for SGMs trained with the denoising score matching (DSM) objective.
- Study how different sources of error influences the bound especially focusing on how dataset imbalancing

To carry out this analysis, we use the MNIST dataset, a widely used benchmark dataset consisting of 70,000 grayscale images of handwritten digits (0-9). We assume that the true data distribution of MNIST is approximated using Kernel Density Estimation (KDE), calculated over the entire dataset.
This report summarizes the underlying theoretical framework derived from the paper [2], describes the model architecture, and explains the methodology used for evaluating the bound under different dataset imbalance conditions.

2 Theoretical Background

This study is based on the WUP theorem, which provides a model-form uncertainty quantification (UQ) bound that explains the robustness of SGMs under various sources of error. Specifically, it describes how errors in learning the score function propagate to a Wasserstein-1 (d_1) ball around the true data distribution, which is governed by the Fokker-Planck equation.

2.1 The Bound for Score-Based Generative Models

SGMs rely on learning a score function $s_\theta(x,t)$, which approximates the gradient of the log density of the perturbed data distribution:

$$s(x,t) = \nabla_x \log p_t(x).$$

The paper establishes a generalization bound under different sources of errors, such as:

- Finite sample approximation error: The error due to training the score model on a finite dataset.
- Early stopping error: The deviation from the optimal solution due to stopping training early.
- Choice of score-matching objective: We use denoising score matching (DSM) instead of an exact score-matching objective.
- Expressiveness of the neural network: The ability of the model to approximate the true score function.

The key result states that the error in the denoising score-matching (DSM) objective translates to an error bound in terms of the Wasserstein-1 (d_1) metric:

$$d_1(\pi, m_g(T)) \leq CR^{3/2}(1 + \|\nabla s_\theta\|_\infty) \left(R^2 e^{-\omega T/R^2} d_1(\pi, p_{\text{ref}}) + \sqrt{e'_{\text{nn}}} \right).$$

where

$$e'_{\text{nn}} \leq e_{\text{nn}} + \left(1 + \frac{|\log(\delta)|}{\epsilon} + T\|s_\theta\|^2_{C^2([0,T]\times\Omega)} \right) d_1(\pi_N, \pi).$$

Here:

- $d_1(\pi, m_g(T))$ is the Wasserstein-1 distance between the true distribution π and the generated distribution $m_g(T)$.
- e_{nn} is the score function approximation error, which we approximate using the DSM loss.
- $d_1(\pi, p_{\text{ref}})$ is the Wasserstein-1 distance between the true distribution and a chosen reference distribution.

- R and ω are constants, which depend on the model architecture.
- $\|\nabla s_\theta\|_\infty$ represents the Lipschitz constant of the estimated score function, which we estimate empirically.
- e_{nn}: The empirical denoising score matching (DSM) loss, which directly measures the error in learning the score function.
- δ: A small parameter ensuring that the bound holds for a finite dataset.
- ϵ: The noise variance level used in training.
- $T\|s_\theta\|^2_{C^2([0,T]\times\Omega)}$: A smoothness constraint on the score function, penalizing large variations.
- $d_1(\pi_N, \pi)$: The finite sample error, representing the difference between the dataset distribution π_N (empirical) and the true data distribution π (KDE-estimated MNIST).

2.2 Calculation of Terms in the Bound

To compute the bound, we estimated each term using the following methodology:

- **Approximating the True Distribution π using KDE:** We assume that the true data distribution of MNIST can be approximated using Kernel Density Estimation (KDE). We fit a Gaussian KDE model to the full MNIST dataset to obtain a density estimate.
- **Computing $d_1(\pi, p_{\text{ref}})$:** The reference distribution p_{ref} is chosen as a Gaussian prior. We compute the Wasserstein-1 distance between the KDE-based MNIST distribution and this Gaussian prior.
- **Computing the Score Function Approximation Error e_{nn}:** Using the loss function in training, which directly measures the difference between the learned and true score function.
- **Computing the Lipschitz Constant $\|\nabla s_\theta\|_\infty$:** We approximate the Lipschitz constant by computing the maximum gradient norm of the score function across a batch of test images.
- **Computing the Finite Sample Error $d_1(\pi_N, \pi)$:** This term accounts for the error introduced by training on a finite dataset. Given our assumption that the KDE-estimated MNIST distribution approximates π, we compare the empirical dataset to the KDE reference.
- **Computing the smoothness of the learned score function $\|s_\theta\|^2_{C^2}$:** Estimated from the model by computing higher-order gradients of the score function.

