Research Project (PRe)

Speciality : Applied Mathematics
Academic Year : 2024/2025

From Noise to Images : Implementing Score-Based Generative Models using Stochastic Differential Equation

NON-CONFIDENTIAL DOCUMENT

Abdoulaye TRAORE

Engineering Student - Promotion 2026

ENSTA Paris Supervisor :
Prof. Francesco Russo
University Professor

Host Organization Supervisor :
Prof. Elena Issoglio
Associate Professor in Probability

Internship Period : May 21, 2025 - July 27, 2025
Presented on August 21, 2025

Host Organization :
Università degli Studi di Torino
Department of Mathematics ”Giuseppe Peano”
Via Verdi, 8 - 10124 Torino, Italy

Non-Confidentiality Note

This document is **non-confidential**. It can therefore be made publicly accessible online by the ENSTA Paris community. The document will be archived in electronic format by the ENSTA Paris library.

Acknowledgments

I would like to express my profound gratitude to all the individuals and institutions who made this research internship both possible and enriching.

First and foremost, I wish to thank ENSTA Paris for the quality of its education and the numerous opportunities it offers to its students, enabling them to develop at the highest level.

I am deeply grateful to Professor Francesco Russo, my supervisor at ENSTA Paris, for his guidance, expertise, and continuous support throughout this research project. His insights into stochastic differential equations and mathematical modeling proved invaluable for this work that would not have been possible without his initiative and support.

My sincere appreciation goes to Professor Elena Issoglio from the University of Torino, who welcomed me into the Department of Mathematics ”Giuseppe Peano” and provided exceptional mentorship throughout this internship. I am especially thankful for the trust she placed in me, the intellectual freedom she allowed in exploring my ideas, and her thoughtful advice at each stage of the project. I also had the opportunity to attend her introductory course on Deep Learning, which provided solid foundations and valuable insights for my research.

I would also like to thank the entire Department of Mathematics ”Giuseppe Peano” at the University of Torino for the warm welcome and the stimulating research environment they provided during my stay in Italy, especially to Matteo Cagnotti for his help and support during this period.

I am also sincerely grateful to the Erasmus+ program and particularly to Mr Dramane DJARA and Mr Maicol VILLEGAS (International Mobility Officer of IP Paris and Outgoing International Mobility Officer and Erasmus+ Coordinator of ENSTA Paris respectively), which provided financial support through a mobility grant. This assistance made it possible for me to live comfortably in Italy and carry out this internship in optimal conditions.

Finally, I cannot conclude these acknowledgments without addressing a special thought to my family, who gives meaning to my life and provides me with unwavering support throughout my academic journey.

Abstract

This report presents the work accomplished during a research internship at the University of Torino, Department of Mathematics ”Giuseppe Peano,” focusing on score-based generative models using Stochastic Differential Equations (SDEs). Building upon Song et al. (ICLR 2021), this research explores Variance Exploding (VE), Variance Preserving (VP), and sub-Variance Preserving (sub-VP) SDEs for high-quality image generation.
A key contribution involves custom training an AFHQ-512 model from scratch and developing three adaptive classifier guidance strategies: amplified scaling, adaptive linear scaling, and sigma truncation techniques. Experiments were conducted on multiple high-resolution datasets (FFHQ-1024, CelebA-HQ-256, LSUN Church-256) using Lambda Cloud infrastructure.
The work demonstrates versatility through downstream applications including inpainting, colorization, and controllable generation using predictor-corrector sampling, providing valuable insights into scalability challenges and practical implementation of score-based generative models.

Keywords: Score-based models, Stochastic Differential Equations, Diffusion models, VE-SDE, Classifier guidance, Image generation, Deep learning

Résumé

Ce rapport présente le travail effectué lors d'un stage de recherche à l'Université de Turin, Département de Mathématiques ”Giuseppe Peano”, portant sur les modèles génératifs basés sur le score utilisant les Équations Différentielles Stochastiques (SDEs). S'appuyant sur Song et al. (ICLR 2021), cette recherche explore les SDEs Variance Explosive (VE), Variance Préservée (VP), et sous-Variance Préservée (sub-VP) pour la génération d'images haute qualité.
Une contribution clé implique l'entraînement personnalisé d'un modèle AFHQ-512 et le développement de trois stratégies de guidance adaptatives par classificateur. Les expériences ont été menées sur plusieurs jeux de données haute résolution (FFHQ-1024, CelebA-HQ-256, LSUN Church-256) utilisant l'infrastructure Lambda Cloud.
Le travail démontre la versatilité à travers des applications incluant l'inpainting, la colorisation, et la génération contrôlable, fournissant des insights sur les défis de passage à l'échelle et l'implémentation pratique des modèles génératifs basés sur le score.

Mots-clés : Modèles basés sur le score, Équations Différentielles Stochastiques, Modèles de diffusion, VE-SDE, Guidance par classificateur, Génération d'images, Apprentissage profond

Contents

Table of Illustrations and Appendices

List of Figures

List of Tables

List of Appendices

Structure of the Report

This report is organized as follows:

- Chapter 1 provides the introduction with context, motivation, research objectives, and internship framework;
- Chapter 2 introduces the theoretical foundations of score-based generative modeling, transitioning from classical approaches to the unified SDE-based framework, and contextualizes our approach within current state-of-the-art developments;
- Chapter 3 describes the model implementation, architectural choices, and training configurations across multiple high-resolution datasets;
- Chapter 4 presents experimental results and analysis for unconditional sample generation with visual and qualitative assessment;
- Chapter 5 details downstream tasks enabled by SBGM: image inpainting, colorization, and class-conditional generation with adaptive classifier guidance strategies;
- Chapter 6 focuses on limitations and failure modes including technical limitations, computational constraints, and methodological limitations;
- Chapter 7 concludes with critical reflections on achievements, limitations, lessons learned, and perspectives for future research;
- Appendices 8 provide comprehensive mathematical derivations, theoretical foundations for downstream tasks, and extended experimental results.
- Here is a link to the github repository, https://github.com/score-sde-project, that contains all materials and code run in this project.

1 Introduction

1.1 Context and Motivation

In recent years, deep generative models have become central tools in modern machine learning, enabling the synthesis of realistic images, audio, and text. Among them, diffusion models such as DALL-E, MidJourney or Stable diffusion have emerged as particularly powerful for generating high-quality samples in high-dimensional spaces, outperforming GANs in several benchmarks. These models simulate a reverse-time diffusion process that gradually denoises a random variable into a data sample, a process described mathematically using Stochastic Differential Equations (SDEs).

The connection between SDEs and deep learning offers a compelling continuous-time formulation of generative processes and opens the door to more principled analysis and new sampling methods. A key innovation in this domain is the use of **score-based models**, which aim to learn the gradient of the log-density of data (the "score function") at varying noise levels. These gradients are then used to guide the reverse-time evolution of a stochastic process from pure noise to a coherent data sample.

This approach was introduced and popularized by Song et al. (2021), whose work laid the foundation for a new family of generative models based on stochastic processes and score estimation. Their framework relies on different types of SDEs (VE, VP, sub-VP), various predictor-corrector sampling techniques, and a theoretical link to ODEs for likelihood computation.

The primary motivation of this internship was to study this emerging framework in depth, from both a theoretical and practical perspective, and to implement a full generative pipeline based on the Variance Exploding SDE (VE-SDE)—selected for its empirical robustness and ease of implementation.
As a secondary focus, we examined practical challenges such as those reported by Song et al. when scaling to high-resolution datasets : generated samples often lacked global coherence, symmetry, or semantic alignment. We explored how training strategies and guided sampling techniques, especially classifier guidance, could help address these issues in practice.

Image generated by DALL·E 3

Image generated by own AFHQ-512 model

Image generated by Stable Diffusion

Figure 1.1: Images generated by modern score-based generative models (text-to-image).

1.2 Research Objectives

The internship was carried out at the Department of Mathematics ”Giuseppe Peano” (University of Torino), under the supervision of Professor Elena Issoglio. The work was structured around the following research objectives:

- Investigate the mathematical foundations of score-based generative modeling using stochastic differential equations and their continuous-time formulations;
- Implement and train a VE-SDE-based model from scratch on high-resolution image datasets (FFHQ-1024, AFHQ-512, CelebA-HQ-256, and LSUN Church-256);
- Design and evaluate various sampling strategies using predictor-corrector (PC) samplers, including reverse-diffusion and fast adaptive methods;
- Develop adaptive classifier guidance strategies by integrating a ResNet-50 classifier for controllable generation;
- Apply the framework to multiple downstream tasks: unconditional sampling, inpainting, colorization, and class-conditional generation;
- Evaluate the quality of the generated samples using visual inspection and quantitative metrics (FID, Inception Score);
- Analyze observed limitations and propose directions for future improvements in high-resolution generation.

1.3 Internship Environment

This research internship took place in the Department of Mathematics “Giuseppe Peano” at the University of Torino, under the supervision of Professor Elena Issoglio, Associate Professor in probability. The department hosts several teams working on stochastic analysis, differential equations, and mathematical modeling, offering a stimulating and dynamic environment for mathematical research under the direction of Susanna Terracini, Head of the Department of Mathematics.
Throughout the internship, I worked on-site at the university’s library from Monday to Friday, from 9.00 a.m to 6.00 p.m, alongside a team of researchers specialized in stochastic calculus and probability theory. My tasks were carried out in autonomy while maintaining regular exchanges with my supervisor, who provided valuable feedback during weekly meetings.
In addition to research activities, I had access to extensive academic resources, including books from university’s library, online databases and access to computational resources. These materials were instrumental in deepening my understanding of both the theoretical and applied aspects of the project.
I also had the opportunity to attend a PhD-level course on Deep Learning delivered by Professor Elena which concluded with a lecture on score-based models by Antonio Ocello (Postdoctoral Researcher). Furthermore, I participated in a seminar on stochastic analysis and modeling organized by the probability group of the department. These academic events enriched my experience and supported the progression of my work.

2 Theoretical Background

2.1 Generative Modeling and Score Functions

Generative models aim to learn the underlying data distribution $p_{data}(x)$ and generate new samples that are statistically similar to the training data. Prior approaches include Variational Autoencoders (VAEs) which optimize a variational lower bound, Generative Adversarial Networks (GANs) using adversarial training, and Flow-based models with invertible transformations. While these methods achieved notable success, they face limitations: VAEs often produce blurry samples due to variational approximation, GANs suffer from training instability and mode collapse, and flow-based models require restrictive architectural constraints.

These limitations motivated the development of a fundamentally different approach introduced by Hyvärinen [6]: instead of directly modeling the data distribution $p_{data}(x)$, we can learn its **score function**, defined as the gradient of the log-density:

$$s(x) = \boldsymbol{\nabla_x} \mathbf{log} \boldsymbol{p}_{\mathbf{data}}(\boldsymbol{x}) \tag{2.1}$$

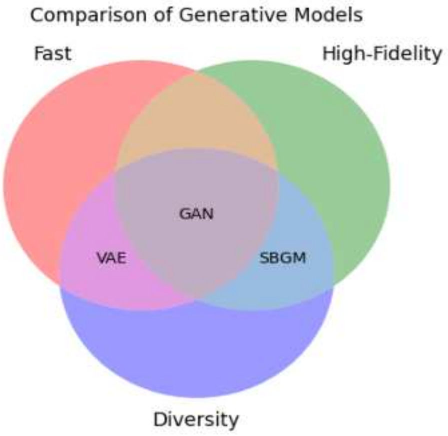


Figure 2.1: Comparison of Generative Models

This method is called Score-Based Generative Models (SBGM).
The figure 2.1 (based on Deep Learning Introduction course presented by Professor Elena Issoglio and Antonio Ocello, PostDoct Researcher at X) shows the relevant qualities of theses differents methods underlying the efficiency of SGBM for diversity ang high-fidelity sampling specially.

Motivation for Score-Based Modeling

To understand why learning the score function is advantageous, consider the fundamental challenge in density estimation.
Suppose we are given a dataset $x_1, x_2, ..., x_N$, where each point is drawn independently from an underlying data distribution $p_{data}(x)$.
Given this dataset, the goal of generative modeling is to fit a model to the data distribution such that we can synthesize new data points at will by sampling from the distribution.

We want to model this probability density $p_{data}(x)$ using an energy-based approach:

$$p_{data}(x) = \frac{1}{Z}\exp(-E_\theta(x)) \tag{2.2}$$

where $E_\theta(x)$ is an energy function parameterized by θ, a learnable parameter, and $Z > 0$ is the normalization constant (partition function):

$$Z = \int \exp(-E_\theta(x))dx$$

The critical problem is that computing Z requires integrating over the entire data space, which is intractable in high dimensions. This makes direct likelihood-based training impossible. However, when we take the logarithm and then the gradient of equation 2.2, something remarkable happens:

$$\boxed{\nabla_x \log p_{data}(x) = \nabla_x \log\left(\tfrac{1}{Z}\exp(-E_\theta(x))\right) = \nabla_x(\log(1/Z) - E_\theta(x)) = -\nabla_x E_\theta(x) \qquad (2.3)}$$

The normalization constant Z disappears!

This means we can learn the score function without ever computing the intractable partition function. The score function captures the local structure of the probability density—it points toward regions of higher probability while keeping all informations from it by a simple integration.

Challenges in Score Estimation

The key challenge is now to estimate the score functions by a parametrized function $s_\theta(x)$ such that $s_\theta(x) \approx \nabla_x \log p_{data}(x)$ with a Deep Neural Network (DNN). This is expected as score matching minimizes the Fisher divergence :

$$\mathbb{E}_{p(\mathbf{x})}\left[\|\nabla_{\mathbf{x}} \log p(\mathbf{x}) - s_\theta(\mathbf{x})\|_2^2\right] = \int p(\mathbf{x})\, \|\nabla_{\mathbf{x}} \log p(\mathbf{x}) - s_\theta(\mathbf{x})\|_2^2 \,\mathrm{d}\mathbf{x}.$$

Despite this theoretical elegance, a fundamental practical challenge emerges :

Score functions are difficult to estimate accurately in low-density regions where data is sparse. Neural networks trained on finite datasets struggle to learn meaningful gradients in regions where few or no training samples exist.

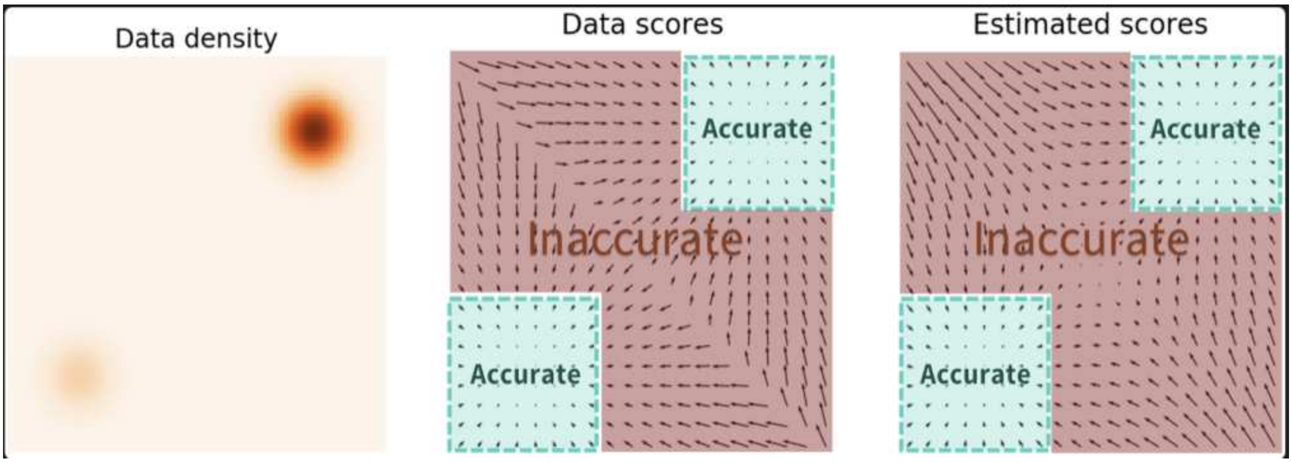


Figure 2.2: True score field with accurate gradients in high-density regions vs DNN-estimated score field. Source: [12]

This limitation poses a significant obstacle for sampling algorithms that rely on the score function, as they may get trapped in low-density regions with poorly estimated gradients, leading to poor sample quality or sampling failure.
The figure 2.2 shows poor estimation in low-density regions.

The key insight that addresses this challenge is conceptually elegant:

Instead of learning the score of the original data distribution $p_{data}(x)$, we can learn the score of **noise-perturbed versions** of the data. By injecting noise into all regions of the space, we artificially increase the density everywhere, enabling more reliable score estimation even in originally low-density regions. This approach, known as **denoising score matching**, forms the foundation of modern score-based generative models.

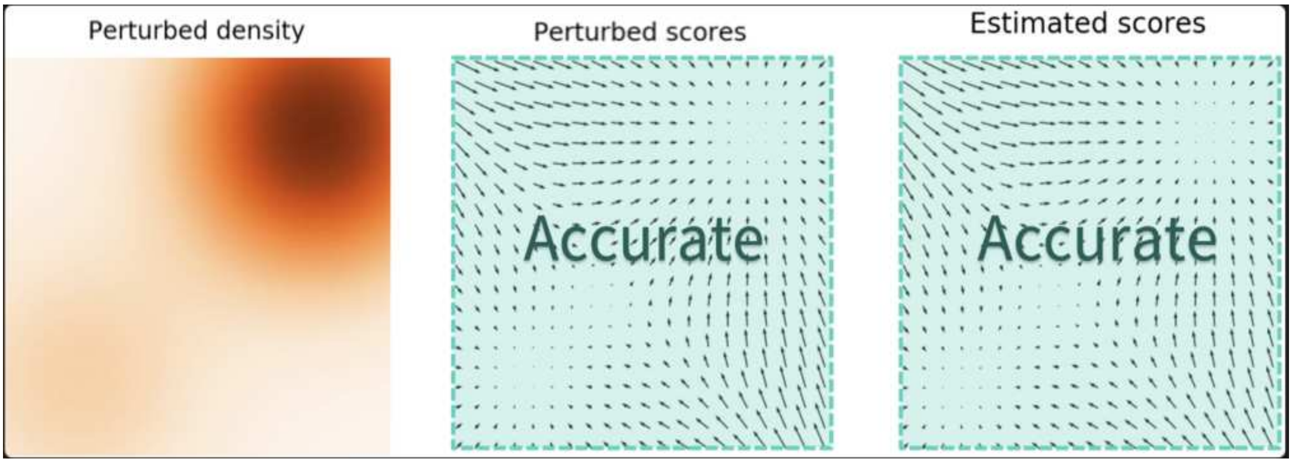


Figure 2.3: True perturbed-score field with accurate gradients in high-density regions vs DNN-estimated perturbed-score field. Source: [12]

For example, here, as shown by the figure 2.3, is what happens when we perturb a mixture of two Gaussians perturbed by additional Gaussian noise, showing satistying estimation in previous low-density regions.

How can we determine the optimal level of noise to apply during perturbation ?

Choosing the right noise level for the perturbation process is a trade-off. Using a high noise scale helps to reach low-density areas of the data space, improving score estimation in those regions. However, it also distorts the data significantly, moving it away from the true distribution. On the other hand, a smaller noise scale preserves the original data distribution more faithfully but fails to adequately cover low-density regions, which can hinder accurate score learning.

2.2 From Discrete Denoising to Continuous SDEs

The insight of using noise-perturbed data distributions led to two major discrete approaches: Score Matching with Langevin Dynamics (SMLD) and Denoising Diffusion Probabilistic Models (DDPM). These methods demonstrated remarkable success but operated with discrete noise schedules. The breakthrough came with recognizing that both approaches can be viewed as discretizations of continuous-time stochastic processes, leading to a unified SDE framework [14].

Score Matching with Langevin Dynamics (SMLD)

The SMLD approach, introduced by Song & Ermon (2019) [13], addresses the score estimation challenge by considering a sequence of noise scales.
Let $p_\sigma(\tilde{x}|x) := \mathcal{N}(\tilde{x}; x, \sigma^2 I)$ be a perturbation kernel, and

$$p_\sigma(\tilde{x}) := \int p_{data}(x) p_\sigma(\tilde{x}|x) dx, \quad \text{where} \quad p_{data}(x) \quad \text{denotes the data distribution.} \tag{2.4}$$

Consider a sequence of positive noise scales $0 < \sigma_{\min} = \sigma_1 < \sigma_2 < \cdots < \sigma_N = \sigma_{\max}$ where σ_1 is chosen small enough that $p_{\sigma_1}(x) \approx p_{data}(x)$, and σ_N is large enough that $p_{\sigma_N}(x) \approx \mathcal{N}(0, \sigma_N^2 I)$. Song & Ermon (2019) propose to train a Noise Conditional Score Network (NCSN), denoted by $s_\theta(x, \sigma)$, with a weighted sum of denoising score matching objectives:

$$\theta^* = \arg\min_\theta \sum_{i=1}^{N} \sigma_i^2 \mathbb{E}_{p_{data}(x)} \mathbb{E}_{p_{\sigma_i}(\tilde{x}|x)} \left[\| s_\theta(\tilde{x}, \sigma_i) - \nabla_{\tilde{x}} \log p_{\sigma_i}(\tilde{x}|x) \|_2^2 \right] \tag{2.5}$$

For Gaussian perturbations, the score function has a closed form:

$$\nabla_{\tilde{x}} \log p_{\sigma_i}(\tilde{x}|x) = -\frac{\tilde{x} - x}{\sigma_i^2}$$

A typical choice for the noise schedule is geometric progression:

$$\sigma_i = \sigma_{\min} \left(\frac{\sigma_{\max}}{\sigma_{\min}} \right)^{\frac{i-1}{N-1}}, \quad i = 1, 2, \ldots, N \tag{2.6}$$

For sampling, SMLD uses annealed Langevin dynamics. Starting from $x_N \sim \mathcal{N}(0, \sigma_N^2 I)$, the sampling process proceeds as :

$$x_i^{(m)} = x_i^{(m-1)} + \epsilon_i s_\theta^*(x_i^{(m-1)}, \sigma_i) + \sqrt{2\epsilon_i} z_i^{(m)}, \quad m = 1, 2, \ldots, M \tag{2.7}$$

where $z_i^{(m)} \sim \mathcal{N}(0, I)$, ϵ_i is a step size, and m indexes the Langevin steps at noise level i.

