





# Score-Based Generative Modeling Through Stochastic Differential Equations (SDE)

 Read when	@December 25, 2024
 Field	AI: Generative Model
 DOI	<a href="https://arxiv.org/pdf/2011.13456">https://arxiv.org/pdf/2011.13456</a>
 Status	Reading

## Giới thiệu

- **Vấn đề đặt ra:**

Trong học máy, việc thêm nhiễu vào dữ liệu  $p_0(x)$  từ một phân phối phức tạp (ví dụ, ảnh) để chuyển đổi thành một phân phối đơn giản như Gaussian là tương đối dễ dàng. Tuy nhiên, việc tái tạo dữ liệu từ phân phối nhiễu này (quy trình ngược) là một thách thức lớn, đặc biệt khi cần sinh mẫu có chất lượng cao và bảo toàn tính ngẫu nhiên.

Các phương pháp truyền thống thường gặp khó khăn trong việc:

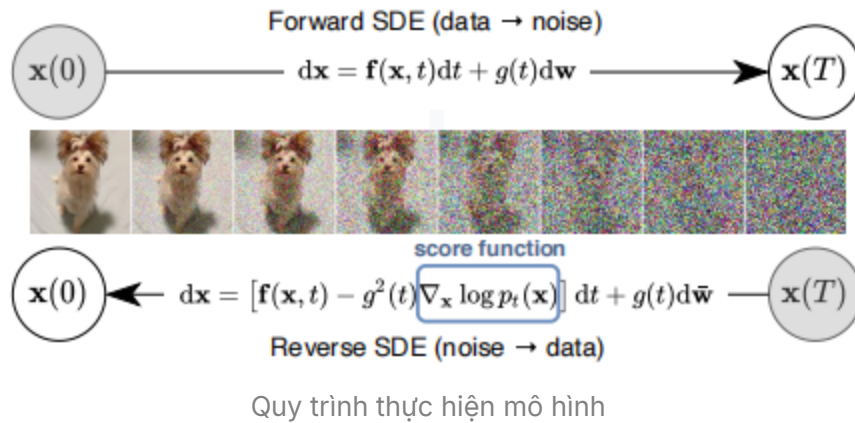
- Cân bằng giữa chất lượng mẫu và tốc độ sinh
- Đảm bảo tính đa dạng của dữ liệu sinh ra
- Kiểm soát quá trình sinh một cách linh hoạt

- **Mục tiêu:**

Báo cáo này giới thiệu một phương pháp tổng quát dựa trên phương trình vi phân ngẫu nhiên (Stochastic Differential Equations - SDE) để:

1. Làm nhiễu dữ liệu một cách liên tục (Forward SDE).
2. Tái tạo dữ liệu từ nhiễu bằng cách loại bỏ nhiễu (Reverse SDE).

3. Tích hợp các kỹ thuật ước lượng **score** để định hướng dữ liệu về phân phối ban đầu.



## Lý thuyết nền tảng

### Score và Score Matching

- Score là gì?**

Score trong xác suất là gradient của log xác suất  $p(x)$  theo dữ liệu  $x$ :  $\nabla_x \log p(x)$

Score cho biết hướng mà xác suất  $p(x)$  tăng nhanh nhất khi di chuyển trong không gian dữ liệu.

Ví dụ, với phân phối Gaussian  $p(x) = \mathcal{N}(x; \mu, \sigma^2)$ , score là:  $\nabla_x \log p(x) = -\frac{x - \mu}{\sigma^2}$

- Score Matching là gì?**

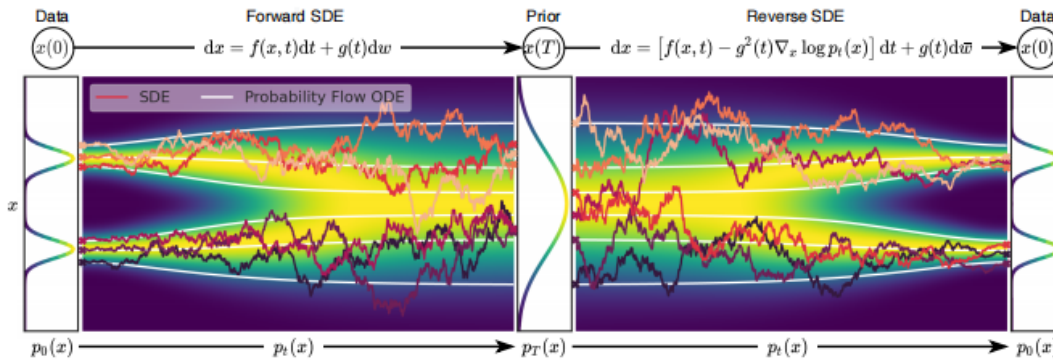
Đây là kỹ thuật học máy nhằm huấn luyện mô hình  $s_\theta(x)$  (thường là mạng neural) để xấp xỉ score thực.

Hàm loss:

$$L(\theta) = \mathbb{E}_{p_{\text{data}}(x)} \|s_\theta(x) - \nabla_x \log p(x)\|_2^2$$

Ưu điểm: Không cần biết hằng số chuẩn hóa của phân phối.