This empirical approach allows us to approximate the theoretical bound and evaluate how dataset imbalance affects model performance.

3 Model Architecture

To estimate the score function, we implemented a U-Net-inspired deep neural network, which processes image data and learns the gradient of the log-likelihood at different noise levels.

3.1 ScoreNet: The Time-Dependent Score-Based Model

Our score-based network (ScoreNet) is structured as follows:

- **Time Embedding Layer:** Encodes diffusion time t using Gaussian Fourier Features.
- **Encoding Path (Downsampling):** Sequential convolutional layers reduce the spatial dimension, increasing the number of feature maps.
- **Bottleneck Layer:** A deep representation of the image, which integrates information from different resolutions.
- **Decoding Path (Upsampling):** Transposed convolutional layers progressively restore the image size.
- **Final Output Layer:** Predicts the score function $s_\theta(x, t)$.

Mathematically, the network is parameterized as:

$$h_i = \text{Conv2D}(h_{i-1}) + \text{Dense}(\text{TimeEmbedding}(t)),$$

where h_i represents intermediate activations, and the time embedding is added at each layer.

3.2 Predictor-Corrector (PC) Sampler

To generate samples, we use a Predictor-Corrector (PC) sampler, which improves sample quality by combining:

- **Predictor Step:** Numerically solves the reverse-time SDE to generate new samples.
- **Corrector Step:** Applies Langevin MCMC using the score function to refine the samples.

Given an initial noise sample x_0, the predictor step follows:

$$x_{t-\Delta t} = x_t + f(x_t)\Delta t + \sigma(x_t)\sqrt{\Delta t}z,$$

where $f(x_t)$ is the learned drift term and $z \sim N(0, I)$. The corrector step further adjusts x_t using score-based gradient updates.

4 Results

The results presented in Table below illustrate the effect of dataset imbalance on the empirical bound of the Wasserstein-1 distance.

Keep Ratio	X1	X2	X3	X4	Empirical Bound
1.0	0.6673	15.9335	0.0329	0.0002	5.0646
0.9	0.6673	16.0009	0.0592	0.0005	5.2596
0.7	0.6673	16.2069	0.0602	0.0016	5.3871
0.4	0.6673	16.2186	0.0601	0.0038	5.4057
0.1	0.6673	16.3476	0.2331	0.0075	6.4915

Table 1: Computed empirical bounds on the Wasserstein-1 distance for different dataset imbalance levels.

The results show that dataset imbalance has a measurable impact on the theoretical bound and the performance of the generative model. While the Wasserstein-1 distance to the Gaussian prior (X_1) obviously remains constant across different imbalance levels, other terms in the bound exhibit notable variations. The key observations are:

- **DSM Loss (X_2) Increases:** As the dataset becomes more imbalanced, the DSM loss rises, indicating that the score function struggles to accurately estimate gradients of the data distribution. This suggests that training becomes less effective with increasing imbalance.
- **Lipschitz Constant (X_3) Rises Sharply:** While relatively stable for moderate imbalance, the Lipschitz constant increases significantly at extreme imbalance levels (e.g., 0.1 keep ratio). This indicates that the learned score function becomes less smooth, potentially leading to unstable sampling behavior.
- **Finite Sample Error (X_4) Grows:** The finite sample error increases as the dataset becomes more imbalanced, implying that the empirical distribution deviates further from the true MNIST distribution estimated via KDE. This contributes to a looser bound on the Wasserstein distance.
- **Overall Bound (Last Column) Expands:** The theoretical bound on the Wasserstein-1 distance to the true data distribution increases as the dataset becomes more imbalanced. This suggests that extreme class imbalance weakens the model's ability to generate realistic samples.

5 Conclusion

These findings highlight that maintaining sufficient class diversity in training datasets for score-based generative models is beneficiary for the models performance. While mild imbalance does not severely impact performance, extreme imbalance degrades the score function and generative quality.

References

[1] Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., Poole, B. (2021). *Score-Based Generative Modeling through Stochastic Differential Equations.*

[2] Mimikos-Stamatopoulos, N., Zhang, B. J., Katsoulakis, M. A. (2024). *Score-Based Generative Models are Provably Robust: An Uncertainty Quantification Perspective.*