Denoising Diffusion Probabilistic Models (DDPM)

Parallel to SMLD, Sohl-Dickstein et al. (2015); Ho et al. (2020) [11, 5] developed DDPM, which considers a discrete Markov chain that gradually adds Gaussian noise to data. For each training data point $x_0 \sim p_{data}(x)$, a discrete Markov chain $\{x_0, x_1, \ldots, x_N\}$ is constructed such that:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1 - \beta_t} x_{t-1}, \beta_t I) \tag{2.8}$$

where $\{\beta_t\}_{t=1}^N$ is a variance schedule with $0 < \beta_1, \beta_2, \ldots, \beta_N < 1$.
Defining $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$, we can directly sample x_t from x_0:

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t} x_0, (1 - \bar{\alpha}_t) I) \tag{2.9}$$

Similar to SMLD, we can denote the perturbed data distribution as :

$$p_{\bar{\alpha}_t}(\tilde{x}) := \int p_{data}(x) p_{\bar{\alpha}_t}(\tilde{x}|x) dx. \tag{2.10}$$

A variational Markov chain in the reverse direction is parameterized with

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \frac{1}{\sqrt{1-\beta_t}}(x_t + \beta_t s_\theta(x_t,t)), \beta_t I) \tag{2.11}$$

and trained with a re-weighted variant of the evidence lower bound :

$$\theta^* = \arg\min_\theta \sum_{t=1}^{N} (1-\bar{\alpha}_t)\mathbb{E}_{p_{data}(x)}\mathbb{E}_{p_{\bar{\alpha}_t}(\tilde{x}|x)}\left[\|s_\theta(\tilde{x},t) - \nabla_{\tilde{x}} \log p_{\bar{\alpha}_t}(\tilde{x}|x)\|_2^2\right] \tag{2.12}$$

The relationship between noise prediction and score estimation is:

$$s_\theta(x_t,t) = -\frac{\epsilon_\theta(x_t,t)}{\sqrt{1-\bar{\alpha}_t}} \tag{2.13}$$

where ϵ_θ is a neural network that predicts the noise ϵ.
For sampling, DDPM uses ancestral sampling by starting from $x_N \sim \mathcal{N}(0,I)$ and following:

$$x_{t-1} = \frac{1}{\sqrt{1-\beta_t}}(x_t + \beta_t s_\theta^*(x_t,t)) + \sqrt{\beta_t} z_t, \quad t = N, N-1, \dots, 1 \tag{2.14}$$

where $z_t \sim \mathcal{N}(0,I)$.
For DDPM, a common variance schedule is linear:

$$\beta_t = \beta_{\min} + \frac{t-1}{T-1}(\beta_{\max} - \beta_{\min}) \tag{2.15}$$

Transition to Continuous Formulation

Rather than using discrete noise levels or timesteps, Song et al. (2021) proposed to consider a continuum of noise levels, leading to stochastic differential equations. This transition provides several advantages: unified framework, flexible sampling, and theoretical insights from stochastic calculus.

From SMLD to Variance Exploding SDE

When using a total of N noise scales, each perturbation kernel $p_{\sigma_i}(x|x_0)$ of SMLD corresponds to the distribution of x_i in the following Markov chain:

$$x_i = x_{i-1} + \sqrt{\sigma_i^2 - \sigma_{i-1}^2} z_{i-1}, \quad i = 1, \cdots, N \tag{2.16}$$

where $z_{i-1} \sim \mathcal{N}(0,I)$, and we have introduced $\sigma_0 = 0$ to simplify the notation.
In the limit of $N \to \infty$, $\{\sigma_i\}_{i=1}^N$ defined in 2.6 becomes a function $\sigma(t)$, z_i becomes a process $w(t)$, and the Markov chain $\{x_i\}_{i=1}^N$ becomes a continuous stochastic process $\{x(t)\}_{t=0}^1$, where we have used a continuous time variable $t \in [0,1]$ for indexing, rather than an integer i. The process $\{x(t)\}_{t=0}^1$ is given by the following SDE:

$$dx = \sqrt{\frac{d[\sigma^2(t)]}{dt}} dw \tag{2.17}$$

where $\sigma(t)$ is given by

$$\sigma(t) = \sigma_{\min}\left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)^t \tag{2.18}$$

and $w(t)$ is a standard Wiener process.

This is the **Variance Exploding (VE) SDE**, which corresponds exactly to the continuous limit of SMLD. (see complete derivations in Appendix A.1)

From DDPM to Variance Preserving SDE

Following a similar approach for DDPM, we consider the discrete Markov chain structure and examine its continuous limit. The DDPM discrete process can be written as:

$$x_i = \sqrt{1-\beta_i}x_{i-1} + \sqrt{\beta_i}z_{i-1}, \quad i = 1, \cdots, N \tag{2.19}$$

where $z_{i-1} \sim \mathcal{N}(0, I)$. When we take the limit $N \to \infty$, this discrete Markov chain converges to the continuous SDE:

$$\boxed{dx = -\frac{1}{2}\beta(t)xdt + \sqrt{\beta(t)}dw \tag{2.20}}$$

where

$$\beta_t = \beta_{\text{min}} + t(\beta_{\text{max}} - \beta_{\text{min}}) \tag{2.21}$$

is derived from equation 2.15 and $w(t)$ is still a standard Wiener process with $t \in [0, 1]$

This formulation is known as the **Variance Preserving (VP) SDE** (see complete derivations in Appendix A.2) because, unlike the VE case, the variance remains bounded when t → ∞.

sub-Variance Preserving SDE

Building upon the VP framework, Song et al. proposed a third SDE that offers improved performance for likelihood computation:

$$\boxed{dx = -\frac{1}{2}\beta(t)xdt + \sqrt{\beta(t)(1 - e^{-2\int_0^t \beta(s)ds})}dw \tag{2.22}}$$

This **sub-VP SDE** maintains the same drift as VP but modifies the diffusion term. The crucial property is that its variance remains bounded by the VP SDE at all intermediate time steps, hence the "sub" designation. This modification often yields superior likelihood values while preserving the computational advantages of affine drift coefficients.

In the following, we denote by $p_t(x)$ the probability density of $x(t)$, and by $p_{st}(x(t) \mid x(s))$ the transition kernel from time s to t, with $0 \leq s < t \leq T$.

2.3 The Three SDE Families: VE, VP, sub-VP

The transition from discrete noise perturbations (SMLD and DDPM) to continuous-time formulations leads naturally to a family of stochastic differential equations that capture different noise injection behaviors: Variance Exploding (VE), Variance Preserving (VP), and sub-Variance Preserving (sub-VP) SDEs. Each SDE family offers distinct mathematical properties and practical advantages.

General Form of Forward SDEs

All three SDE families can be written in the general form:

$$\boxed{dx = f(x,t)dt + g(t)dw \qquad (2.23)}$$

where $f(x,t) : \mathbb{R}^d \times [0,T] \to \mathbb{R}^d$ is the drift coefficient, $g(t) : [0,T] \to \mathbb{R}$ is the diffusion coefficient, and w is a standard Wiener process.

From the general theory of stochastic differential equations, the three SDEs (VE, VP, and sub-VP) admit unique strong solutions under standard regularity conditions. Specifically, the drift and diffusion coefficients in each formulation are globally Lipschitz and bounded, ensuring both existence and pathwise uniqueness of solutions according to the classical results such as the Itô existence and uniqueness theorem.

Furthermore, since VE, VP and sub-VP SDEs all have affine drift coefficients, their perturbation kernels $p_{0t}(x(t)|x(0))$ are all Gaussian and can be computed in closed-forms, as discussed in 2.5

Variance Exploding (VE) SDE

The VE-SDE corresponds to the continuous limit of SMLD and is characterized by:

$$dx = \sqrt{\frac{d[\sigma^2(t)]}{dt}}dw \tag{2.24}$$

Key Properties (VE-SDE):

- Drift coefficient: $f(x,t) = 0$ (no drift)

- Diffusion coefficient: $g(t) = \sigma(t)\sqrt{2\log\frac{\sigma_{\max}}{\sigma_{\min}}}$ for $t \in (0,1]$ and $\sigma(t)$ is defined in 2.18

- Mean: $\mathbb{E}[x(t)] = x(0)$

- Variance: $\mathrm{Var}[x(t)] = \sigma^2(t)I$ where $\sigma^2(t) \to \infty$ as $t \to \infty$

The VE-SDE is not differentiable at $t = 0$ because $\sigma(0) = 0$ but $\sigma(0^+) = \sigma_{\min} \neq 0$.
In practice, we solve the SDE in the range $t \in [\epsilon, 1]$ for some small $\epsilon > 0$.

Variance Preserving (VP) SDE

The VP-SDE corresponds to the continuous limit of DDPM and is given by:

$$dx = -\frac{1}{2}\beta(t)xdt + \sqrt{\beta(t)}dw \tag{2.25}$$

Key Properties (VP-SDE):

- Drift coefficient: $f(x,t) = -\frac{1}{2}\beta(t)x$ (linear drift) with $\beta(t)$ defined in 2.21

- Diffusion coefficient: $g(t) = \sqrt{\beta(t)}$

- Mean: $\mathbb{E}[x(t)] = x(0)e^{-\frac{1}{2}\int_0^t \beta(s)ds}$

- Variance: $\mathrm{Var}[x(t)] = [1 - e^{-\int_0^t \beta(s)ds}]I$

The VP-SDE has a fixed variance for all $t \in [0,\infty)$ when $p(x(0))$, data distribution has unit variance. Since the VP-SDE has affine drift and diffusion coefficients, we can use standard techniques to obtain the perturbation kernel.

sub-Variance Preserving (sub-VP) SDE

The sub-VP SDE is a new type of SDE derived by Yang Song et al., designed to achieve better likelihood values:

$$dx = -\frac{1}{2}\beta(t)xdt + \sqrt{\beta(t)(1 - e^{-2\int_0^t \beta(s)ds})}dw \tag{2.26}$$

Key Properties (sub-VP SDE):

- Drift coefficient: $f(x,t) = -\frac{1}{2}\beta(t)x$ (same as VP)

- Diffusion coefficient: $g(t) = \sqrt{\beta(t)(1 - e^{-2\int_0^t \beta(s)ds})}$

- Mean: $\mathbb{E}[x(t)] = x(0)e^{-\frac{1}{2}\int_0^t \beta(s)ds}$

- Variance: $\text{Var}[x(t)] = [1 - e^{-\int_0^t \beta(s)ds}]^2 I$

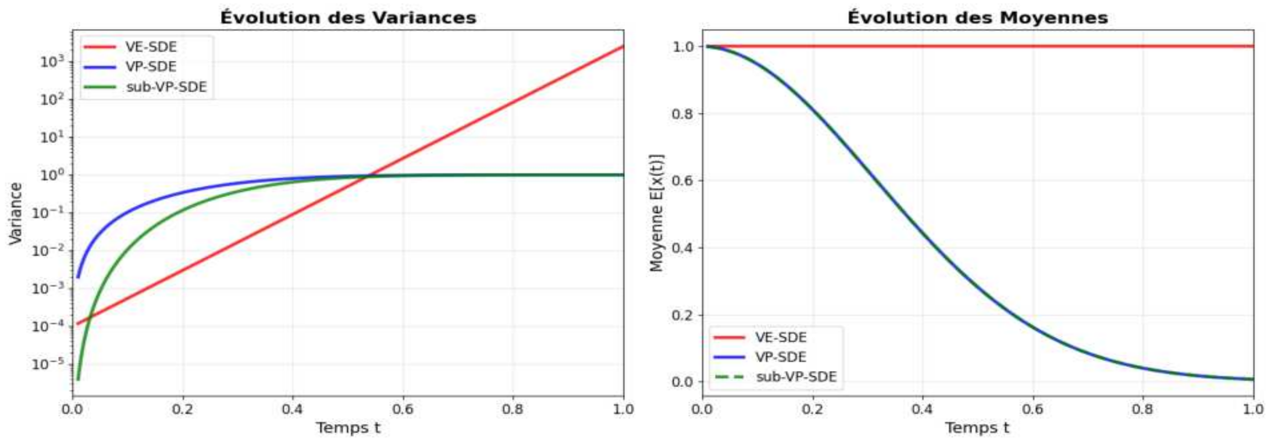
The variance of the sub-VP SDE is always bounded by the VP SDE at every intermediate time step, hence the "sub" designation.

Complete mean and variance calculations for all three SDE are provided in Appendix A.3.

SDE's mean and variance behaviors

As discussed in the previous sections, we can remark from figure 2.4 below that :

- VE-SDE: Variance explodes to infinity, constant mean, suitable for high-dimensional problems
- VP-SDE: Variance remains bounded, vanishing mean
- sub-VP SDE: Variance bounded below VP with vanishing mean while often achieves better likelihood values



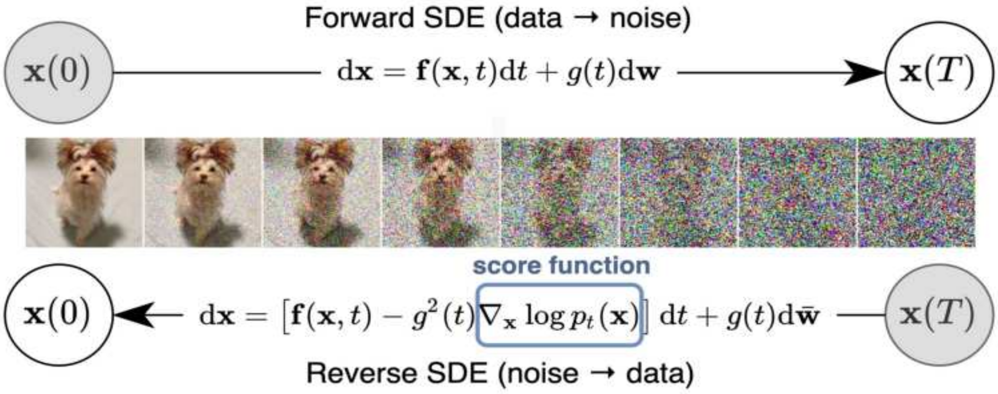
Variance and Mean evolution over time for the three SDE families

Figure 2.4: Comparison of variance and mean behavior for VE, VP, and sub-VP SDEs

The choice between these SDE families depends on the specific application requirements, computational constraints, and desired sample quality characteristics.

2.4 Reverse-Time SDE and Probability Flow ODE

The key insight enabling score-based generative modeling is that any forward diffusion process can be reversed to generate samples. This reversal is mathematically formalized through two equivalent formulations: the reverse-time SDE and the probability flow ODE. Both approaches yield the same marginal distributions but offer different computational trade-offs.


Forward SDE (data → noise)
$\mathbf{x}(0)$
$\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}$
$\mathbf{x}(T)$
score function
$\mathbf{x}(0)$
$\mathrm{d}\mathbf{x} = [\mathbf{f}(\mathbf{x}, t) - g^2(t)\nabla_{\mathbf{x}} \log p_t(\mathbf{x})]\, \mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{w}}$
$\mathbf{x}(T)$
Reverse SDE (noise → data)

The forward SDE transforms data into noise, while the reverse SDE can generate samples by reversing this process using learned score functions.

Figure 2.5: The forward and backward process. Source: [12]

Obviously, at the initial time $t = 0$, we have $p_0(\mathbf{x}) = p(\mathbf{x})$, which represents the original data distribution without any noise corruption. As we apply the stochastic diffusion process over an extended period T, the distribution $p_T(\mathbf{x})$ gradually transforms and eventually approximates a simple, analytically tractable noise distribution $\pi(\mathbf{x})$, known as the **prior distribution**. This final distribution $p_T(\mathbf{x})$ serves the same role as $p_{\sigma_L}(\mathbf{x})$ in discrete noise scale approaches, where σ_L represents the maximum noise level applied to corrupt the original data.

Reverse-Time SDE

A fundamental result from stochastic calculus, due to Anderson (1982) [1], establishes that the reverse of any diffusion process is also a diffusion process. Given a forward SDE of the general form:

$$dx = f(x, t)dt + g(t)dw$$

where $f(x, t) : \mathbb{R}^d \times [0, T] \to \mathbb{R}^d$ is the drift coefficient and $g(t) : [0, T] \to \mathbb{R}$ is the diffusion coefficient, the corresponding reverse-time SDE that runs backwards in time from T to 0 is:

$$\boxed{dx = [f(x, t) - g(t)^2 \nabla_x \log p_t(x)]dt + g(t)d\bar{w} \qquad (2.27)}$$

where $\bar{w}$ is a standard Wiener process when time flows backwards from T to 0, and dt represents an infinitesimal negative timestep.
The complete existence and uniqueness conditions are detailed in Appendix A.4.

The crucial observation is that the reverse SDE depends on the score function $\nabla_x \log p_t(x)$ of the marginal distributions at each time t. This is precisely what enables our approach: by learning to estimate these score functions, we can simulate the reverse process to generate samples from $p_0(x)$.

Intuition Behind the Reverse Process

The reverse-time SDE essentially "undoes" the forward diffusion process. Starting from pure noise $x(T) \sim p_T$, the reverse process gradually removes noise and adds structure, eventually producing a sample from the data distribution $p_0(x)$.
The score term $\nabla_x \log p_t(x)$ acts as a "steering" force for the the reverse process at each timestep.

Probability Flow ODE

While the reverse-time SDE provides a stochastic sampling procedure, Yang Song et al. discovered a remarkable connection: there exists a deterministic ordinary differential equation [2] that produces samples from the same marginal distributions as the reverse SDE.
For any forward SDE with the form of equation 2.23, the corresponding probability flow is :

$$\boxed{dx = \left[f(x,t) - \frac{1}{2}g(t)^2\nabla_x \log p_t(x)\right] dt \qquad (2.28)}$$

The key insight is that this ODE shares the same marginal probability densities $\{p_t(x)\}_{t=0}^{T}$ as the original SDE, but follows deterministic trajectories. (partial derivation in Appendix A.5).

Advantages of the Probability Flow Formulation

The probability flow ODE offers several computational and theoretical advantages over the stochastic reverse-time SDE :

Exact Likelihood Computation

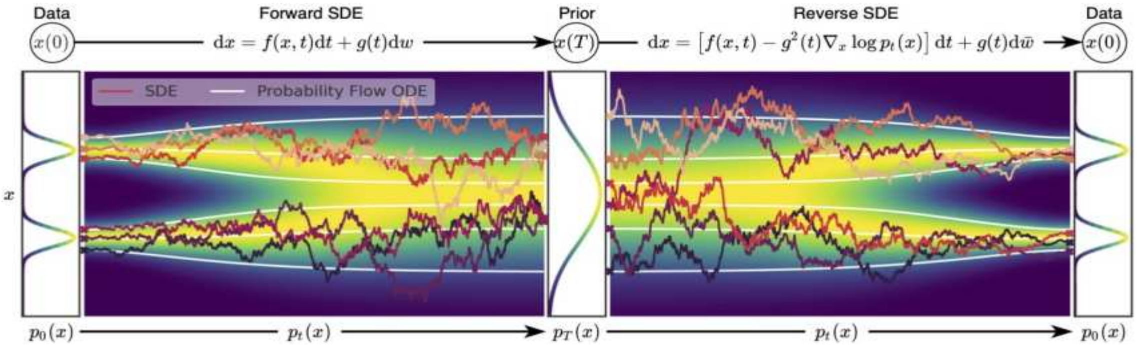
One of the most significant advantages is the ability to compute exact likelihoods. Using the instantaneous change of variables formula from neural ODE theory (Chen et al., 2018), we can compute:

$$\boxed{\log p_0(x(0)) = \log p_T(x(T)) + \int_0^T \nabla \cdot \tilde{f}_\theta(x(t),t)dt \qquad (2.29)}$$

where $x(t)$ is obtained by solving the probability flow ODE. The divergence $\nabla \cdot \tilde{f}_\theta(x,t)$ can be efficiently estimated using the Skilling-Hutchinson trace estimator, making likelihood computation tractable.

Faster Sampling

The deterministic nature of the ODE allows for more efficient numerical integration. Unlike stochastic samplers that require many small steps to maintain accuracy, ODE solvers can adaptively choose step sizes and employ higher-order methods.



The forward SDE transforms data into noise, while both the reverse SDE and probability flow ODE can generate samples by reversing this process using learned score functions.

Figure 2.6: Overview of score-based generative modeling through SDEs showing both forward SDE, reverse SDE, and probability flow ODE paths. Source: [12]

2.5 Learning the Score Function

While the theoretical framework of reverse-time SDEs and probability flow ODEs provides the mathematical foundation for score-based generative modeling, the practical success of these methods hinges on our ability to accurately estimate the score function $\nabla_x \log p_t(x)$. This section explores how score functions can be learned from data, starting from the fundamental score matching approach and leading to the denoising score matching objective that forms the core of modern implementations.

Naive Score Matching

The most direct approach to learning score functions was introduced by Hyvärinen [6] in the context of estimating unnormalized probability models. Given access to samples from a data distribution $p_{data}(x)$, we might naively attempt to learn the score function by minimizing:

$$\mathcal{L}_{naive}(\theta) = \frac{1}{2}\mathbb{E}_{x\sim p_{data}}\left[\|s_\theta(x) - \nabla_x \log p_{data}(x)\|^2\right] \tag{2.30}$$

where $s_\theta(x)$ is a parametrized function such that $s_\theta(x) \approx \nabla_x \log p_{data}(x)$ with a Deep Neural Network (DNN).

However, this objective is fundamentally impractical because it requires knowledge of the true score $\nabla_x \log p_{data}(x)$, which we are trying to estimate in the first place.

Denoising Score Matching

The key insight that enables practical score estimation comes from considering the noise-perturbed versions of the data, as explored in SMLD and DDPM. Instead of directly estimating $\nabla_x \log p_{data}(x)$, we estimate the score of noise-perturbed distributions.
The transition from discrete noise scales to continuous-time SDEs requires a corresponding generalization of the denoising score matching objective.

For a general SDE with perturbation kernel $p_{0t}(x(t)|x(0))$, we can show thanks to denoising score matching introduced by A. Hyvarinen [6] that the continuous score matching objective becomes (derivation of denoising score matching objective detailed in Appendix A.6) :

$$\boxed{\theta^* = \arg\min_\theta \mathbb{E}_t \left\{\lambda(t)\mathbb{E}_{x(0)\sim p_0}\mathbb{E}_{x(t)|x(0)}\left[\left\|s_\theta(x(t),t) - \nabla_{x(t)} \log p_{0t}(x(t)|x(0))\right\|^2\right]\right\} \qquad (2.31)}$$

where: - t is uniformly sampled over $[0, T]$
- $\lambda(t) : [0, T] \to \mathbb{R}_{>0}$ is a positive weighting function
- $x(0) \sim p_0(x)$ represents clean data samples
- $x(t) \sim p_{0t}(x(t)|x(0))$ represents noisy samples at time t

The weighting function $\lambda(t)$ plays an important role in balancing the importance of score estimation across different noise levels. Following the approach established in SMLD and DDPM, a common choice is:

$$\lambda(t) \propto \frac{1}{\mathbb{E}\left[\|\nabla_{x(t)} \log p_{0t}(x(t)|x(0))\|^2\right]} \tag{2.32}$$

This choice ensures that the loss is properly normalized across different time steps, preventing any single noise level from dominating the training process.