### Phương trình vi phân ngẫu nhiên (SDE)



Tổng quan về SDEs

- **SDE là gì?**

Phương trình vi phân ngẫu nhiên (SDE) mô tả sự thay đổi của một hệ thống ngẫu nhiên theo thời gian. Công thức tổng quát:

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

- $x$ : trạng thái hệ thống.
- $f(x, t)$ : drift coefficient (định hướng di chuyển).
- $g(t)$ : diffusion coefficient (độ nhiễu).
- $w$ : quá trình Wiener (chuyển động Brownian).

- **Forward SDE:**

Dùng để thêm nhiễu vào dữ liệu, chuyển đổi  $p_0(x)$  thành phân phối Gaussian  $\mathcal{N}(0, I)$ :

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

- **Reverse SDE:**

Dùng để tái tạo dữ liệu từ nhiễu:

$$dx = [f(x, t) - g(t)^2 \nabla_x \log p_t(x)] dt + g(t)dw$$

## Langevin Dynamics

Langevin Dynamics là một phương pháp sinh dữ liệu từ phân phối  $p(x)$  thông qua:

$$x_{t+1} = x_t + \epsilon \cdot \nabla_x \log p(x) + \sqrt{2\epsilon}z$$

- $\nabla_x \log p(x)$ : Score kéo trạng thái  $x$  về vùng xác suất cao hơn.
- $z \sim \mathcal{N}(0, I)$ : Nhiễu Gaussian.

## Framework: Score-Based Generative Modeling with SDE

### Forward SDE: Làm nhiễu dữ liệu

Forward SDE thêm nhiễu vào dữ liệu theo thời gian. Hai loại phổ biến:

#### 1. Variance Exploding SDE (VE SDE):

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

#### 2. Variance Preserving SDE (VP SDE):

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

- Biến đổi  $p_0(x)$  thành  $\mathcal{N}(0, I)$
- Cho phép theo dõi phân phối trung gian  $p_t(x)$

### Reverse SDE: Tái tạo dữ liệu

Reverse SDE tái tạo dữ liệu từ nhiễu:

$$dx = [f(x, t) - g(t)^2 \nabla_x \log p_t(x)] dt + g(t)dw$$

### Probability Flow ODE

Reverse SDE có thể chuyển thành một phương trình ODE:

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

## Huấn luyện mô hình Score

Huấn luyện mạng  $s_\theta(x, t)$  qua loss function có trọng số theo thời gian:

$$\theta^* = \arg \min_{\theta} \mathbb{E}_t \left[ \lambda(t) \mathbb{E}_{p_0(x)} \mathbb{E}_{p_t(x_t|x)} \|s_\theta(x_t, t) - \nabla_x \log p_t(x_t|x)\|_2^2 \right]$$

## Kỹ thuật sinh mẫu

### Giải Reverse SDE bằng số (Numerical SDE Solver)

Dùng Euler-Maruyama:

$$x_{t+\Delta t} = x_t + [f(x_t, t) - g(t)^2 \nabla_x \log p_t(x_t)] \Delta t + g(t) \sqrt{\Delta t} z$$

- $z \sim \mathcal{N}(0, I)$ .

Adaptive step size để cân bằng tốc độ và chất lượng.

### Predictor-Corrector Sampler

1. **Predictor:** Dự đoán bước tiếp theo bằng cách sử dụng Reverse SDE.
2. **Corrector:** Tinh chỉnh với Langevin Dynamics:

$$x_{t+1} = x_t + \epsilon \cdot s_\theta(x_t, t) + \sqrt{2\epsilon} z$$

Cho phép kiểm soát trade-off giữa chất lượng và tốc độ.

## Kết quả thực nghiệm

1. **Chất lượng mẫu:**

- **CIFAR-10:**
  - FID: 2.20 (state-of-the-art)
  - Inception Score: 9.89
  - Ổn định trong huấn luyện

- Đa dạng trong mẫu sinh ra
- **Ảnh độ phân giải cao:**
  - Sinh được ảnh 1024×1024 chất lượng cao
  - Giữ được chi tiết và tính nhất quán

## 2. Ứng dụng:

- **Tô màu ảnh (Colorization):** Biến ảnh grayscale thành ảnh màu.
- **Làm đầy ảnh (Inpainting):** Hoàn thiện các vùng bị thiếu.
- **Sinh mẫu có điều kiện (Conditional Generation):** Dựa trên nhãn lớp.

# Kết luận

## Đóng góp chính

1. Framework thống nhất cho sinh dữ liệu dựa trên SDE
2. Phương pháp huấn luyện score hiệu quả
3. Kỹ thuật sinh mẫu linh hoạt

## Hạn chế và hướng phát triển

1. Chi phí tính toán cao khi sinh mẫu
2. Cần cải thiện tốc độ sinh
3. Mở rộng cho các dạng dữ liệu khác

## Tác động

- Tiến bộ trong sinh dữ liệu chất lượng cao
- Mở ra ứng dụng thực tiễn mới
- Cơ sở cho nghiên cứu tiếp theo