Closed-Form Scores for Affine SDEs

Since the VE (2.17), VP (2.20) and sub-VP (2.22) SDEs have all affine drift coefficients and deterministic diffusion coefficients, then $x(t)$ is Gaussian and their perturbation kernel $p_{0t}(x(t)|x(0))$ are all Gaussian and can be computed since we have computed the mean and variance of $x(t)$ for the three equations.
The perturbation kernels are:

VE SDE:

$$p_{0t}(x(t)|x(0)) = \mathcal{N}(x(t); x(0), \sigma^2(t)I) \tag{2.33}$$

VP SDE:

$$p_{0t}(x(t)|x(0)) = \mathcal{N}(x(t); x(0)e^{-\frac{1}{2}\int_0^t \beta(s)ds}, [1 - e^{-\int_0^t \beta(s)ds}]I) \tag{2.34}$$

sub-VP SDE:

$$p_{0t}(x(t)|x(0)) = \mathcal{N}(x(t); x(0)e^{-\frac{1}{2}\int_0^t \beta(s)ds}, [1 - e^{-\int_0^t \beta(s)ds}]^2 I) \tag{2.35}$$

where we choose $\Sigma(0) = 0$.

Let assume that $p_{0t}(x(t)|x(0)) = \mathcal{N}(x(t); \mu(t)x(0), \Sigma(t)I)$,

then, $\quad x(t) = \mu(t)x(0) + \sqrt{\Sigma(t)}z(t)$, where $z(t)$ is a standard gaussian.

$$\Rightarrow \nabla_{x(t)} \log p_{0t}(x(t)|x(0)) = -\frac{z(t)}{\sqrt{\Sigma(t)}}$$

Finally, the objective function becomes:

$$\boxed{\theta^* = \arg\min_\theta \mathbb{E}_t \left\{ \lambda(t)\mathbb{E}_{x(0)}\mathbb{E}_{x(t)|x(0)} \left[\left\| s_\theta(x(t), t) + \frac{z(t)}{\sqrt{\Sigma(t)}} \right\|_2^2 \right] \right\} (2.36)}$$

This procedure is computationally efficient because it only requires forward evaluation of the score network and does not involve running the full SDE dynamics during training. The ability to compute target scores in closed form for affine SDEs makes this approach particularly tractable and forms the foundation of the practical success of score-based generative models.

2.6 Sampling Strategies

Once we have trained a score-based model $s_\theta^*(x, t)$ to approximate $\nabla_x \log p_t(x)$, the next crucial step is to solve the reverse-time SDE to generate samples. This requires numerical methods to integrate the stochastic differential equation backwards in time from $t = T$ to $t = 0$. The choice of sampling strategy significantly impacts both the quality of generated samples and the computational efficiency of the process.

The Backward SDE Integration Challenge

Given the reverse-time SDE:

$$dx = [f(x, t) - g(t)^2\nabla_x \log p_t(x)]dt + g(t)d\bar{w}$$

the fundamental challenge is to solve this equation numerically while maintaining accuracy and computational efficiency. Unlike forward SDEs where we can often use standard numerical schemes, the reverse-time nature and the dependence on learned score functions create unique computational demands.

Predictor-Corrector Framework

A significant advancement in SDE-based sampling is the Predictor-Corrector (PC) framework. This approach combines numerical SDE solvers with score-based MCMC methods to improve sample quality.

The PC framework alternates between two steps:
1. **Predictor step**: Uses a numerical SDE solver to advance the solution
2. **Corrector step**: Applies score-based MCMC (e.g., Langevin dynamics) to correct the marginal distribution

Predictor-Corrector Sampling Algorithm

Require: N (discretization steps), M (corrector steps), score model $s_\theta(x, t)$

```
Initialize $x_N \sim \mathcal{N}(0, \sigma^2_{\max} I)$
for $i = N - 1$ to $0$ do
    $x_i \leftarrow$ Predictor($x_{i+1}$)
    for $j = 1$ to $M$ do
        $x_i \leftarrow$ Corrector($x_i$)
    end for
end for
return $x_0$
```

The predictor advances the solution using numerical SDE solvers, while the corrector refines the marginal distribution using score-based MCMC methods.

In the next section, we present some predictors and correctors used in practice

Classical Numerical SDE Solver as predictors

Euler-Maruyama Method

The most straightforward approach is the Euler-Maruyama (EM) method, which discretizes the reverse SDE using fixed step sizes. For a time discretization with step size h, the EM scheme becomes:

$$x_{i-1} = x_i + h[f(x_i, t_i) - g(t_i)^2 s^*_\theta(x_i, t_i)] + \sqrt{h} g(t_i) z_i \tag{2.37}$$

where $z_i \sim \mathcal{N}(0, I)$ and $t_i = ih$ for $i = N, N - 1, \ldots, 1$.

While EM is simple to implement and theoretically well-understood, it suffers from several limitations:
- Requires very small step sizes for stability, leading to many function evaluations
- Fixed step size cannot adapt to the varying complexity of the score function
- Poor performance in high-dimensional spaces typical of image generation

Reverse Diffusion Sampling

Yang Song et al. proposed reverse diffusion samplers that discretize the reverse-time SDE using the same discretization strategy as the forward process. Given a forward SDE:

$$dx = f(x,t)dt + G(t)dw \tag{2.38}$$

and its discretization:

$$x_{i+1} = x_i + f_i(x_i) + G_i z_i, \quad i = 0, 1, \dots, N-1 \tag{2.39}$$

where $z_i \sim \mathcal{N}(0, I)$, Yang Song et al. propose to discretize the reverse-time SDE:

$$dx = [f(x,t) - G(t)G(t)^T \nabla_x \log p_t(x)]dt + G(t)d\bar{w} \tag{2.40}$$

with a similar functional form, giving the following iteration rule:

$$x_i = x_{i+1} - f_{i+1}(x_{i+1}) + G_{i+1}G_{i+1}^T s_\theta^*(x_{i+1}, i+1) + G_{i+1}z_{i+1} \tag{2.41}$$

where $s_\theta^*(x_i, i)$ is the trained score-based model conditioned on iteration number i.
When applied to VE and VP SDEs, this yields specific sampling algorithms. These methods are named (based on this discretization strategy) **reverse diffusion samplers**. This approach ensures consistency between forward and backward processes and often performs better than generic EM schemes.

Fast Adaptive Sampling: "Gotta Go Fast" Method

While traditional methods like EM or Reverse diffusion use fixed step sizes, recent work by Jolicoeur-Martineau et al. [7] demonstrated that adaptive step size methods can dramatically improve both speed and quality for score-based generative models. Their approach addresses key challenges specific to high-dimensional image generation.

- **Motivation for Adaptive Methods**

Standard SDE solvers face unique challenges when applied to score-based generative models:
1. *High dimensionality*: Image generation involves SDEs in extremely high-dimensional spaces (e.g., $256 \times 256 \times 3 = 196,608$ dimensions)
2. *Expensive score evaluations*: Each step requires a forward pass through a large neural network
3. *Reduced precision requirements*: Visual quality tolerance allows for larger errors than traditional numerical applications

- **The Fast Sampling Algorithm**

The core innovation is an adaptive step size SDE solver that uses two score function evaluations per step and employs extrapolation to achieve higher-order accuracy. The algorithm maintains two integration schemes:

−− **Euler-Maruyama step** (lower-order):

$$x' = x - hf(x,t) + hg(t)^2 s_\theta^*(x,t) + \sqrt{h}g(t)z \tag{2.42}$$

−− **Improved Euler step** (higher-order):

$$\boxed{\tilde{x} = x - hf(x', t-h) + hg(t-h)^2 s_\theta^*(x', t-h) + \sqrt{h}g(t-h)z \qquad (2.43)}$$

−− **Extrapolated result**:

$$\boxed{x'' = \tfrac{1}{2}(x' + \tilde{x}) \qquad (2.44)}$$

- **Error Control and Step Size Adaptation**
  The algorithm estimates the local error using:

$$E_2 = \sqrt{\frac{1}{n} \left\| \frac{x' - x''}{\delta(x', x'_{\text{prev}})} \right\|_2^2} \tag{2.45}$$

where:
- n is the dimensionality of the data (e.g., $n = 256 \times 256 \times 3$ for RGB images)
- x' is the Euler-Maruyama estimate (lower-order)
- x'' is the extrapolated result from improved Euler (higher-order)
- x'_{prev} is the previous state from the last accepted step
- $\delta(x', x'_{\text{prev}})$ is the mixed tolerance defined element-wise as:

$$\delta(x', x'_{\text{prev}}) = \max(\text{abs}, \text{rel} \times \max(|x'|, |x'_{\text{prev}}|)) \tag{2.46}$$

Complete parameter specifications and performance comparisons are provided in Appendix A.7.

Langevin Dynamics as Corrector

The corrector step typically employs Langevin dynamics:

$$x_i^{(m)} = x_i^{(m-1)} + \epsilon_i s_\theta^*(x_i^{(m-1)}, t_i) + \sqrt{2\epsilon_i} z_i^{(m)} \tag{2.47}$$

where $z_i^{(m)} \sim \mathcal{N}(0, I)$ represents Gaussian noise at corrector step m, and the step size ϵ_i is determined by the signal-to-noise ratio (SNR) approach.
The step size ϵ_i is computed using the signal-to-noise ratio r (also called target SNR):

$$\epsilon_i = \frac{2(r \times \|z_i\|_2)^2}{\|s_\theta^*(x_i, t_i)\|_2^2} \times \alpha_i \tag{2.48}$$

where:
- r is the signal-to-noise ratio hyperparameter (typically $r \in [0.01, 0.22]$ depending on the SDE)
- α_i is a scaling factor that equals 1 for VE-SDE and α_i (from the VP-SDE formulation) for VP/sub-VP SDEs

2.7 State of the Art

Having established the theoretical foundations of score-based generative models using SDEs, it is essential to contextualize our approach within the rapidly evolving landscape of diffusion-based generation methods. The field has experienced remarkable progress since the foundational work of Song et al. (2021)[14], with significant advances in both theoretical understanding and practical applications that directly inform our research choices and implementation strategies.

Evolution of Score-Based and Diffusion Models

The landscape of score-based generative models has evolved rapidly since the foundational work of [14], with significant advances in both theoretical understanding and practical applications.

Architectural Progressions: The evolution from NCSN to NCSN++ demonstrated improvements in sample quality through better architectural choices, while modern variants like EDM (Elucidating the Design Space of Diffusion-Based Generative Models) by Karras et al. have refined theoretical foundations and optimal training configurations. These advances directly motivated our choice of the NCSN++ continuous architecture for our VE-SDE implementation, as it has demonstrated superior performance for high-resolution image generation tasks.

DDPM Variants and Improvements: Building upon Ho et al.'s [5] original DDPM framework, numerous improvements have emerged. DDPM++ variants show particular promise for likelihood computation while maintaining sample quality, demonstrating the continued relevance of discrete-time perspectives alongside continuous SDE formulations. However, for our research focus on VE-SDE implementation and classifier guidance, the continuous formulation provides the mathematical elegance and flexibility needed for our downstream applications.

Classifier-Free Guidance and Recent Advances

A major development has been the widespread adoption of classifier-free guidance (CFG) [4], which addresses many limitations of classifier-based approaches like the one we implement in this work.

Classifier-Free Guidance (CFG): CFG eliminates separate classifier training by jointly training conditional and unconditional models, then combining their score estimates during sampling. This approach has become standard in modern text-to-image systems, avoiding the hyperparameter sensitivity we encounter with classifier guidance. However, CFG requires additional computational resources during training and more complex model architectures.

CFG Improvements and Variants: Recent work has addressed CFG's fundamental limitations. CFG++ emerged as a solution, reformulating text-guidance as an inverse problem and enabling effective use of smaller guidance scales ($0 < \lambda < 1$). PostCFG and other alternatives demonstrate ongoing evolution of guidance techniques. While these methods represent more advanced approaches, our focus on classifier-based guidance provides valuable insights into the fundamental challenges of conditional generation and offers a more accessible implementation for research purposes.

Why We Focus on Classifier Guidance: Despite the emergence of CFG, classifier-based guidance remains relevant for several reasons. First, it allows us to explore the fundamental score decomposition $\nabla_x \log p_t(x|y) = \nabla_x \log p_t(x) + \nabla_x \log p_t(y|x)$ in a direct and interpretable manner. Second, it provides educational value in understanding how external knowledge can be integrated into the sampling process. Finally, our adaptive strategies for improving classifier effectiveness contribute to the broader understanding of guidance mechanisms in score-based models.

Modern Text-to-Image Applications and Implementation Choices

The practical applications of score-based models have expanded dramatically, with commercial systems achieving unprecedented quality and adoption.

Leading Commercial Systems: Modern text-to-image models like DALL-E, Midjourney, and Stable Diffusion all employ diffusion-based architectures, confirming the centrality of score-based approaches to current state-of-the-art generation. Stable Diffusion, powered by Latent Diffusion Models (LDM), represents a significant advancement by operating in learned latent spaces rather than pixel space, addressing computational efficiency concerns similar to those we encounter in our high-resolution experiments.

Our Research Focus Within This Context: Given this landscape, our work focuses on several key areas that complement existing developments:

- **VE-SDE Implementation:** We concentrate on the VE-SDE formulation due to its empirical robustness and mathematical elegance, providing insights into scaling behavior across multiple resolutions (256×256 to 1024×1024).
- **Custom Training from Scratch:** Our AFHQ-512 model training from scratch provides practical insights into resource requirements and training dynamics that complement the typically pre-trained models used in research.
- **Adaptive Classifier Guidance:** While CFG dominates current applications, our development of adaptive classifier guidance strategies (amplified scaling, adaptive linear scaling, sigma truncation) contributes to understanding guidance effectiveness zones and classifier limitations across noise spectrums.
- **Comprehensive Downstream Applications:** Our implementation covers the full spectrum of score-based applications—unconditional sampling, inpainting, colorization, and conditional generation—using a unified framework that demonstrates the versatility of the SDE approach.

Positioning Relative to Current Methods: Our approach represents a comprehensive study of the foundational SDE framework that underlies many modern advances. While commercial systems have moved toward more complex architectures and training procedures, understanding the core mechanisms through direct implementation of VE-SDE provides valuable insights into the fundamental principles that drive these more advanced systems. Our work bridges the gap between theoretical understanding and practical implementation, offering lessons in resource management, training strategies, and guidance mechanisms that remain relevant regardless of architectural sophistication.

This contextual understanding shapes our experimental design and guides our focus toward areas where our implementation can provide meaningful contributions to the field's understanding of score-based generative modeling fundamentals.

3 Model Implementation and Architecture

3.1 Codebase Overview

Our implementation builds upon the official PyTorch codebase from Yang Song et al. [14], providing a modular and extensible framework for score-based generative modeling. The codebase uses ml_collections for configuration management and implements auto-adaptive configurations that automatically detect GPU capabilities and optimize batch sizes accordingly.

The framework is organized around several key components:

- Model definitions in `models/` directory, SDE implementations in `sde_lib.py`.
- Sampling algorithms in `sampling.py`, and dataset handling in `datasets.py`.
- Training is orchestrated through `run_lib.py`, which provides a unified interface for model training, evaluation, and checkpointing.

Each dataset and model combination has its own configuration file, facilitating reproducible experiments and systematic comparison across different settings.

3.2 Score Model Architecture

For this work, we employ the **NCSN++** continuous architecture, which has proven superior for high-resolution image generation tasks. The choice of NCSN++ is motivated by its specific design for VE-SDE training and its demonstrated effectiveness in generating high-quality samples at resolutions up to 1024×1024.

A critical component for training stability is the Exponential Moving Average (EMA) of model parameters. Following Yang Song et al.'s empirical findings, we use an EMA rate of 0.9999 for VE-SDE models. This higher EMA rate ensures smoother convergence and improved sample quality during the extended training required for high-resolution datasets.

Exponential Moving Average (EMA)

Mechanism: Maintain smoothed version of model weights over time

Update rule during training:

$$\theta_t^{\text{EMA}} \leftarrow \text{decay} \cdot \theta_{t-1}^{\text{EMA}} + (1 - \text{decay}) \cdot \theta_t^{\text{model}}$$

At sampling: Use θ_T^{EMA} instead of θ_T^{model}

For VE-SDE: decay = 0.9999 gives more importance to recent weights while ensuring smooth parameter evolution. Small decay value allows gradual adaptation to correct bias.

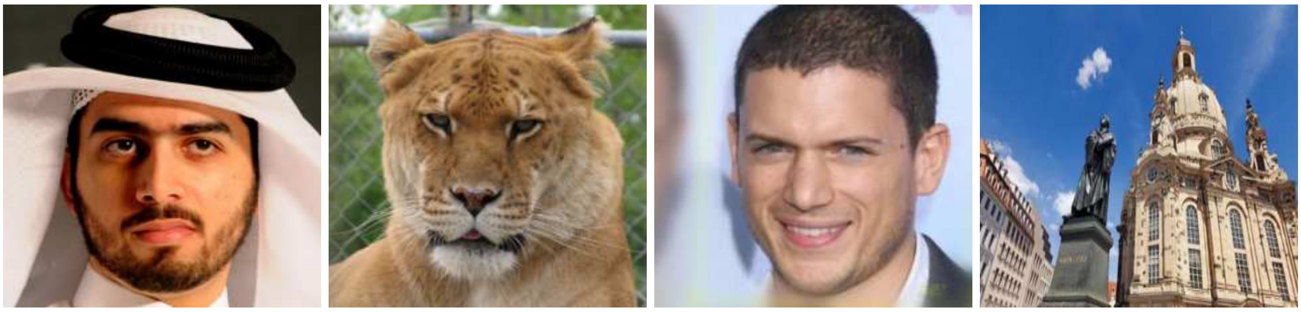
3.3 Training Configuration and Datasets

Dataset Selection Strategy

Our experimental design encompasses multiple datasets at different resolutions to evaluate the scalability and effectiveness of score-based generative models across various domains.

The selection includes FFHQ-1024 [8] for high-resolution human faces, AFHQ-512 [3] for animal faces, CelebA-HQ-256 [9]for celebrity faces, and LSUN Church-256 [15] for architectural imagery.

This multi-resolution approach allows us to assess the computational trade-offs between image quality and training feasibility. Higher resolutions demand significantly more computational resources but enable the generation of fine-grained details that are crucial for practical applications. The diversity of domains—human faces, animal faces, and architecture—provides insights into the generalization capabilities of the VE-SDE framework across different visual modalities.



FFHQ-1024: High-resolution faces. Source : [8]

AFHQ-512: Animal faces (cats, dogs, wild). Source : [3]

CelebA-HQ-256: Celebrity faces. Source : [9]

LSUN Church-256: Architectural imagery. Source : [15]

Figure 3.1: Representative samples from the four datasets used in our experiments, showcasing the diversity of domains and resolutions.

Hyperparameter Configuration

The following table summarizes the key training parameters for each dataset:

Dataset	**Resolution**	$\sigma_{\min}$	$\sigma_{\max}$	$N_{\textbf{scales}}$	**Batch**	**Iterations**	**Status**
AFHQ-512	512 × 512	0.01	375	2000	32	520K	Own training
FFHQ-1024	1024 × 1024	0.01	1348	2000	8	2.4M	Pre-trained
CelebA-HQ	256 × 256	0.01	348	2000	64	2.4M	Pre-trained
LSUN Church	256 × 256	0.01	380	2000	64	2.4M	Pre-trained

Table 3.1: Training hyperparameters for different datasets

The maximum noise level $\sigma_{\max}$ in the noise schedule $\sigma(t) = \sigma_{\min}\left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)^t$ is determined by the maximum Euclidean distance between samples in the dataset $\{x^{(i)}\}_{i=1}^{N}$:

$$\sigma_{\max} \approx \max_i \sum_{j=1}^{N} \|x^{(i)} - x^{(j)}\|_2 \tag{3.1}$$

where $x^{(i)}$ represents the i-th data sample (flattened image vector) from the training dataset. For computational efficiency, we sampled 4000 images per class from the AFHQ training set and calculated all pairwise distances between flattened image vectors.
This yielded $\boldsymbol{\sigma_{\max}} = \mathbf{375}$, ensuring the forward diffusion process reaches sufficient noise levels to learn the score function and reverse the process perfectly.

AFHQ-512 Training Details

The AFHQ (Animal Faces-HQ) dataset consists of high-quality animal face images across three categories: cats, dogs, and wild animals. The dataset contains approximately 15,000 images at 512×512 resolution, with roughly 5,000 images per category. Originally introduced by Choi et al. [3] for few-shot image generation tasks, AFHQ provides a challenging benchmark for generative models due to the diversity of animal species, poses, and backgrounds within each category.

The training was planned for 750,000 iterations but concluded at 520,000 iterations when visual quality assessment and TensorBoard monitoring indicated satisfactory convergence.

Training progress was monitored through multiple metrics including training loss, evaluation loss, and periodic visual sampling. The absence of FID computation during training was compensated by careful visual assessment and loss curve analysis, which provided sufficient indication of model convergence and quality.

Numerical stability considerations were particularly important for VE-SDE training, requiring careful handling of the $t \to 0$ limit where the diffusion coefficient approaches zero. We employed $\varepsilon = 1\text{e}{-5}$ as a small positive lower bound to prevent numerical instabilities while maintaining the mathematical integrity of the continuous formulation.

3.4 Sampling Configuration

For sample generation, we employ the Predictor-Corrector (PC) framework with specific choices optimized for high-resolution image synthesis. Our sampling configuration uses the reverse-diffusion predictor developed by Yang Song et al. combined with Langevin dynamics as the corrector.

Due to computational resource limitations, we could not calculate FID scores, which require substantial computational power that was not available to us. Therefore, our evaluation relies primarily on visual quality assessment of the generated samples.

In our study, we compared two sampling approaches: reverse-diffusion and fast adaptive sampling. The reverse-diffusion method demonstrated superior visual quality in the generated images. While fast sampling showed significantly better computational efficiency (3-4 times faster according to our logs), it produced slightly lower visual quality, though the difference was often not substantial. Given our focus on sample quality over speed, we selected the reverse-diffusion predictor as our primary sampling method.

For the corrector, we use SNR = 0.16 for AFHQ-512 experiments. This signal-to-noise ratio provides an effective balance between correction strength and sampling stability based on several tests where SNR $\in \{0.15, 0.16\}$ offers the best trade-offs. The number of corrector steps is typically set to 1 per predictor update, following the established practice in the literature.

The sampling process is configured for 2000 discretization steps for high-resolution datasets (FFHQ-1024, AFHQ-512), with noise removal enabled to improve final sample quality. The specific choice of 2000 steps represents a balance between sample quality and computational cost, providing sufficient resolution of the reverse-time dynamics while maintaining practical sampling times.

Here is a link to the github repository, https://github.com/score-sde-project, that contains all materials and code run in this project.

4 Results and Analysis

4.1 Sample Quality Assessment

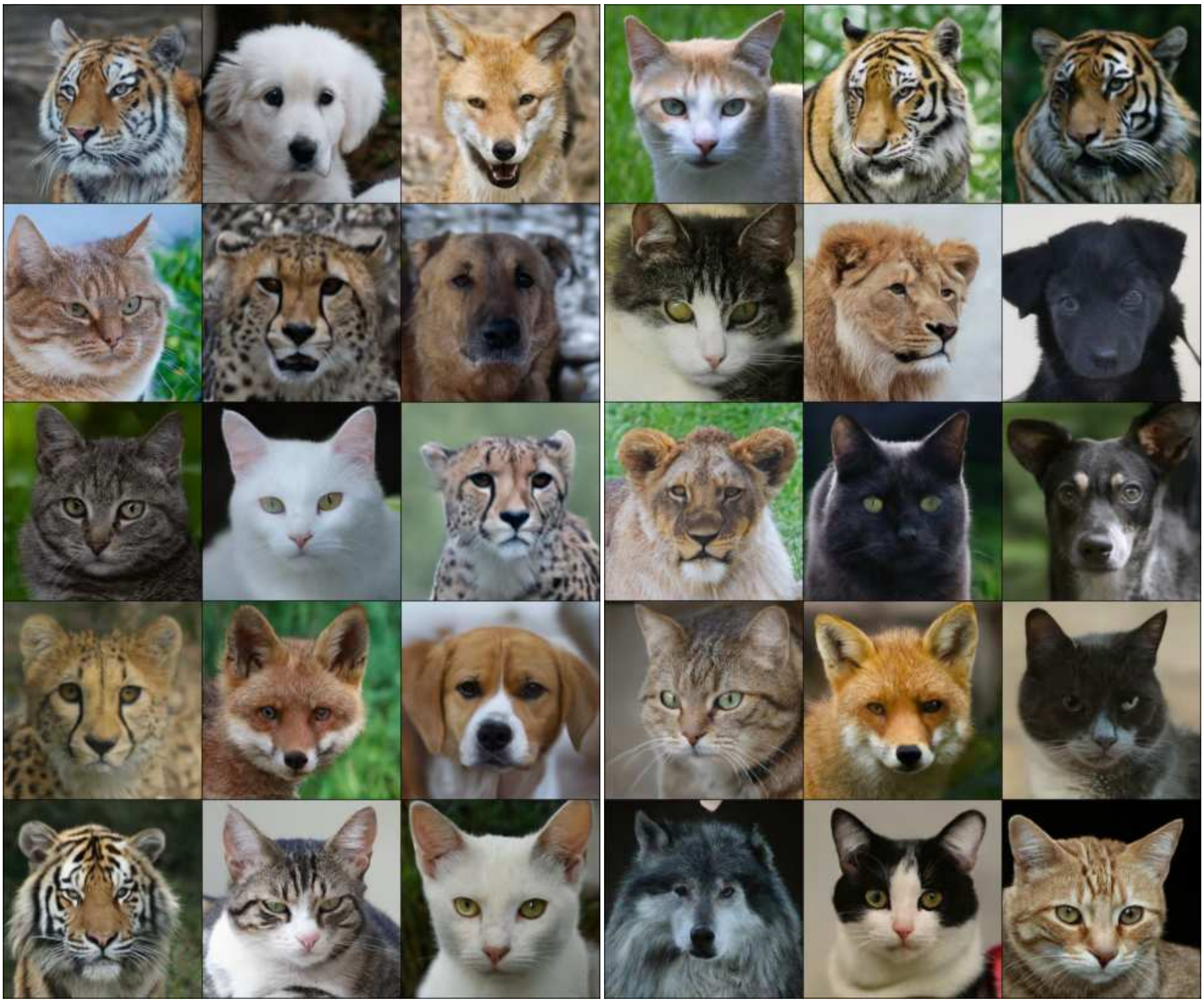
This chapter presents the experimental results obtained through our implementation of score-based generative models. We evaluate the performance across multiple datasets and tasks, providing both qualitative and quantitative analysis of the generated samples. The evaluation encompasses unconditional sampling, various downstream tasks, and our main contribution: adaptive classifier guidance strategies.

All samples in this chapter and the next have been generated with my own means.

Visual Results

The quality of generated samples serves as the primary indicator of model performance. We present unconditional samples from all four datasets, showcasing the diversity and fidelity achieved by our VE-SDE implementation across different domains and resolutions.

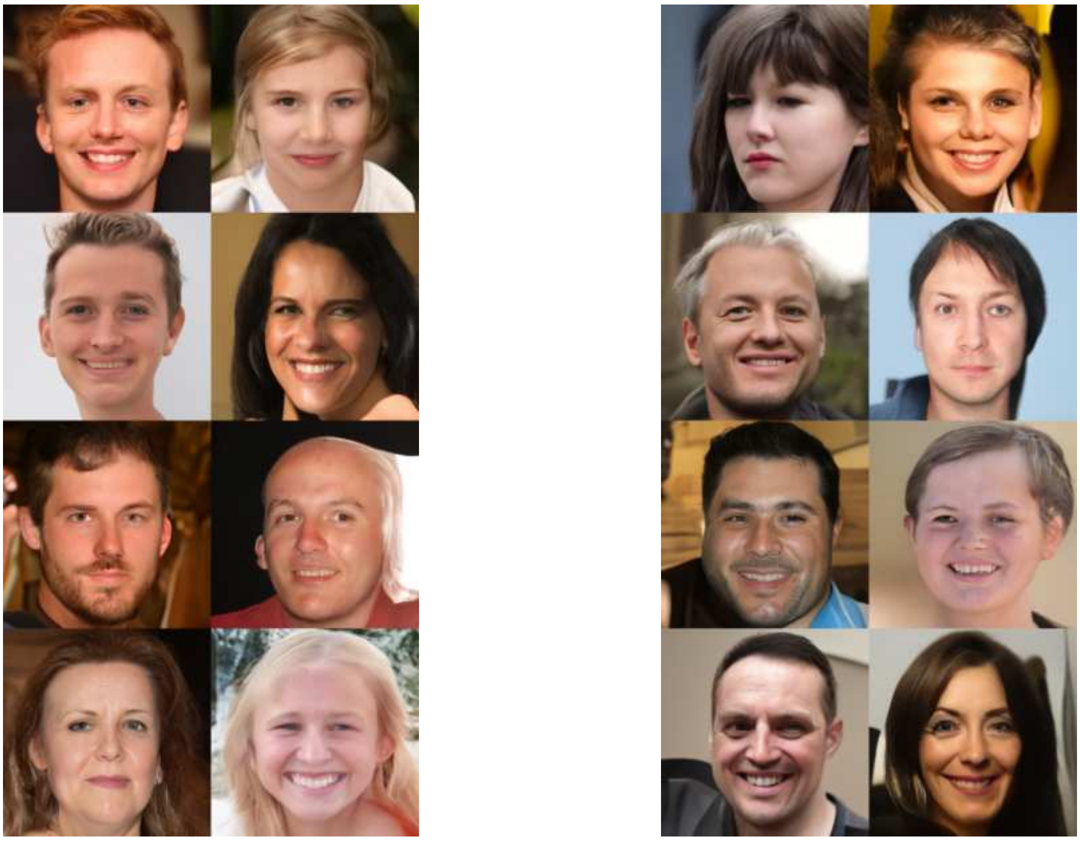
Unconditional Sample Quality



AFHQ-512 samples (Grid 1)

AFHQ-512 samples (Grid 2)

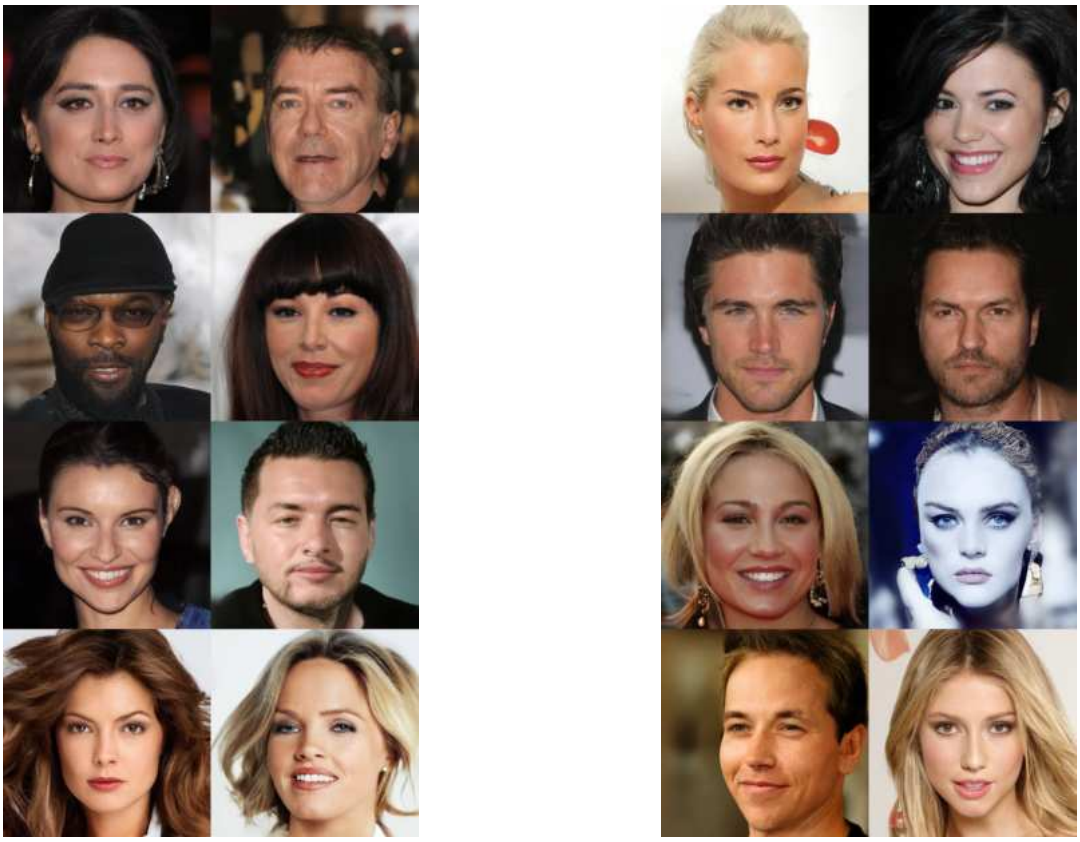
Figure 4.1: AFHQ-512 unconditional samples from our custom-trained model showing diverse animal faces



FFHQ-1024 samples (Grid 1)

FFHQ-1024 samples (Grid 2)

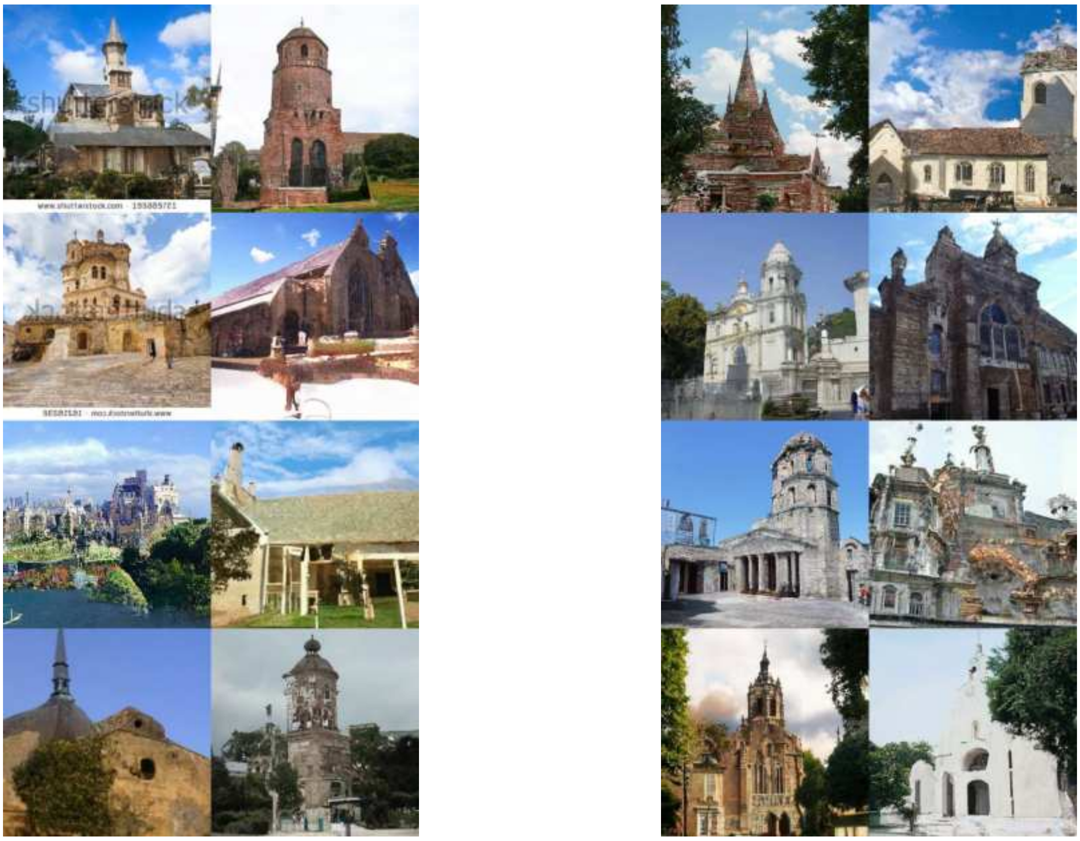
Figure 4.2: FFHQ-1024 unconditional samples showing high-resolution human face generation



CelebA-HQ-256 samples (Grid 1)

CelebA-HQ-256 samples (Grid 2)

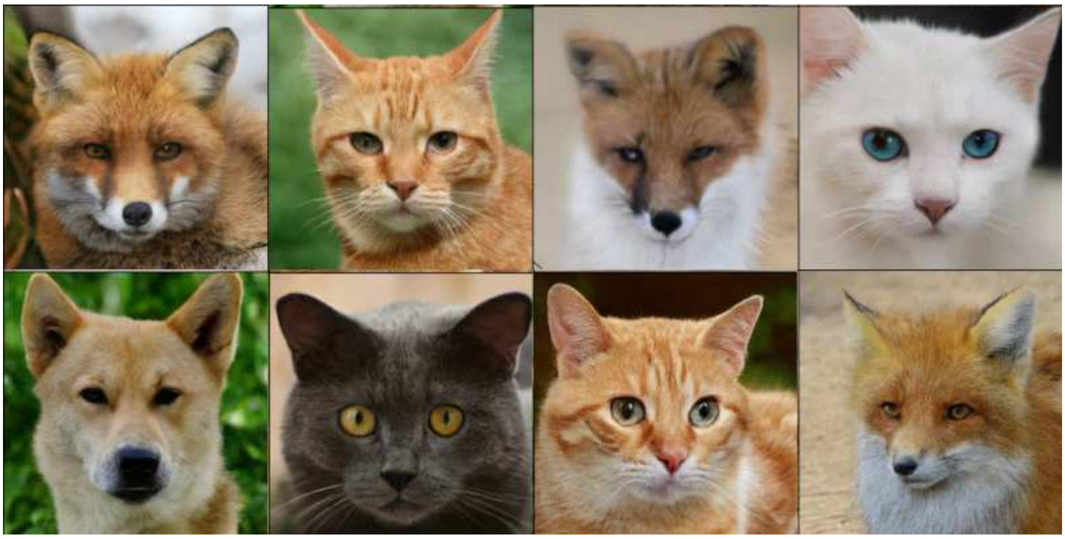
Figure 4.3: CelebA-HQ-256 unconditional samples demonstrating celebrity face generation



LSUN Church-256 samples (Grid 1)

LSUN Church-256 samples (Grid 2)

Figure 4.4: LSUN Church-256 unconditional samples showcasing architectural structure generation



AFHQ-512 samples (Grid 3)

Figure 4.5: AFHQ-512 unconditional samples from our custom-trained model AFHQ-512

Qualitative Analysis

Our analysis reveals several key observations about the quality and characteristics of generated samples across different datasets:

AFHQ-512 Analysis: Our custom-trained model shows remarkable performance across the three animal categories (cats, dogs, wild animals). The samples exhibit excellent inter-class diversity, with clear distinctions between domestic cats, various dog breeds, and wild animals. Intra-class coherence is well-maintained, with cats displaying typical feline features, dogs showing breed-specific characteristics, and wild animals maintaining their distinctive appearances. Notably, the 512×512 resolution appears to be a sweet spot for this dataset, producing cleaner results compared to the 1024×1024 FFHQ samples while maintaining sufficient detail for recognition.

FFHQ-1024 Analysis: The high-resolution human face samples demonstrate excellent detail preservation and realistic facial features. However, we observe occasional symmetry issues, particularly in eye alignment and ear positioning. This asymmetry becomes more pronounced at higher resolutions, suggesting that the model sometimes struggles with global coherence constraints during the lengthy sampling process. Despite these limitations, the overall quality remains impressive with natural skin textures, realistic lighting, and diverse demographic representation.

CelebA-HQ-256 Analysis: The celebrity face samples demonstrate consistent quality with good facial structure preservation. The lower resolution (256×256) reduces symmetry issues observed in FFHQ-1024, suggesting that the model performs more reliably at moderate resolutions. The samples show diverse facial expressions, ages, and ethnicities, indicating good coverage of the training distribution without obvious mode collapse.

LSUN Church-256 Analysis: The architectural samples showcase the model's capability to generate structured content beyond faces. Church samples exhibit realistic architectural elements including spires, windows, and stonework. The geometric consistency is generally good, though occasional artifacts appear in complex architectural details. The model successfully captures various architectural styles and lighting conditions present in the training data.

Identified Limitations

Several consistent issues emerge across our generated samples:

High-Resolution Symmetry Issues: FFHQ-1024 samples frequently exhibit facial asymmetry, particularly in eye positioning and ear alignment. This suggests that maintaining global coherence becomes increasingly challenging as resolution increases, likely due to the independence of pixel-level score predictions.

Generation Artifacts: AFHQ-512 samples occasionally show slight artifacts, particularly in fur texture transitions and background elements. However, these artifacts are relatively minor and don't significantly impact the overall quality. Interestingly, despite being trained from scratch for only 520K iterations, the AFHQ-512 model appears to produce cleaner results than the pre-trained FFHQ-1024 model, suggesting that the intermediate resolution provides a better balance between detail and coherence.

Texture Consistency: Across all datasets, we observe occasional issues with texture coherence, particularly in areas requiring fine detail such as hair, fur, or architectural ornamentation. These manifest as localized blurring or inconsistent texture patterns.

Quality-Diversity Trade-off: While our models successfully avoid mode collapse and generate diverse samples, we notice that the highest-quality samples tend to be more conservative, staying closer to typical examples from the training data. This suggests an inherent trade-off between sample quality and diversity that is characteristic of score-based generative models.

These observations provide important insights into the capabilities and limitations of VE-SDE-based generation, informing both the interpretation of our results and directions for future improvements.

Additional unconditional samples are provided in Appendix C.1.

4.2 Quantitative Evaluation

Evaluation Metrics Introduction

Quantitative evaluation of generative models relies on established metrics that capture both sample quality and diversity. Two primary metrics dominate the evaluation of image generation models:

Fréchet Inception Distance (FID): The FID metric measures the distance between the distributions of real and generated images in the feature space of a pre-trained Inception network. Mathematically, it is computed as:

$$\mathrm{FID} = ||\mu_r - \mu_g||_2^2 + \mathrm{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2}) \tag{4.1}$$

where μ_r, Σ_r and μ_g, Σ_g are the mean and covariance of the real and generated data distributions respectively in the Inception feature space. Lower FID scores indicate better sample quality and diversity, with the metric being sensitive to both mode collapse and generation artifacts.

Inception Score (IS): The IS metric evaluates both the quality and diversity of generated images by measuring the mutual information between generated samples and their predicted labels:

$$\mathrm{IS} = \exp(\mathbb{E}_{x \sim p_g}[D_{KL}(p(y|x)||p(y))]) \tag{4.2}$$

where $p(y|x)$ is the conditional label distribution for a generated sample x, and $p(y)$ is the marginal label distribution. Higher IS scores indicate better quality and diversity, though the metric has known limitations including bias toward ImageNet-like images.

Importance for Generative Model Evaluation: These metrics provide essential quantitative benchmarks for comparing different generative approaches. FID captures the overall distributional similarity between real and generated data, while IS focuses on the clarity and diversity of generated samples. Together, they offer complementary perspectives on model performance that supplement qualitative visual assessment.

Resource Constraints and Evaluation Limitations

Despite the importance of these quantitative metrics, we were unable to compute FID and IS scores for our generated samples due to significant computational resource constraints. Both metrics require generating large quantities of samples (typically 10,000-50,000 samples) to obtain statistically reliable estimates. Given our limited computational budget and the expensive nature of score-based sampling (requiring 2000 discretization steps per sample), this evaluation was beyond our available resources.

As a result, our evaluation relies primarily on qualitative visual assessment and comparison with Yang Song et al.'s reported benchmark results. While this limitation prevents direct quantitative comparison with other methods, the visual quality of our generated samples provides strong evidence of successful model performance, particularly for our custom-trained AFHQ-512 model.

5 Downtown tasks enabled by SBGM

5.1 Image Inpainting

Theoretical Foundation

Intuitive Understanding

What happens when part of an image is missing or corrupted? Image inpainting addresses exactly this problem by intelligently filling in unknown or masked regions.

Consider a photograph of a cat where a rectangular region has been removed—the missing area could contain the cat's nose, whiskers, fur patterns, or background. The answer isn't unique, and any completion should look natural and seamlessly blend with surrounding content.

Score-based inpainting excels at capturing this uncertainty by generating multiple plausible completions that all appear realistic and contextually appropriate.

Theoretical Foundation

The task involves sampling from the conditional distribution $p(x(0)|\Omega(y))$, where $\Omega(y)$ represents the known regions of a partially observed image y.
The goal is to generate plausible completions for the missing regions while maintaining consistency with the observed parts.

The theoretical foundation relies on approximating the intractable score $\boldsymbol{\nabla_x \log p_t(x(t)|\Omega(y))}$. Yang Song et al. provide an elegant solution through the approximation:

$$\boxed{\nabla_x \log p_t(z(t)|\Omega(x(0)) = y) \approx \nabla_z \log p_t([z(t); \hat{\Omega}(x(t))]) \qquad (5.1)}$$

where $z(t)$ represents the unknown regions, $\hat{\Omega}(x(t))$ represents the known regions at time t, and $[z(t); \hat{\Omega}(x(t))]$ denotes the reconstruction of the complete image. This approximation allows us to use our unconditional score model directly without training auxiliary networks.
The complete theoretical derivation and approximation analysis for inpainting is provided in Appendix B.1.

Results and Analysis

We evaluated inpainting performance across all four datasets using various masking strategies including geometric shapes (squares, circles, rectangles), random patches, and creative patterns. The results demonstrate the versatility and effectiveness of the score-based approach for image completion.

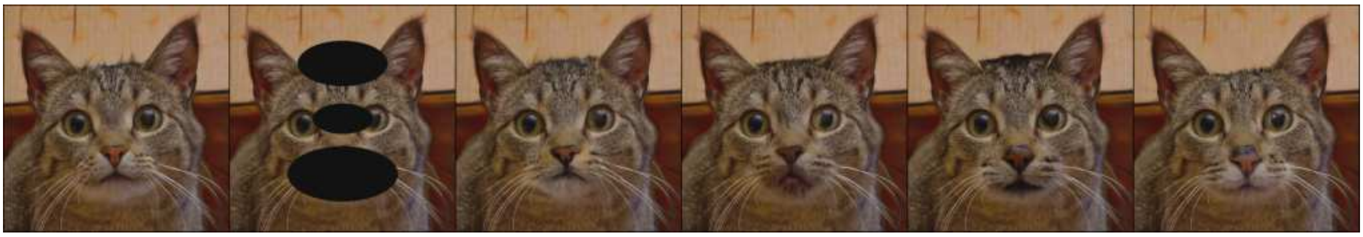


Figure 5.1: AFHQ-512 inpainting results demonstrating coherent animal face completion

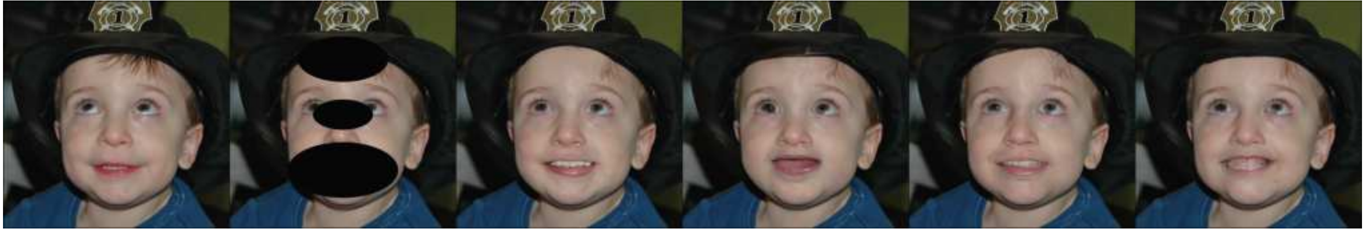


Figure 5.2: FFHQ-1024 inpainting results showing diverse completions for masked facial regions

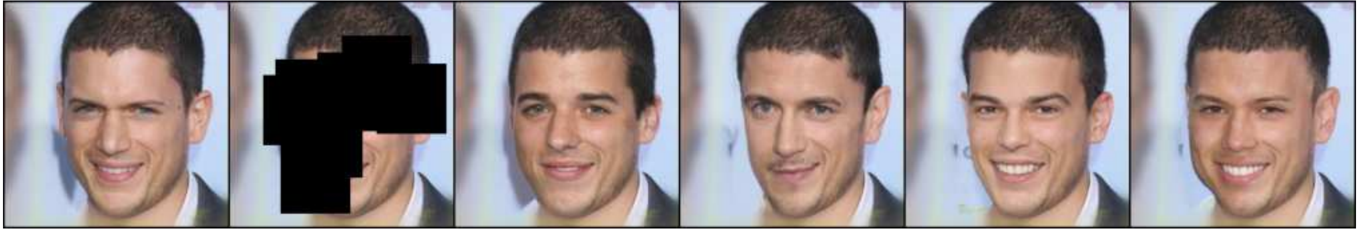


Figure 5.3: CelebA-HQ-256 inpainting results showing plausible facial feature reconstruction

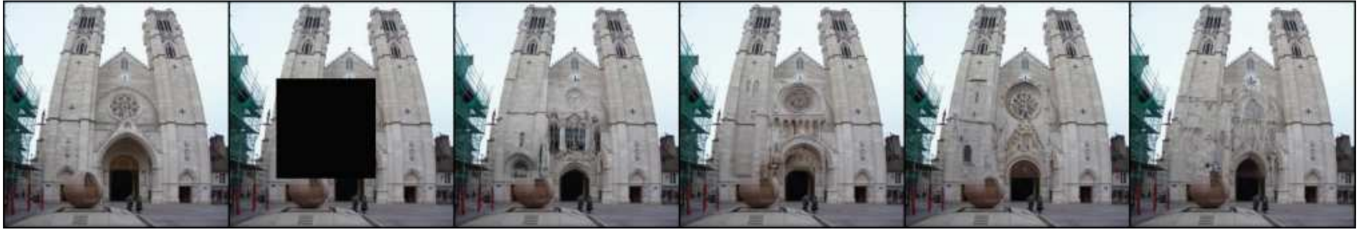


Figure 5.4: LSUN Church-256 inpainting results exhibiting architectural structure completion

Quality Assessment: The inpainting results demonstrate remarkable quality across all datasets. The reconstructed regions show excellent coherence with the surrounding context, maintaining consistent lighting, texture, and semantic content. The diversity of plausible completions highlights the model's ability to capture the inherent uncertainty in the inpainting task.

Dataset-Specific Analysis: Church architecture images prove easier to inpaint than human faces, likely due to the more structured and predictable nature of architectural elements. In contrast, facial inpainting requires more subtle understanding of human and animal facial structure and expressions, making it more challenging but still achieving impressive results.

Observed Limitations: While global coherence is well-maintained, some fine details are occasionally lost or slightly altered during the inpainting process. This is particularly noticeable in high-frequency textures and faces.

Extended inpainting results with various masking strategies are shown in Appendix C.2.

5.2 Colorization

Theoretical Approach

Intuitive Understanding

Colorization transforms grayscale images into full-color versions, but involves more complexity than simple inpainting.

Consider an old black-and-white photo of a person in a garden—while we know brightness values, we need to determine appropriate colors: what color should their shirt be? Is the grass green or autumn brown?
Unlike inpainting where regions are completely unknown, colorization provides partial information (luminance) that constrains but doesn't uniquely determine the result.

Theoretical Foundation

The challenge lies in color channel interdependence—we cannot treat RGB independently because they share the same brightness structure. Score-based colorization handles this by separating known grayscale information from unknown color information.

Yang Song et al. solve this elegantly using an orthogonal transformation that decouples grayscale information from color information. The process involves three steps:
(1) transforming RGB values to a space Yuv where grayscale and color information are separated,
(2) applying inpainting to the missing color channels, and
(3) transforming back to RGB space.

The principle of the Yuv color space is to represent colors using a luminance component Y, and 2 chrominance components (u,v) corresponding to the blue and red components in reduced chromaticity coordinates as presented by the figure 5.5 below.

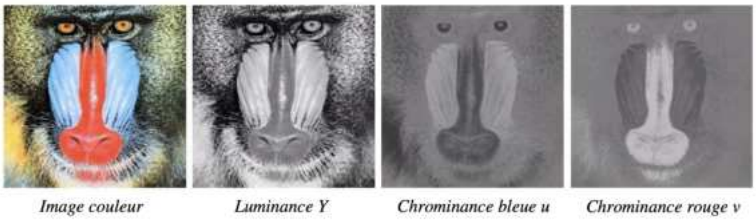


Figure 5.5: Components in Yuv espace. Source : [10]

The specific orthogonal matrix used for this transformation is:

$$M = \begin{pmatrix} 0.577 & -0.816 & 0 \\ 0.577 & 0.408 & 0.707 \\ 0.577 & 0.408 & -0.707 \end{pmatrix} \tag{5.2}$$

This transformation preserves the properties of the Wiener process since orthogonal transformations maintain Gaussian distributions, allowing seamless application of the SDE framework in the transformed space.

Complete colorization theory including the orthogonal transformation properties is detailed in Appendix B.2.

Results and Analysis

We evaluated colorization performance across all datasets, converting selected images to grayscale and then generating diverse colorizations using our score-based approach.

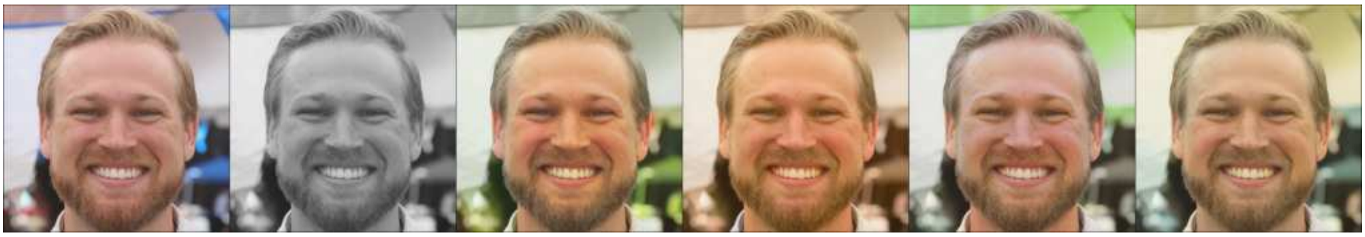


Figure 5.6: FFHQ-1024 colorization results showing diverse plausible color schemes for faces

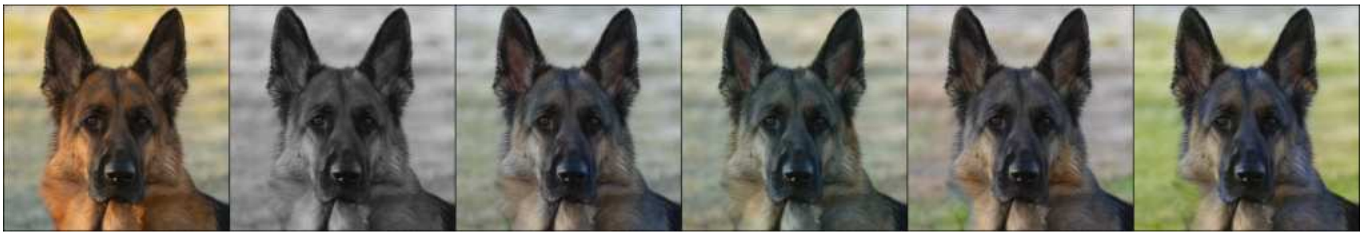


Figure 5.7: AFHQ-512 colorization results demonstrating natural animal coloring

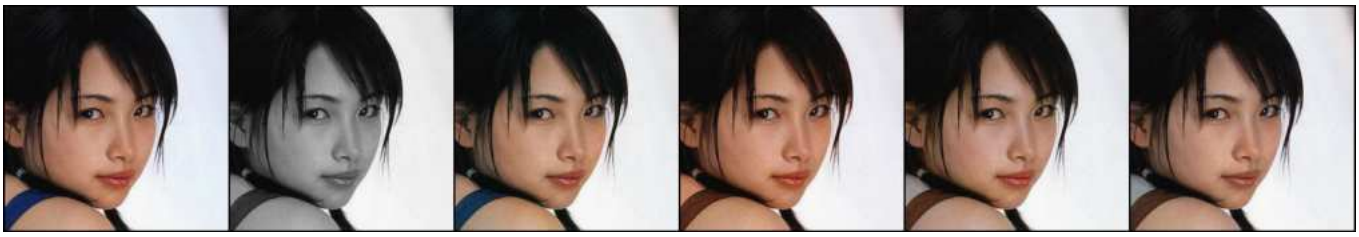


Figure 5.8: CelebA-HQ-256 colorization results exhibiting realistic facial tones

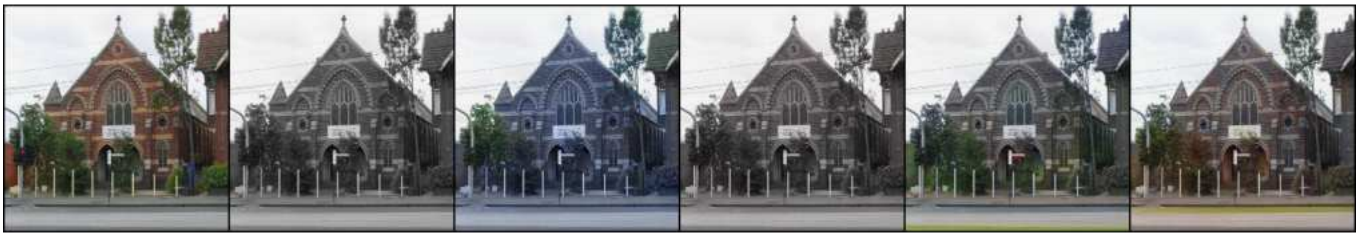


Figure 5.9: LSUN Church-256 colorization results showing architectural color schemes

Quality Assessment: The colorization results demonstrate impressive realism with natural color choices that respect the underlying grayscale structure. The generated colors appear plausible and consistent with the semantic content of the images. The diversity of possible colorizations highlights the model's ability to capture the inherent ambiguity in the colorization task.

Limitations: Some semantic inconsistencies occasionally appear, such as grass receiving blue tones or sky areas showing unnatural colors. Additionally, color saturation is sometimes not perfectly faithful to realistic expectations, though this varies by dataset and individual samples.

Extended colorization results across different scenarios are presented in Appendix C.3.

5.3 Class-Conditional Generation

Score Decomposition Theory

Intuitive Understanding

Rather than generating random samples, class-conditional generation produces images from specific desired categories. An unconditional model might randomly produce cats, dogs, or wild animals with equal probability, but what if we specifically want only cat images ?

The solution combines unconditional score functions with classifier gradients that recognize the desired class even in noisy, partially generated images, effectively ”nudging” the sampling process toward the intended category throughout generation.

Theoretical Foundation

Class-conditional generation represents the core contribution of this work, requiring the decomposition of conditional scores to enable controllable sampling. The theoretical foundation relies on the fundamental relationship: $\boxed{\nabla_x \log p_t(x|y) = \nabla_x \log p_t(x) + \nabla_x \log p_t(y|x)}$ (5.3)

This decomposition reveals that the conditional score function can be expressed as the sum of the unconditional score (learned by our base model) and the gradient of the classifier's log-likelihood. This insight enables conditional generation without retraining the base generative model.

The conditional reverse-time SDE becomes:

$$dx = [f(x,t) - g(t)^2(\nabla_x \log p_t(x) + \nabla_x \log p_t(y|x))]dt + g(t)d\bar{w} \tag{5.4}$$

where the additional term $\boldsymbol{\nabla_x \log p_t(y|x)}$ provides the class-specific guidance.

Classifier Training and Architecture

We fine-tuned a ResNet-50 classifier specifically adapted for AFHQ-512 classification with an embedding size of 128 and a pre-trained backbone. The classifier was trained on temporally corrupted data pairs $(x(t), y)$ where $x(t) \sim p_{0t}(x(t)|x(0))$ and y represents the specific class desired, enabling it to classify images across different noise levels.

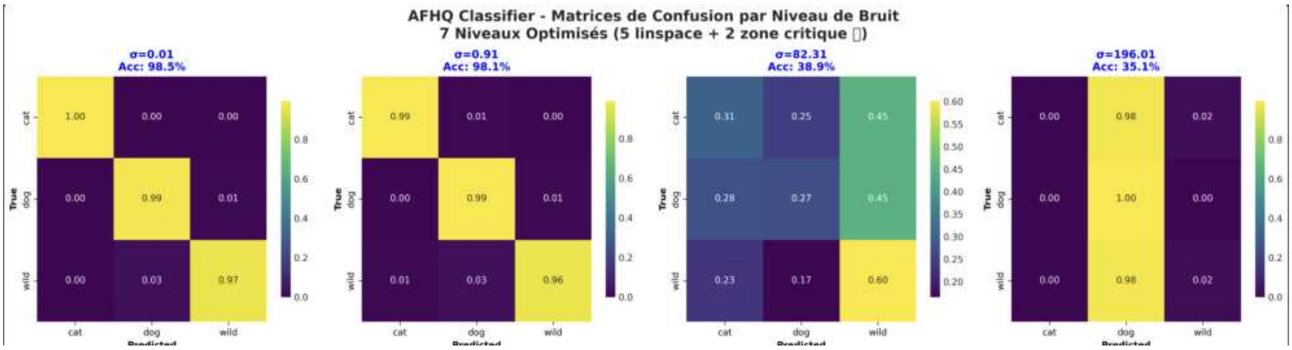
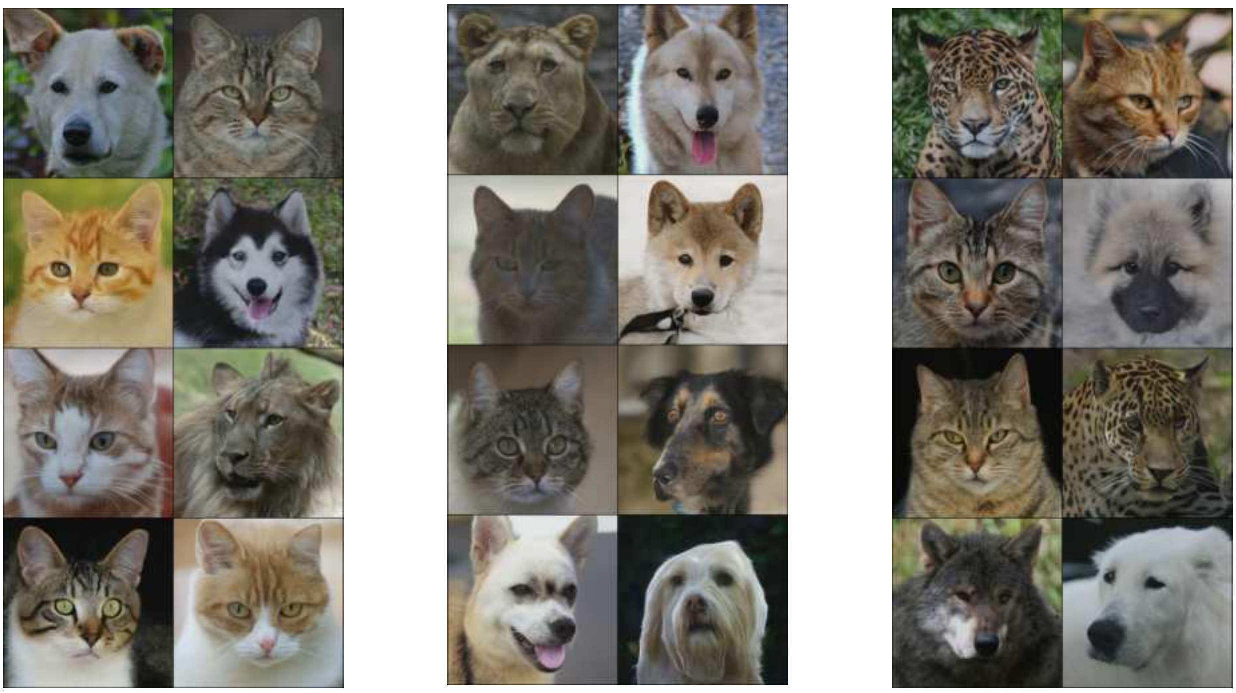


Figure 5.10: Confusion matrix across noise levels

The confusion matrix demonstrates the expected degradation at higher noise levels, which is natural given the increasing difficulty of classification as images become more corrupted. This observation proved crucial for developing our adaptive guidance strategies.

Initial Guidance Results

Our initial implementation used the standard theoretical formulation without any modifications. The results were not satisfying, showing weak class control that was nearly identical to unconditional generation.



Cat class - initial results

Dog class - initial results

Wild class - initial results

Figure 5.11: Initial guidance results showing poor class control

Problem Analysis: The poor performance stemmed from the classifier being ineffective across the full noise spectrum σ, with gradients too weak to meaningfully impact the sampling process. Analysis revealed that the L2 norm ratio between the score function and classifier gradients was approximately 30,000:1, explaining why the classifier had virtually no effect on sample generation.

Adaptive Strategies Exploration

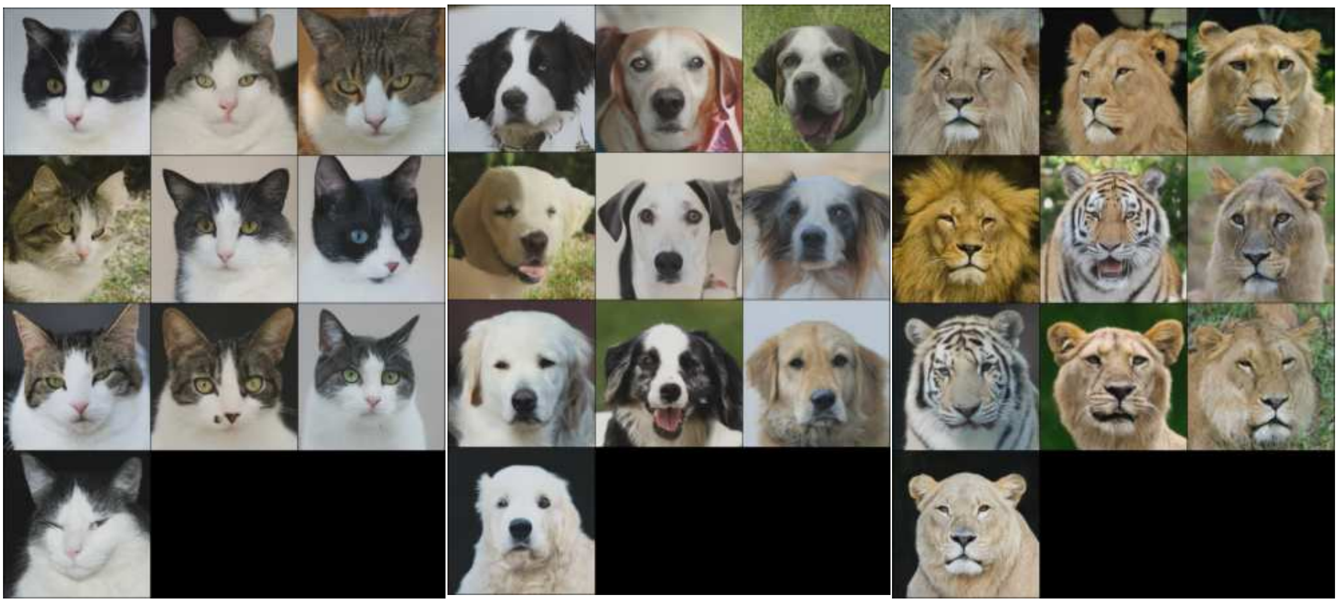
Based on our analysis of the initial results, we developed three adaptive strategies to improve classifier guidance effectiveness. All subsequent experiments used fixed seeds to ensure fair comparison.

Strategy 1: Amplified Guidance Scale

Approach: We introduced a guidance scale parameter λ set to 500 after calibration experiments. This amplifies the classifier gradient to balance the ratio between score and classifier contributions.

$$\nabla_x \log p_t(x|y) = \nabla_x \log p_t(x) + \boldsymbol{\lambda} \times \nabla_x \log p_t(y|x) \tag{5.5}$$

Hypothesis: Increasing the guidance ratio would amplify the classifier's effect, enabling meaningful class control during sampling.



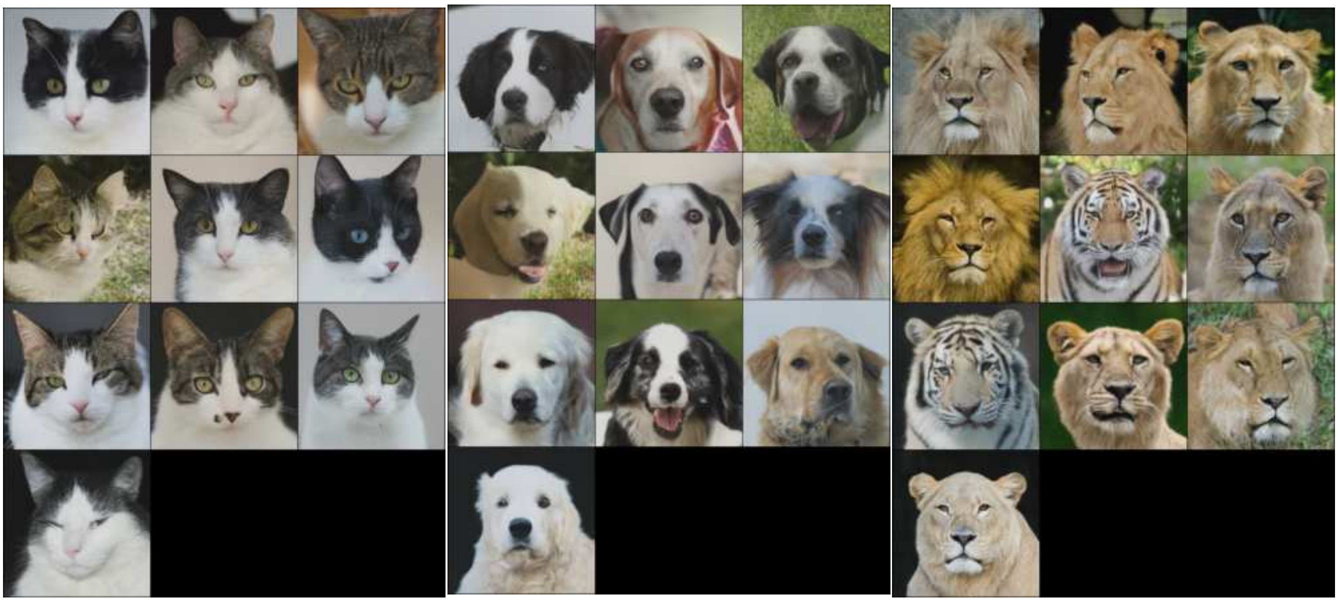
Cat class - Method 1 Dog class - Method 1 Wild class - Method 1

Figure 5.12: Strategy 1 results showing improved class-specific generation

Results: This approach showed significant improvement, successfully generating animals from the desired class only, which was exactly our objective. However, we observed slight artifacts in the generated images, motivating exploration of alternative approaches as showned by the figure 5.12.

Strategy 2: Adaptive Guidance Scale

Approach: We implemented a linearly decreasing guidance scale λ from 500 to 400 based on the observation that the classifier importance (L2 norm) diminishes at low noise levels while the factor λ appears to well-guide the generation process in the this regions : $\lambda \in \{400,\ 500\}$.



Cat class - Method 2 Dog class - Method 2 Wild class - Method 2

Figure 5.13: Strategy 2 results showing slight improvement in detail quality

Hypothesis: Reducing guidance progressively accounts for the classifier's natural attenuation at low noise levels. As images become more visible, the score model naturally guides toward the desired class region while the classifier provides scores close to zero (due to high confidence leading to gradients approaching zero through the softmax function).

Results: This approach showed slight improvement in detail quality compared to Strategy 1 as demonstrated by the figure 5.13. However, the improvements were modest, leading us to explore a more targeted approach.

Strategy 3: Sigma Truncation

Approach: We restricted classifier usage to its effective zones by implementing sigma truncation.
Based on our analysis of classifier ineffectiveness in high-noise regions while being unuseful in low-noise ones due to vanishing contribution, we limited classifier application to differents regions configurations such as $\sigma \in \{1,\ 374\}$, $\{1,\ 100\}$ or $\{1,\ 50\}$ for each class and run many experiments.

Implementation : The figure below 5.14 details the strategy :

Strategy 3: Double Truncation Implementation

Variables:

- σ_{current}: actual noise level during sampling
- σ_max_classifier: maximum noise level σ where classifier is effective (here it's ≈ 50)
- σ_limit_pred_min, σ_limit_pred_max : guidance zone boundaries where guidance has effect on sampling process such as $\sigma \in \{1,\ 100\}$;

Two-Level Control:

1) Guidance Zone Check:

if $\sigma_{\text{current}} \leq \sigma$_limit_pred_max or $\sigma_{\text{current}} \geq \sigma$_limit_pred_max:
return 0 (Outside guidance zone)

2) Classifier Truncation (inside guidance zone):

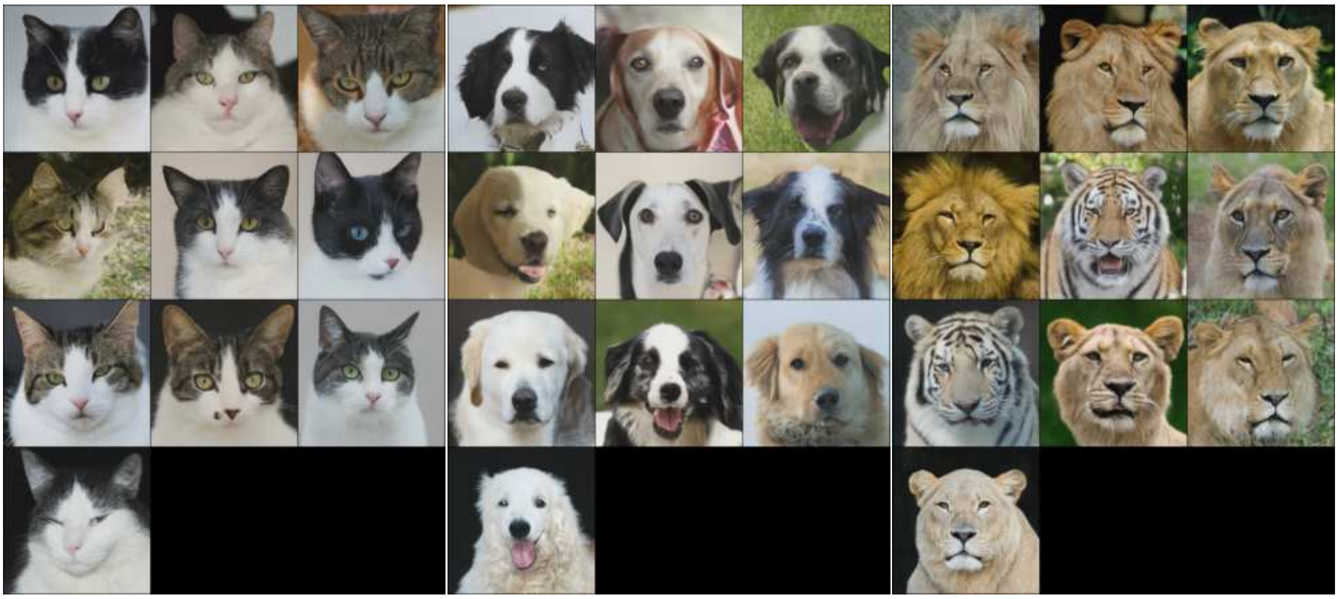
$\sigma_{\text{clamped}} = \min(\sigma_{\text{current}}, \sigma$_max_classifier)
∇_xclassifier = compute_gradient(x, $\sigma_{clamped}$)
return ∇_xclassifier * *guidance_scale*$_\lambda$

Key Insight: Even when current_sigma = 100, classifier sees $\sigma_{\text{clamped}} = 50$, operating in its effective range

Figure 5.14: Double truncation: guidance zone + classifier effectiveness limit

Results: This method, see figure 5.15, achieved similar results to the previous improvement strategies, demonstrating that our assumption about classifier ineffectiveness in certain noise zones was well-founded.

The truncation approach, on a side, provides a principled way to optimize classifier usage but don't significantly improve sample quality on a other side.



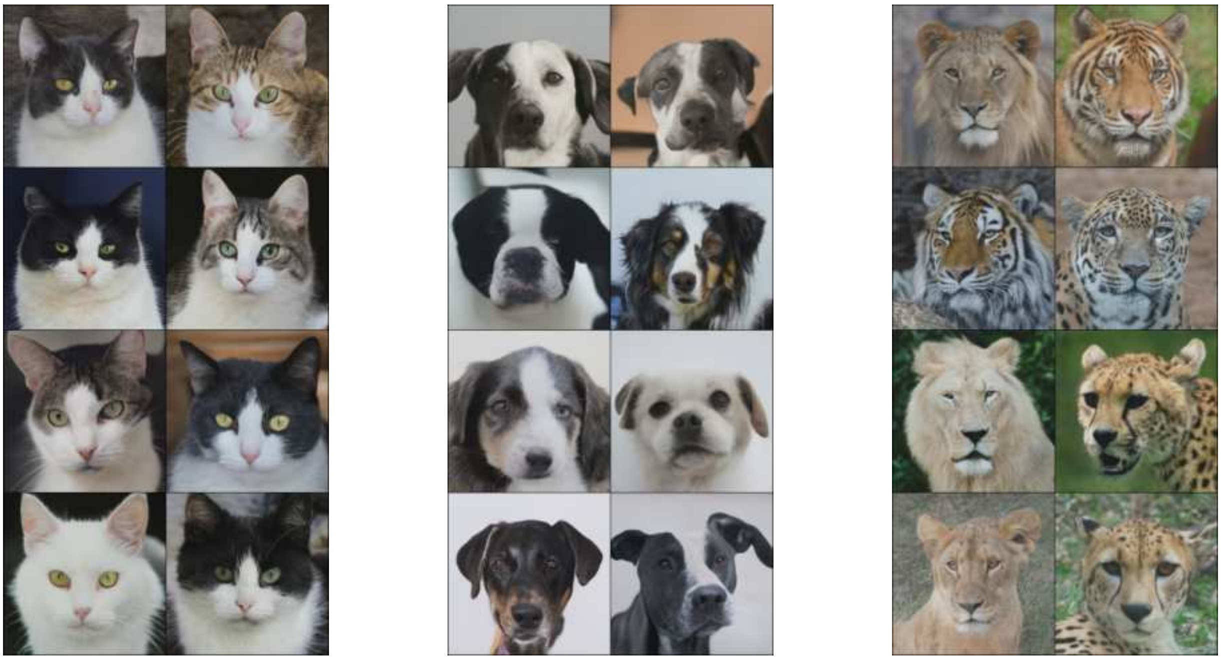
Cat class - Method 3 Dog class - Method 3 Wild class - Method 3

Figure 5.15: Strategy 3 results demonstrating effectiveness of targeted classifier usage

Summary

All three adaptive strategies significantly outperformed the baseline theoretical formulation :

• Strategy 1 (amplified guidance scale) proved most practical and effective, providing strong class control with reasonable computational overhead.
• Strategy 2 (adaptive scaling) offered marginal improvements in detail quality, while
• Strategy 3 (sigma truncation) validated our theoretical understanding of classifier effectiveness zones.
The final comparison uses Strategy 1 without fixed seeds, extended results are provided in Appendix C.4.



Cat class - Final sampling Dog class - Final sampling Wild class - Final sampling

Figure 5.16: Final sampling using Strategy 1 without fixed seeds, showing natural variation

6 Limitations and Failure Modes

6.1 Technical Limitations Observed

Our implementation and experimentation revealed several technical limitations inherent to score-based generative models, particularly when scaling to high-resolution generation and complex conditional sampling tasks.

High-Resolution Coherence Challenges: FFHQ-1024 samples frequently exhibited facial asymmetry, particularly in eye alignment and ear positioning. This asymmetry becomes more pronounced at higher resolutions, suggesting that maintaining global coherence constraints during lengthy sampling processes becomes increasingly difficult.

Dataset-Specific Artifacts: Our custom-trained AFHQ-512 model showed repetitive pose patterns, primarily due to the centered face composition in the original dataset combined with our sampling strategy. Despite augmentation improvements, this limitation persisted throughout training. Interestingly, our AFHQ-512 model produced notably cleaner results than the pre-trained FFHQ-1024 model despite the resolution difference, suggesting that intermediate resolutions may provide better quality-coherence trade-offs.

Complex Geometry Generation Difficulties: LSUN Church samples revealed particular challenges with complex architectural geometries. The model struggled with intricate structural details, fine architectural elements, and maintaining consistent perspective across complex building facades. This difficulty appears more pronounced with geometrically complex subjects compared to organic forms like faces and animals.

Texture and Detail Inconsistencies: Across all datasets, we observed localized blurring and inconsistent texture transitions, particularly in high-frequency details such as hair, fur, and architectural ornamentation. These artifacts manifest as abrupt texture changes or loss of fine detail during the reverse diffusion process.

6.2 Computational and Resource Constraints

The computational demands of score-based generative models present significant practical limitations that extend beyond technical performance to economic and environmental considerations.

Economic and Environmental Impact: Training our AFHQ-512 model from scratch required approximately €300 in Lambda Cloud computing costs, representing a substantial investment from personal funds. This cost reflects the broader environmental concern associated with large-scale neural network training. As these methods become increasingly democratized, the cumulative environmental impact of widespread adoption raises important sustainability questions that the field must address through more efficient training paradigms.

Sampling Performance Limitations: Generation times proved prohibitive for practical applications, requiring approximately 3 minutes per image on a G200 GPU and over 1 hour on consumer hardware (Mac). This performance bottleneck severely limits real-time applications and makes iterative experimentation computationally expensive.

Evaluation Resource Constraints: Computing standard quantitative metrics (FID, IS) proved impossible due to the requirement of generating 10,000-50,000 samples. This evaluation limitation forces reliance on qualitative assessment and limits direct comparison with other methods, highlighting a significant barrier for comprehensive model evaluation in resource-constrained environments.

6.3 Methodological Limitations

Our adaptive classifier guidance strategies, while effective, revealed several methodological constraints that limit their generalizability and practical applicability.

Guidance Strategy Constraints: The three classifier guidance approaches we developed amplified guidance scale, adaptive linear scaling, and sigma truncation—each required dataset-specific hyperparameter tuning. No universal method emerged for automatically determining optimal guidance zones, necessitating manual calibration for each new dataset or configuration. This limitation significantly reduces the practical utility of our methods for general applications.

Classifier Effectiveness Dependencies: Our analysis revealed that classifier guidance effectiveness varies dramatically across noise levels, being ineffective at high noise levels while providing diminishing returns at very low noise levels. However, determining these effectiveness boundaries requires empirical testing for each specific classifier-dataset combination, limiting the transferability of our findings.

Downstream Task Limitations: While our inpainting implementation maintained global coherence, fine details were occasionally lost or altered during the completion process.
Colorization tasks showed occasional semantic inconsistencies, such as unnatural color assignments to objects (grass appearing blue, sky showing unrealistic hues), and saturation levels that deviated from realistic expectations. These limitations suggest that while the score-based framework provides powerful tools for conditional generation, careful task-specific considerations remain necessary.

7 Conclusion and Perspectives

This internship successfully implemented and evaluated score-based generative models using stochastic differential equations, providing both theoretical insights and practical contributions. Our primary technical achievement was training an AFHQ-512 model from scratch, demonstrating VE-SDE feasibility with limited computational resources. Despite terminating at 520,000 iterations due to constraints, the model achieved satisfactory quality and enabled our classifier guidance experiments. Multi-resolution evaluation across FFHQ-1024, AFHQ-512, CelebA-HQ-256, and LSUN Church-256 provided insights into scaling behavior and domain-specific challenges.

We developed three distinct classifier guidance strategies addressing classifier ineffectiveness across noise spectrums. The amplified guidance scale approach ($\lambda = 500$) proved most practical, successfully enabling class-conditional generation despite slight artifacts. Our sigma truncation validated theoretical assumptions about classifier effectiveness zones, while adaptive scaling provided marginal improvements. These contributions demonstrate that adaptive strategies can partially overcome naive classifier guidance limitations. The score decomposition framework proved remarkably practical for downstream tasks, enabling inpainting, colorization, and conditional generation using a single unconditional model.

Our work revealed fundamental limitations pointing toward important research directions. The computational cost remains a significant barrier—our €300 training cost for a single 512×512 model highlights sustainability challenges. The hyperparameter sensitivity of guidance strategies requiring dataset-specific tuning represents a significant usability limitation. High-resolution generation faces persistent coherence challenges, particularly for symmetric objects, suggesting current approaches may benefit from explicit geometric constraints or architectural innovations.

The most pressing needs include automatic guidance zone detection algorithms and universal guidance strategies that adapt automatically to different datasets. The environmental impact demands attention, requiring computational efficiency improvements through better initialization, architectures, or training paradigms. As exemplified by Stable Diffusion and classifier-free guidance emergence, the field moves toward more accessible approaches. The evolution from our classifier-based methods to modern CFG demonstrates trends toward reduced hyperparameter sensitivity and improved usability.

Beyond technical achievements, this internship provided invaluable exposure to cutting-edge research. Working at University of Torino under Professor Elena Issoglio's supervision offered unique insights into stochastic analysis and machine learning intersection. The international environment fostered deeper understanding of theoretical-to-practical translation. Challenges from debugging sampling algorithms to managing computational resources provided essential real-world research experience.

In conclusion, while our methods represent an earlier stage in this rapidly evolving field, fundamental insights about adaptive guidance strategies, computational trade-offs, and implementation challenges remain relevant. The continued evolution toward more efficient, accessible, and sustainable approaches suggests promising futures for score-based generative models, with applications extending beyond image generation to diverse fields requiring high-quality data synthesis.

Bibliography

[1] Brian D.O. Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

[2] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in Neural Information Processing Systems*, 31, 2018.

[3] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Animal faces-hq dataset (afhq), 2020. High-quality animal face dataset for few-shot image generation.

[4] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

[5] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[6] Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24):695–709, 2005.

[7] Alexia Jolicoeur-Martineau, Rémi Piché-Taillefer, Rémi Tachet des Combes, and Ioannis Mitliagkas. Gotta go fast when generating data with score-based models. *arXiv preprint arXiv:2105.14080*, 2021.

[8] Tero Karras, Samuli Laine, and Timo Aila. Flickr-faces-hq dataset (ffhq), 2019. High-quality face dataset at 1024×1024 resolution.

[9] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Celeba-hq dataset, 2015. High-quality version of CelebA dataset at 256×256 resolution.

[10] Antoine Manzanera. Apprentissage machine - couleur, 2024. Course material on color spaces and YUV transformation.

[11] 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.

[12] Yang Song. Score-based generative models blog, 2021.

[13] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019.

[14] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *International Conference on Learning Representations*, 2021.

[15] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

Glossary

This glossary presents the technical terms and abbreviations used in this report.

Abbreviations

Abbreviations	**Meaning**
CFG	Classifier-Free Guidance
DDPM	Denoising Diffusion Probabilistic Models
EMA	Exponential Moving Average
FID	Fréchet Inception Distance
IS	Inception Score
MCMC	Markov Chain Monte Carlo
NCSN	Noise Conditional Score Network
ODE	Ordinary Differential Equations
PC	Predictor-Corrector
SDE	Stochastic Differential Equations
SGBM	Score-Based Generative Models
SMLD	Score Matching with Langevin Dynamics
sub-VP	sub-Variance Preserving
VE	Variance Exploding
VP	Variance Preserving

Table 7.1: Abbreviations used in this report

Term	**Definition**
Classifier guidance	Method using external classifier to guide generation
Colorization	Process of adding color to grayscale images
Denoising score matching	Training method for score functions using noisy data
Diffusion models	Generative models based on noise injection/removal
Forward SDE	Stochastic process that adds noise to data
Inpainting	Task of filling missing regions in images
Langevin dynamics	MCMC method using gradient information
Markov Chain Monte Carlo	Sampling method using Markov chains for probability distributions
Predictor-corrector	Sampling method combining SDE solver with MCMC
Reverse-time SDE	Stochastic process that removes noise from data
Score function	Gradient of log probability density
Score matching	Method to estimate score functions
Unconditional generation	Generating samples without specific constraints

Table 7.2: Technical terms and definitions

Datasets

Acronym	**Full Name**
AFHQ	Animal Faces-HQ
CelebA-HQ	Celebrity Faces Attributes - High Quality
FFHQ	Flickr-Faces-HQ
LSUN	Large-scale Scene Understanding

Table 7.3: Datasets used in this report

Appendices

A. Mathematical Derivations

This appendix provides complete mathematical derivations for the key theoretical results underlying score-based generative models using SDEs. The derivations are based on the foundational work of Yang Song et al. (2021) and include the transition from discrete noise perturbations to continuous stochastic processes, closed-form expressions for perturbation kernels, and the theoretical foundations for the denoising score matching objective.

A.1 From Discrete SMLD to Continuous VE-SDE

The transition from the discrete Score Matching with Langevin Dynamics (SMLD) to the continuous Variance Exploding SDE represents a fundamental insight that unifies discrete and continuous approaches to score-based generative modeling.

Starting Point: Discrete SMLD Markov Chain

In SMLD, we consider a sequence of noise scales $0 < \sigma_{\min} = \sigma_1 < \sigma_2 < \cdots < \sigma_N = \sigma_{\max}$ and construct a Markov chain where each step adds Gaussian noise. The discrete process is defined by:

$$x_i = x_{i-1} + \sqrt{\sigma_i^2 - \sigma_{i-1}^2}\, z_{i-1}, \quad i = 1, 2, \ldots, N \tag{7.1}$$

where $z_{i-1} \sim \mathcal{N}(0, I)$ and we introduce $\sigma_0 = 0$ to simplify notation.

Continuous Limit Construction

To obtain the continuous limit as $N \to \infty$, we make the following substitutions: - Let $t \in [0, 1]$ be the continuous time variable
- Define $x(\frac{i}{N}) = x_i$, $\sigma(\frac{i}{N}) = \sigma_i$, and $z(\frac{i}{N}) = z_i$
- Set $\Delta t = \frac{1}{N}$ as the discrete time step

The discrete update rule becomes:

$$x(t + \Delta t) = x(t) + \sqrt{\sigma^2(t + \Delta t) - \sigma^2(t)}\, z(t) \tag{7.2}$$

Derivation of the Differential Form

Using the mean value theorem for small Δt:

$$\sigma^2(t + \Delta t) - \sigma^2(t) = \frac{d[\sigma^2(t)]}{dt} \Delta t + o(\Delta t) \tag{7.3}$$

$$\approx \frac{d[\sigma^2(t)]}{dt} \Delta t \quad \text{as } \Delta t \to 0 \tag{7.4}$$

Substituting this approximation:

$$x(t + \Delta t) = x(t) + \sqrt{\frac{d[\sigma^2(t)]}{dt} \Delta t}\, z(t) \tag{7.5}$$

Convergence to the VE-SDE

In the limit $\Delta t \to 0$, the discrete process $\{x_i\}_{i=1}^N$ converges to the continuous stochastic process $\{x(t)\}_{t=0}^1$ governed by the Variance Exploding SDE:

$$dx = \sqrt{\frac{d[\sigma^2(t)]}{dt}}\, dw \tag{7.6}$$

where $w(t)$ is a standard Wiener process.

Noise Schedule and Perturbation Kernel

For the geometric noise schedule $\sigma_i = \sigma_{\min}\left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)^{\frac{i-1}{N-1}}$, the continuous function becomes:

$$\sigma(t) = \sigma_{\min}\left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)^t, \quad t \in [0,1] \tag{7.7}$$

This yields the specific VE-SDE:

$$dx = \sigma(t)\sqrt{2\log\frac{\sigma_{\max}}{\sigma_{\min}}}\,dw \tag{7.8}$$

The perturbation kernel is:

$$p_{0t}(x(t)|x(0)) = \mathcal{N}\left(x(t); x(0), \sigma^2(t)I\right) \tag{7.9}$$

Note that the VE-SDE is not differentiable at $t = 0$ due to the discontinuity in $\sigma(t)$, requiring computation in the range $t \in [\varepsilon, 1]$ for small $\varepsilon > 0$.

A.2 From Discrete DDPM to Continuous VP-SDE

The derivation of the Variance Preserving SDE from Denoising Diffusion Probabilistic Models (DDPM) follows a similar limiting procedure but with a different discrete structure.

Starting Point: Discrete DDPM Markov Chain

In DDPM, the discrete Markov chain is defined by:

$$x_i = \sqrt{1-\beta_i}x_{i-1} + \sqrt{\beta_i}z_{i-1}, \quad i = 1, 2, \ldots, N \tag{7.10}$$

where $\{\beta_i\}_{i=1}^N$ is a variance schedule with $0 < \beta_i < 1$.

Auxiliary Scaling for Continuous Limit

To obtain the continuous limit, we introduce auxiliary noise scales $\{\tilde{\beta}_i = N\beta_i\}_{i=1}^N$ and rewrite the Markov chain as:

$$x_i = \sqrt{1-\frac{\tilde{\beta}_i}{N}}x_{i-1} + \sqrt{\frac{\tilde{\beta}_i}{N}}z_{i-1} \tag{7.11}$$

Continuous Limit Construction

Following the same procedure as for VE-SDE, we let $N \to \infty$ and define:
- $\beta(\frac{i}{N}) = \tilde{\beta}_i$
- $x(\frac{i}{N}) = x_i$
- $z(\frac{i}{N}) = z_i$
- $\Delta t = \frac{1}{N}$

The discrete update becomes:

$$x(t+\Delta t) = \sqrt{1-\beta(t+\Delta t)\Delta t}x(t) + \sqrt{\beta(t+\Delta t)\Delta t}z(t) \tag{7.12}$$

Taylor Expansion and Approximation

For small Δt, we use the approximation $\sqrt{1-x} \approx 1 - \frac{x}{2}$ for small x:

$$x(t+\Delta t) \approx x(t) - \frac{1}{2}\beta(t+\Delta t)\Delta t\,x(t) + \sqrt{\beta(t+\Delta t)\Delta t}z(t) \tag{7.13}$$

$$\approx x(t) - \frac{1}{2}\beta(t)\Delta t\,x(t) + \sqrt{\beta(t)\Delta t}z(t) \tag{7.14}$$

Convergence to the VP-SDE

In the limit $\Delta t \to 0$, this converges to the Variance Preserving SDE:

$$dx = -\frac{1}{2}\beta(t)x\,dt + \sqrt{\beta(t)}\,dw \tag{7.15}$$

For the typical linear schedule $\beta_i = \beta_{\min} + \frac{i-1}{N-1}(\beta_{\max} - \beta_{\min})$, we obtain:

$$\beta(t) = \beta_{\min} + t(\beta_{\max} - \beta_{\min}) \tag{7.16}$$

A.3 Mean and Variance Calculations for Three SDEs

This section provides complete calculations for the mean and variance of the three SDE families, which are essential for computing the perturbation kernels and score functions.

General Solution Method for Affine SDEs

All three SDEs (VE, VP, sub-VP) have the general form:

$$dx = f(x,t)dt + g(t)dw \tag{7.17}$$

where $f(x,t)$ is at most linear in x and $g(t)$ is deterministic. For such SDEs, the solution is Gaussian and we can compute the mean and variance analytically.

VE-SDE Analysis

For the VE-SDE: $dx = \sqrt{\frac{d[\sigma^2(t)]}{dt}}dw$

Mean: Since there is no drift term, the mean remains constant:

$$\mathbb{E}[x(t)] = x(0) \tag{7.18}$$

Variance: The variance evolves according to:

$$\text{Var}[x(t)] = \text{Var}[x(0)] + \int_0^t \frac{d[\sigma^2(s)]}{ds}ds = \text{Var}[x(0)] + \sigma^2(t)I \tag{7.19}$$

For $x(0) \sim p_{\text{data}}$ and assuming $\text{Var}[x(0)] = 0$ (delta function approximation):

$$\text{Var}[x(t)] = \sigma^2(t)I \tag{7.20}$$

VP-SDE Analysis

For the VP-SDE: $dx = -\frac{1}{2}\beta(t)xdt + \sqrt{\beta(t)}dw$

Mean: The mean evolves according to the ODE:

$$\frac{d\mathbb{E}[x(t)]}{dt} = -\frac{1}{2}\beta(t)\mathbb{E}[x(t)] \tag{7.21}$$

Solving this ODE:

$$\mathbb{E}[x(t)] = x(0)\exp\left(-\frac{1}{2}\int_0^t \beta(s)ds\right) \tag{7.22}$$

Variance: For the variance, we use the fact that for affine SDEs, the variance satisfies:

$$\frac{d\text{Var}[x(t)]}{dt} = -\beta(t)\text{Var}[x(t)] + \beta(t)I \tag{7.23}$$

Solving this ODE with initial condition $\text{Var}[x(0)] = 0$:

$$\text{Var}[x(t)] = \left[1 - \exp\left(-\int_0^t \beta(s)ds\right)\right]I \tag{7.24}$$

sub-VP SDE Analysis

For the sub-VP SDE: $dx = -\frac{1}{2}\beta(t)xdt + \sqrt{\beta(t)(1 - e^{-2\int_0^t \beta(s)ds})}dw$

Mean: The mean evolution is identical to VP-SDE:

$$\mathbb{E}[x(t)] = x(0)\exp\left(-\frac{1}{2}\int_0^t \beta(s)ds\right) \tag{7.25}$$

Variance: The variance equation becomes:

$$\frac{d\text{Var}[x(t)]}{dt} = -\beta(t)\text{Var}[x(t)] + \beta(t)(1 - e^{-2\int_0^t \beta(s)ds}) \tag{7.26}$$

Solving this yields:

$$\text{Var}[x(t)] = \left[1 - \exp\left(-\int_0^t \beta(s)ds\right)\right]^2 I \tag{7.27}$$

A.4 Anderson 1982: Existence and Uniqueness Conditions

This section presents the theoretical foundations for the existence and uniqueness of solutions to reverse-time SDEs, based on the seminal work of Anderson (1982).

Forward SDE Setup

Consider a forward SDE of the form:

$$dx = f(x,t)dt + g(t)dw \tag{7.28}$$

where $f : \mathbb{R}^d \times [0,T] \to \mathbb{R}^d$ is the drift coefficient and $g : [0,T] \to \mathbb{R}$ is the diffusion coefficient.

Reverse-Time SDE Formulation

Anderson (1982) proved that under appropriate regularity conditions, the reverse-time process $\{x(T-t)\}_{t\in[0,T]}$ satisfies the SDE:

$$dx = [f(x,t) - g(t)^2\nabla_x \log p_t(x)]dt + g(t)d\bar{w} \tag{7.29}$$

where $\bar{w}$ is a Wiener process when time flows backward and $p_t(x)$ is the marginal density of $x(t)$.

Existence and Uniqueness Conditions

For the existence and uniqueness of solutions, we require:

Lipschitz Continuity: There exists a constant $L > 0$ such that for all $x, y \in \mathbb{R}^d$ and $t \in [0,T]$:

$$|f(x,t) - f(y,t)| + |g(t)||x - y| \leq L|x - y| \tag{7.30}$$

Linear Growth: There exists a constant $K > 0$ such that for all $x \in \mathbb{R}^d$ and $t \in [0,T]$:

$$|f(x,t)|^2 + |g(t)|^2 \leq K(1 + |x|^2) \tag{7.31}$$

Score Function Regularity: The score function $\nabla_x \log p_t(x)$ must be well-defined and satisfy appropriate regularity conditions.

For our specific SDEs, theses conditions are trivially respected.

Practical Implications

These conditions ensure that:

1. The reverse-time SDE has a unique strong solution
2. The solution is pathwise unique
3. The process is well-defined for numerical simulation

The regularity of the score function is typically ensured by the smoothness of the data distribution and the Gaussian perturbation kernels used in practice.

A.5 Derivation of the Probability Flow ODE

The probability flow ODE emerges from a careful analysis of the Fokker-Planck equation. For a general SDE:

$$dx = f(x,t)dt + G(x,t)dw \tag{7.32}$$

the marginal probability density $p_t(x)$ evolves according to the Kolmogorov forward equation (Fokker-Planck equation):

$$\frac{\partial p_t(x)}{\partial t} = -\sum_{i=1}^{d} \frac{\partial}{\partial x_i}[f_i(x,t)p_t(x)] + \frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{d} \frac{\partial^2}{\partial x_i \partial x_j}\left[\sum_{k=1}^{d} G_{ik}(x,t)G_{jk}(x,t)p_t(x)\right] \tag{7.33}$$

Through careful algebraic manipulation, this can be rewritten as:

$$\frac{\partial p_t(x)}{\partial t} = -\sum_{i=1}^{d} \frac{\partial}{\partial x_i}[\tilde{f}_i(x,t)p_t(x)] \tag{7.34}$$

where:

$$\tilde{f}(x,t) = f(x,t) - \frac{1}{2}\nabla \cdot [G(x,t)G(x,t)^T] - \frac{1}{2}G(x,t)G(x,t)^T\nabla_x \log p_t(x) \tag{7.35}$$

This form corresponds to the Kolmogorov forward equation of a deterministic ODE:

$$dx = \tilde{f}(x,t)dt \tag{7.36}$$

For our simplified case where $G(x,t) = g(t)I$, this reduces to the probability flow ODE in equation 2.28. For more detailled demonstrations, please consult [14].

A.6 Derivation of Denoising Score Matching

We start with the score-matching objective to minimize:

$$\mathcal{L}(\theta) = \frac{1}{2}\mathbb{E}_{q_\sigma(x)}\left[\|\nabla_x \log q_\sigma(x) - s_\theta(x)\|_2^2\right] \tag{7.37}$$

Expansion of the squared norm

Expanding the squared term:

$$\mathcal{L}(\theta) = \frac{1}{2}\int q_\sigma(x)\left[\|\nabla_x \log q_\sigma(x)\|_2^2 + \|s_\theta(x)\|_2^2 - 2\nabla_x \log q_\sigma(x)^T s_\theta(x)\right] dx \tag{7.38}$$

Identification of terms irrelevant for minimization

The first term $\frac{1}{2}\int q_\sigma(x)\|\nabla_x \log q_\sigma(x)\|_2^2 dx$ is independent of θ, so we need to minimize:

$$\mathcal{L}(\theta) = \text{minimize}_\theta \left\{\frac{1}{2}\int q_\sigma(x)\|s_\theta(x)\|_2^2 dx - \int q_\sigma(x)\nabla_x \log q_\sigma(x)^T s_\theta(x)dx\right\} \tag{7.39}$$

Transformation using the definition of the score

Since $\nabla_x \log q_\sigma(x) = \frac{\nabla_x q_\sigma(x)}{q_\sigma(x)}$, the second term becomes:

$$\int q_\sigma(x)\nabla_x \log q_\sigma(x)^T s_\theta(x)dx = \int \nabla_x q_\sigma(x)^T s_\theta(x)dx$$

Expressing in terms of conditional distributions

Using the fact that $q_\sigma(x) = \int q_\sigma(x|x')p_{data}(x')dx'$ and $\nabla_x q_\sigma(x) = \int \nabla_x q_\sigma(x|x')p_{data}(x')dx'$, we can rewrite:

$$\mathcal{L}(\theta) = \text{minimize}_\theta \left\{\frac{1}{2}\iint q_\sigma(x|x')p_{data}(x')\|s_\theta(x)\|_2^2 dxdx' \quad - \iint (\nabla_x q_\sigma(x|x')p_{data}(x'))^T s_\theta(x)dxdx'\right\}$$

Converting to expectation form

This can be expressed as:

$$\mathcal{L}(\theta) = \text{minimize}_\theta \left\{ \frac{1}{2}\mathbb{E}_{p_{data}(x')}\mathbb{E}_{q_\sigma(x|x')} \left[\|s_\theta(x)\|_2^2 \right] \quad - \mathbb{E}_{p_{data}(x')}\mathbb{E}_{q_\sigma(x|x')} \left[s_\theta(x)^T \nabla_x \log q_\sigma(x|x') \right] \right\}$$

Final denoising score matching objective

By including the constant term $\frac{1}{2}\mathbb{E}_{p_{data}(x')}\mathbb{E}_{q_\sigma(x|x')} [\|\nabla_x \log q_\sigma(x|x')\|_2^2]$ that doesn't depend on θ, we obtain the final denoising score matching objective:

$$\boxed{\mathcal{L}_{DSM}(\theta) = \frac{1}{2}\mathbb{E}_{p_{data}(x')}\mathbb{E}_{q_\sigma(x|x')} \left[\|s_\theta(x) - \nabla_x \log q_\sigma(x|x')\|_2^2 \right] \qquad (7.40)}$$

This formulation is tractable because for Gaussian perturbations $q_\sigma(x|x') = \mathcal{N}(x; x', \sigma^2 I)$, the score $\nabla_x \log q_\sigma(x|x')$ can be computed in closed form.

A.7 Parameters of fast sampling and comparision

Tolerance parameters :

The tolerance parameters are:
- abs (absolute tolerance): set to $\frac{y_{\max} - y_{\min}}{256}$ where $y_{\max}, y_{\min}$ define the image value range.
- rel (relative tolerance): Controls the speed-quality trade-off (typically rel $\in [0.01, 0.1]$)

The step size is accepted if $E_2 \leq 1$ and updated according to:

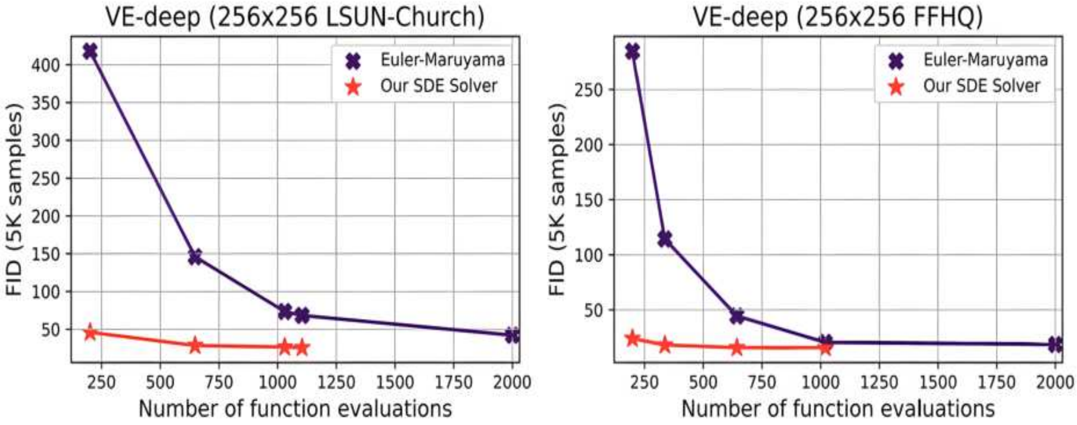
$$h_{\text{new}} = \min(h_{\max}, \theta \times h \times E_2^{-r}) \qquad (7.41)$$

where:
- $h_{\max} = \max(t - \varepsilon, 0)$ is the maximum allowable step size (limited by remaining time)
- $\theta = 0.9$ is a safety factor to prevent aggressive step size increases
- $r = 0.9$ is the exponent scaling term for step size adaptation
- h is the current step size
- ε is a small constant (e.g., 10^{-3}) to avoid numerical issues at $t = 0$

Performance Comparison

According to Jolicoeur-Martineau et al. and as seen in the figure 7.1, the fast sampling method achieves:
- **2-10x speedup** over Euler-Maruyama with equal or better sample quality
- **Superior high-resolution performance**: Particularly effective for generating high-resolution images
- **No hyperparameter tuning**: Automatically adapts step sizes without manual schedule design

The method demonstrates particular advantages for VE-SDEs, where the large noise scales benefit from adaptive step size control.



Performance comparison between the proposed fast adaptive SDE solver (red stars) and standard Euler-Maruyama (purple crosses) on high-resolution datasets. The fast solver achieves significantly better FID scores with fewer function evaluations, demonstrating both speed and quality improvements.

Figure 7.1: Fast adaptive sampling performance comparison on VE-deep models. Source :[7]

B. Theoretical Foundations of Downstream Tasks

This appendix provides complete theoretical foundations for the downstream tasks enabled by score-based generative models.

B.1 Complete Image Inpainting Theory

Problem Formulation

Consider an incomplete image y where only a subset of pixels $\Omega(y)$ are known. Our goal is to sample from the conditional distribution:

$$p(\Omega(\bar{x}(0))|\Omega(x(0)) = y) \tag{7.42}$$

where $\Omega(\bar{x}(0))$ denotes the unknown regions and $\Omega(x(0)) = y$ represents the constraint that known regions match the observed values.

SDE Formulation for Unknown Regions

We define a new diffusion process $z(t) = \Omega(\bar{x}(t))$ that evolves only the unknown dimensions. For the general SDE:

$$dx = f(x,t)dt + G(x,t)dw \tag{7.43}$$

the SDE for the unknown regions becomes:

$$dz = f_\Omega(z,t)dt + G_\Omega(z,t)dw \tag{7.44}$$

where f_Ω and G_Ω are the drift and diffusion coefficients restricted to the unknown dimensions.

Conditional Reverse-Time SDE

The reverse-time SDE conditioned on $\Omega(x(0)) = y$ is:

$$dz = \left[f_\Omega(z,t) - \nabla\cdot[G_\Omega(z,t)G_\Omega(z,t)^T] - G_\Omega(z,t)G_\Omega(z,t)^T\nabla_z \log p_t(z|\Omega(z(0)) = y)\right]dt + G_\Omega(z,t)d\bar{w} \tag{7.45}$$

The crucial challenge lies in computing the intractable conditional score $\nabla_z \log p_t(z(t)|\Omega(x(0)) = y)$. Yang Song et al. provide an elegant approximation based on the following insight.

Derivation of the Approximation

Let A denote the event $\Omega(x(0)) = y$. We want to approximate:

$$p_t(z(t)|\Omega(x(0)) = y) = p_t(z(t)|A) \tag{7.46}$$

Using the law of total expectation:

$$p_t(z(t)|A) = \int p_t(z(t)|\Omega(x(t)), A)p_t(\Omega(x(t))|A)d\Omega(x(t)) \tag{7.47}$$

$$= \mathbb{E}_{p_t(\Omega(x(t))|A)}[p_t(z(t)|\Omega(x(t)), A)] \tag{7.48}$$

The key approximation is:

$$\mathbb{E}_{p_t(\Omega(x(t))|A)}[p_t(z(t)|\Omega(x(t)), A)] \approx \mathbb{E}_{p_t(\Omega(x(t))|A)}[p_t(z(t)|\Omega(x(t)))] \tag{7.49}$$

This assumes that conditioning on the event A (known values at time 0) does not significantly affect the relationship between unknown and known regions at time t.

In practice, we approximate:

$$p_t(z(t)|\Omega(x(0)) = y) \approx p_t(z(t)|\hat{\Omega}(x(t))) \tag{7.50}$$

where $\hat{\Omega}(x(t))$ is a sample from $p_t(\Omega(x(t))|A)$, which is typically a tractable distribution.

Therefore, the conditional score becomes:

$$\nabla_z \log p_t(z(t)|\Omega(x(0)) = y) \approx \nabla_z \log p_t([z(t); \hat{\Omega}(x(t))]) \tag{7.51}$$

where $[z(t); \hat{\Omega}(x(t))]$ denotes the reconstruction of the complete image by combining unknown regions $z(t)$ with a sample of known regions $\hat{\Omega}(x(t))$.

B.2 Complete Colorization Theory

Colorization presents a unique challenge compared to standard inpainting because the known information (grayscale values) is coupled across color channels rather than being localized to specific spatial regions.

Problem Formulation

Given a grayscale image, we want to sample from:

$$p(\text{RGB channels}|\text{luminance information}) \tag{7.52}$$

The Coupling Problem

Unlike spatial inpainting where unknown regions are independent of known regions, in colorization:

- All three RGB channels contribute to the same luminance value
- We cannot treat color channels as independent variables
- Direct application of inpainting would violate the luminance constraint

Orthogonal Transformation Solution

Yang Song et al. solve this by using an orthogonal transformation that decouples luminance from chrominance information using passage from RGB space to Yuv espace.

The specific orthogonal matrix used is:

$$M = \begin{pmatrix} 0.577 & -0.816 & 0 \\ 0.577 & 0.408 & 0.707 \\ 0.577 & 0.408 & -0.707 \end{pmatrix} \tag{7.53}$$

Preservation of Stochastic Properties

A crucial theoretical insight is that orthogonal transformations preserve the properties of Wiener processes:

Wiener Process Preservation

If $w(t) = [w_1(t), w_2(t), w_3(t)]^T$ is a standard 3-dimensional Wiener process, then $\tilde{w}(t) = Mw(t)$ is also a standard 3-dimensional Wiener process.

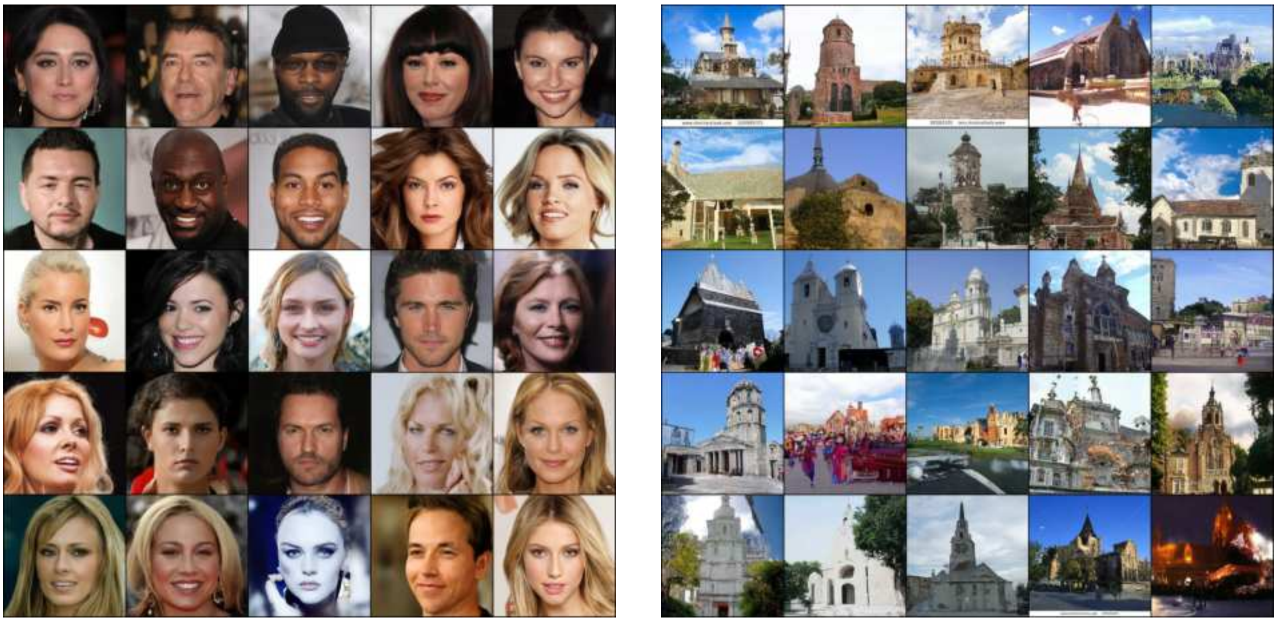
Proof:

- $\mathbb{E}[\tilde{w}(t)] = M\mathbb{E}[w(t)] = 0$
- $\text{Cov}[\tilde{w}(t)] = M\text{Cov}[w(t)]M^T = M \cdot tI \cdot M^T = tI$
- Independence and Gaussian properties are preserved under linear transformations

C. Extended Results

This appendix provides extended visual results and analysis for all experimental tasks conducted during the internship. These additional samples complement the main results presented in Chapter 4 and demonstrate the consistency, diversity, and limitations of our score-based generative models across different datasets and applications.

C.1 Additional Unconditional Samples



CelebA-HQ-256 extended samples

LSUN Church-256 extended samples

Figure 7.2: Extended samples from CelebA-HQ-256 and LSUN Church-256

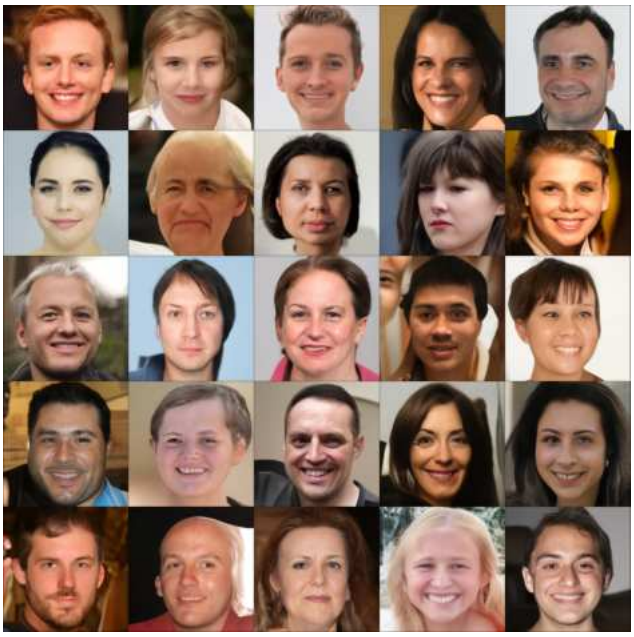
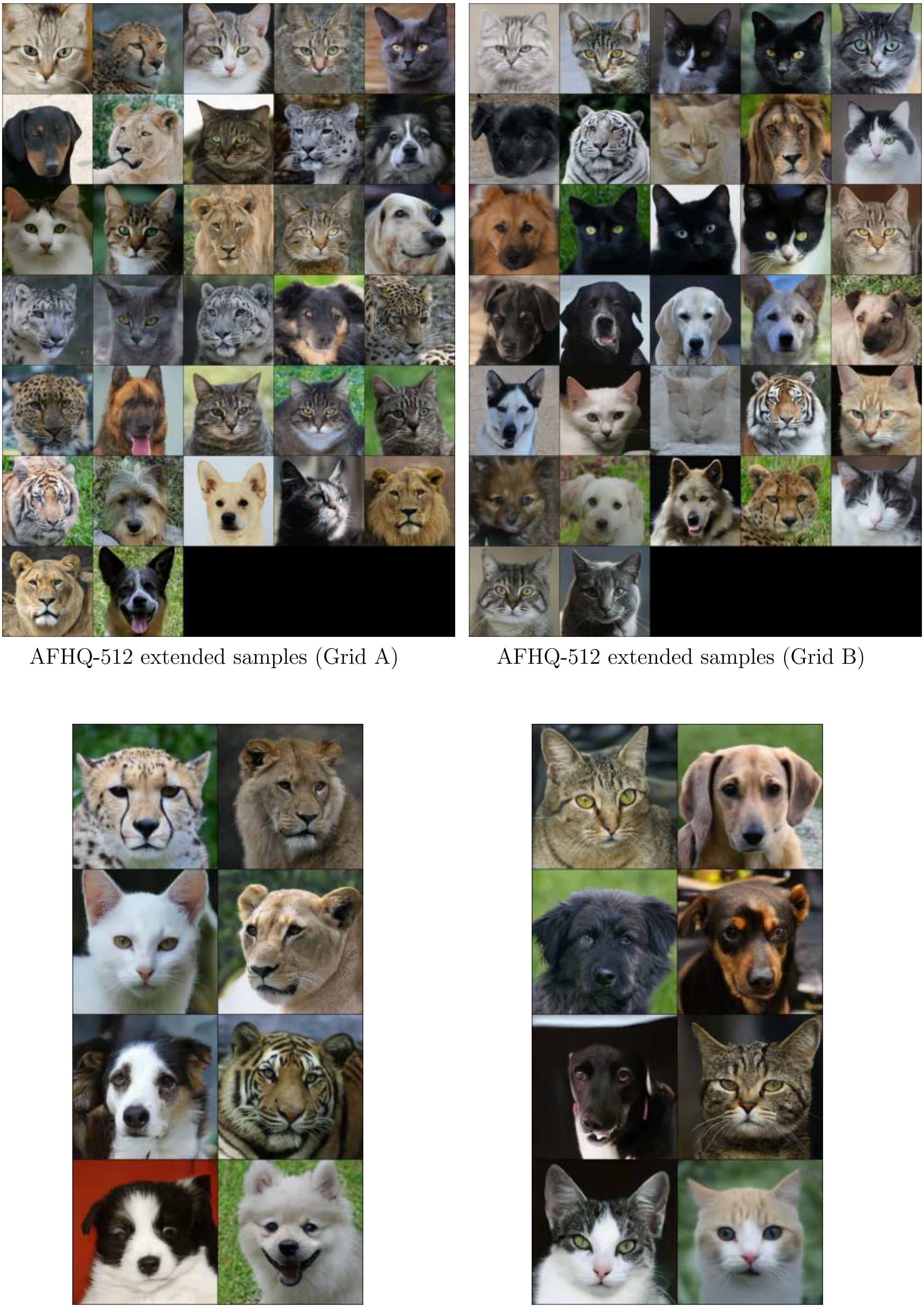


Figure 7.3: Extended FFHQ-1024 samples



AFHQ-512 extended samples (Grid A)

AFHQ-512 extended samples (Grid B)

AFHQ-512 extended samples (Grid C)

AFHQ-512 extended samples (Grid D)

Figure 7.4: Extended unconditional samples from our custom AFHQ-512 model showing consistent quality and diversity

C.2 Extended Inpainting Results

This section presents comprehensive inpainting results across all datasets using various masking patterns and strategies.

Diverse Masking Strategies

We evaluated inpainting performance using multiple masking approaches including geometric shapes (squares, circles, irregular patches), random masking patterns, and creative artistic masks. Each strategy tests different aspects of the model's ability to understand spatial relationships and generate coherent completions.

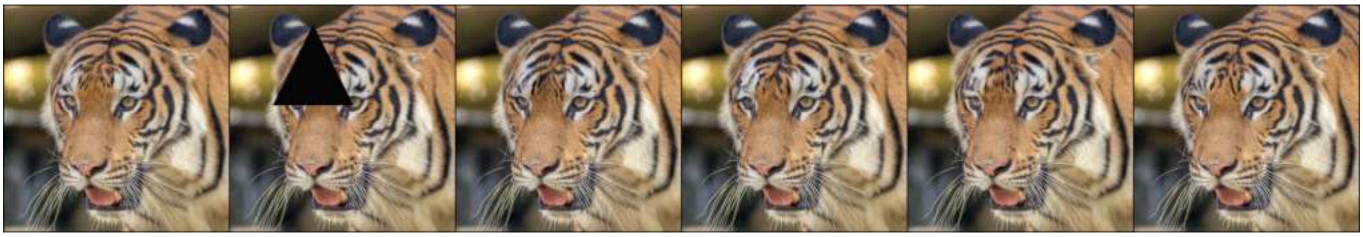


Figure 7.5: AFHQ-512 inpainting

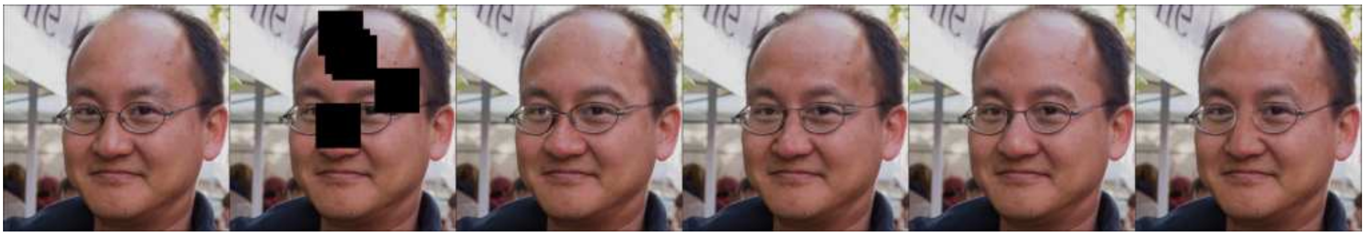


Figure 7.6: FFHQ-1024 inpainting

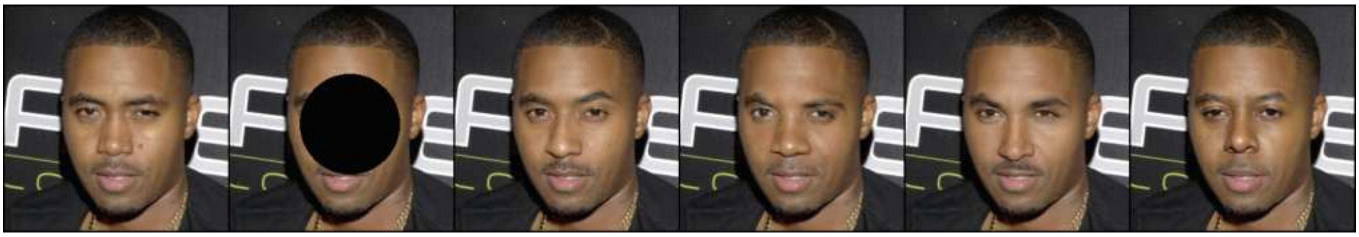


Figure 7.7: Celebahq-256 inpainting

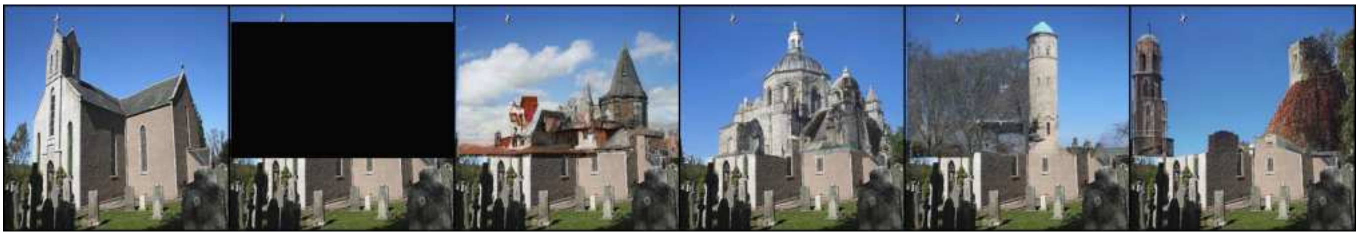


Figure 7.8: LSUN church inpainting

C.3 Extended Colorization Results

The colorization experiments demonstrate the model's ability to generate diverse and realistic color schemes while revealing systematic limitations in semantic consistency and color saturation.

Diverse Colorization Scenarios

We tested colorization across various image types and lighting conditions to assess the robustness of the orthogonal transformation approach.

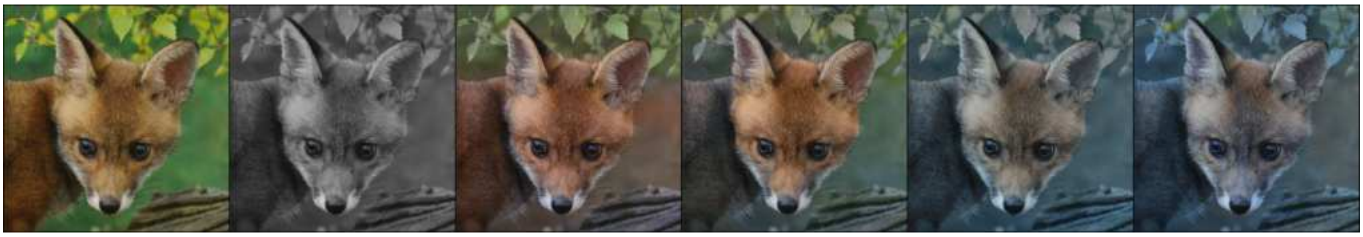


Figure 7.9: AFHQ-512 colorization

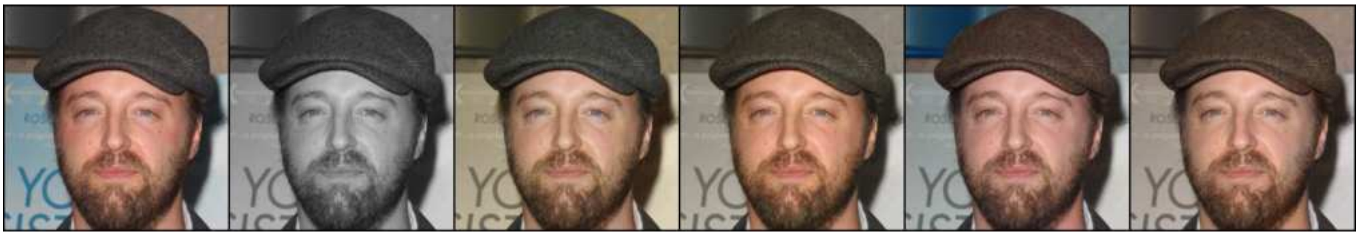


Figure 7.10: CelebA-HQ-256 colorization

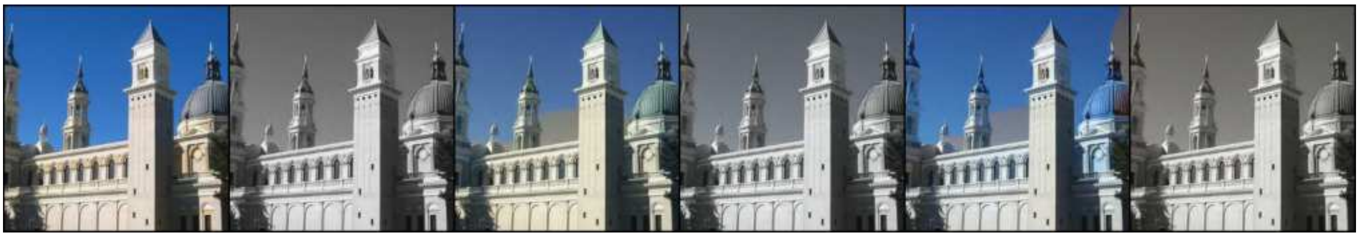


Figure 7.11: LSUN Church colorization

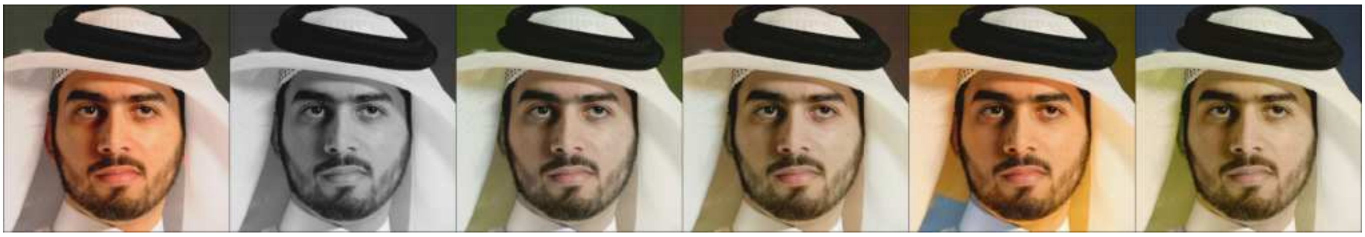
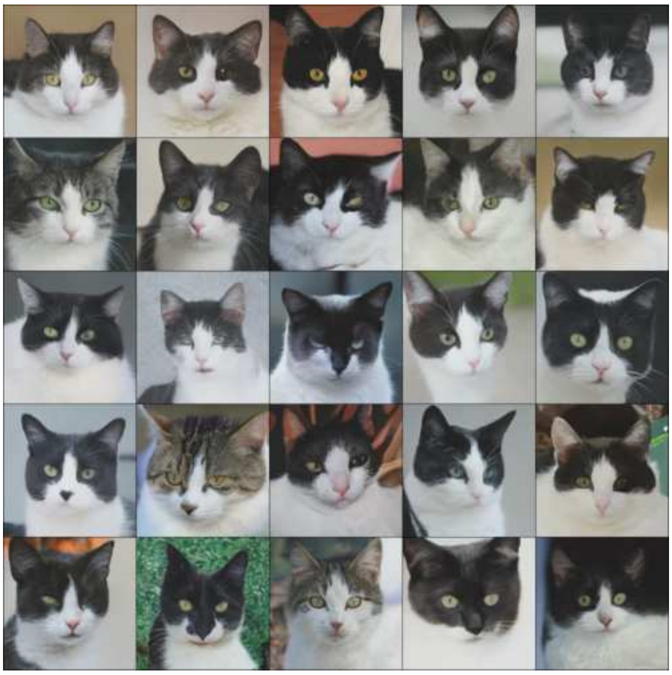


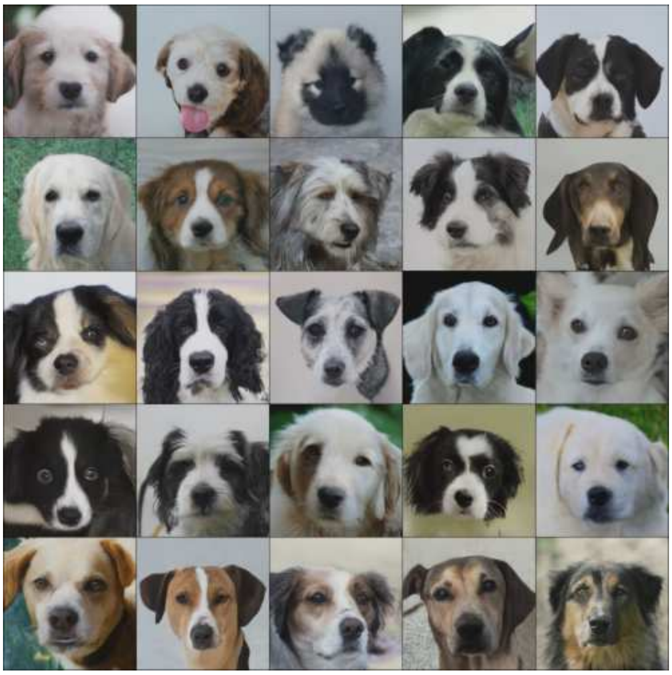
Figure 7.12: FFHQ-1024 colorization

C.4 Extended Class-Conditional AFHQ-512 Results

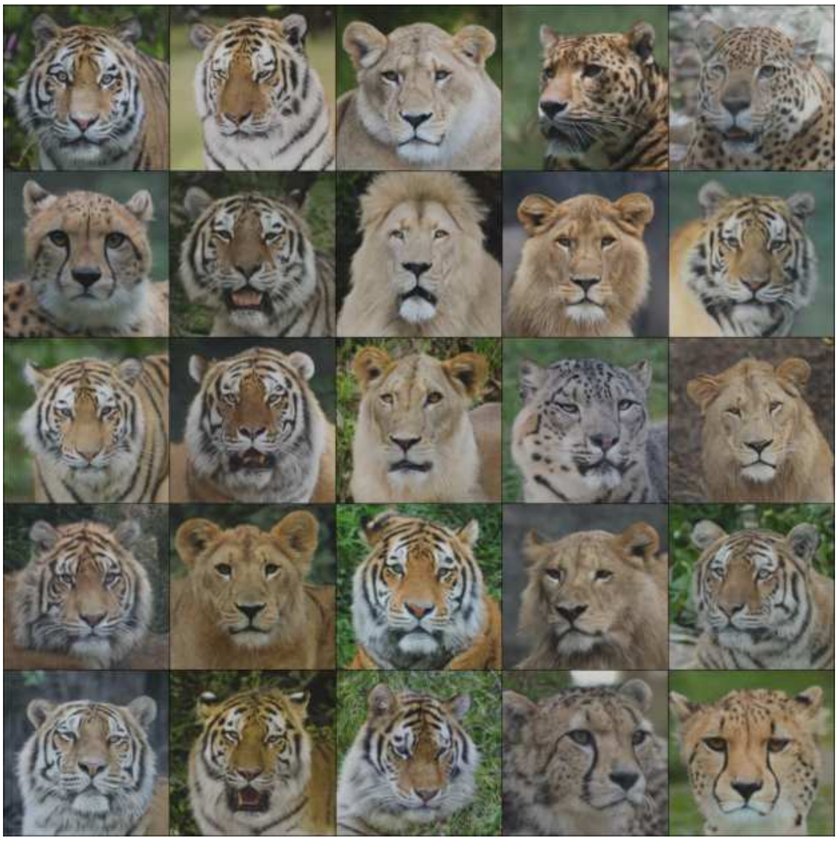
This section provides comprehensive results for all three classifier guidance strategies.



Strategy 1: Amplified guidance scale with cat



Strategy 2: Adaptive linear scaling with dog



Strategy 3: Sigma truncation with wild

Figure 7.13: Controllable generation per class with three strategies