





# Denoising Diffusion Probabilistic Models

 Read when	@December 24, 2024
 Field	AI: Generative Model
 DOI	<a href="https://arxiv.org/pdf/2006.11239">https://arxiv.org/pdf/2006.11239</a>
 Status	Reading

## Giới thiệu về bài báo

Bài báo giới thiệu một lớp các mô hình biến tiềm ẩn (latent variable model) được gọi là **mô hình khuếch tán (diffusion model)** để tổng hợp hình ảnh chất lượng cao.

Trước khi đi sâu hơn vào bài báo, có một số khái niệm cần hiểu:

### ***Biến tiềm ẩn***

Biến tiềm ẩn là biến không được quan sát trực tiếp mà phải quan sát bằng suy luận thống kê thường thông qua một mô hình toán học từ các biến khác được quan sát (trực tiếp đo lường). Ví dụ: lòng tự trọng, sở thích cá nhân, hiệu quả quản lý, ...

### ***Mô hình biến tiềm ẩn***

Đây là các mô hình toán học có mục đích giải thích các biến quan sát dưới dạng biến tiềm ẩn.

### ***Khuếch tán là gì?***

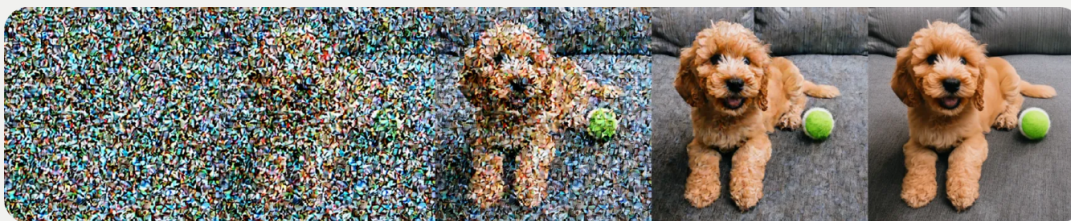
Trong vật lý, sự khuếch tán là quá trình mà các phân tử lan ra từ các vùng có nồng độ cao hơn đến các vùng có nồng độ thấp hơn. Khái niệm này có liên quan chặt chẽ đến chuyển động Brown, trong đó các hạt chuyển động ngẫu nhiên khi chúng va chạm với các phân tử trong chất lỏng và dần dần lan ra theo thời gian.

Những khái niệm này đã truyền cảm hứng cho sự phát triển của các mô hình khuếch tán trong AI tạo sinh.

### ***Mô hình xác suất khuếch tán***

Gọi tắt là mô hình khuếch tán (**diffusion model**) là một chuỗi Markov được tham số hóa sử dụng **variational inference** để tạo ra các mẫu phù hợp với dữ liệu sau thời gian hữu hạn.

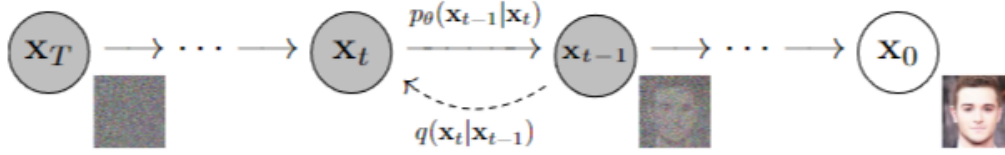
Các mô hình này hoạt động bằng cách dần dần thêm nhiễu vào dữ liệu và sau đó học cách đảo ngược quá trình đó để tạo ra dữ liệu mới. Việc này tương tự như ý tưởng về khuếch tán ngược trong vật lý. Về mặt lý thuyết, khuếch tán có thể được theo dõi ngược lại để đưa các hạt trở lại trạng thái ban đầu của chúng. Tương tự như vậy, các mô hình khuếch tán học cách đảo ngược nhiễu đã thêm vào để tạo ra dữ liệu mới thực tế từ các đầu vào nhiễu.



Một ví dụ về việc sử dụng mô hình khuếch tán để tạo hình ảnh.

## **Ý tưởng cốt lõi: Đảo ngược quá trình phân tán**

Các mô hình phân tán tạo ra dữ liệu bằng cách **đảo ngược một quá trình thêm nhiễu dần dần**. Ý tưởng chính là huấn luyện một chuỗi Markov được tham số hóa để dần dần thêm nhiễu (forward process), sau đó học quy trình ngược lại để khử nhiễu dữ liệu và tạo ra các mẫu mới.



Đồ thị biểu diễn quy trình hoạt động của mô hình

## Kiến thức cần nắm về mô hình khuếch tán

### Kiến thức toán học chi tiết

Mô hình khuếch tán là mô hình biến tiềm ẩn:

$$p_{\theta}(\mathbf{x}_0) := \int p_{\theta}(\mathbf{x}_{0:T}) d\mathbf{x}_{1:T}$$

với  $\mathbf{x}_1, \dots, \mathbf{x}_T$  là các tiềm ẩn cùng chiều với dữ liệu  $\mathbf{x}_0$  được lấy mẫu từ phân phối  $q(\mathbf{x}_0)$

Phân phối chung  $p_{\theta}(\mathbf{x}_{0:T})$  được gọi là **quy trình đảo (reverse process)**. Nó được định nghĩa là một chuỗi Markov với chuyển tiếp Gaussian đã được học, bắt đầu với phân phối:  $p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, I)$ :

$$p_{\theta}(\mathbf{x}_{0:T}) := p(\mathbf{x}_T) \prod_{t=1}^T p_{\theta}(\mathbf{x}_{t-1} | \mathbf{x}_t),$$

$$p_{\theta}(\mathbf{x}_{t-1} | \mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_{\theta}(\mathbf{x}_t, t), \boldsymbol{\Sigma}_{\theta}(\mathbf{x}_t, t))$$

Điều làm mô hình khuếch tán khác với các mô hình biến tiềm ẩn khác là ở việc xấp xỉ  $q(\mathbf{x}_{1:T} | \mathbf{x}_0)$ , được gọi là **quy trình tiến (forward process)** hoặc **quy trình khuếch tán**, được cố định bởi một chuỗi Markov dần thêm nhiễu Gaussian vào dữ liệu tương ứng với một variance schedule  $\beta_1, \dots, \beta_T$ :

$$q(\mathbf{x}_{1:T} | \mathbf{x}_0) := \prod_{t=1}^T q(\mathbf{x}_t | \mathbf{x}_{t-1}),$$

$$q(\mathbf{x}_t | \mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t} \mathbf{x}_{t-1}, \beta_t \mathbf{I})$$

Huấn luyện được biểu diễn bởi việc **tối ưu hóa giới hạn biến thiên thông thường (usual variational bound) trên negative log likelihood**:

$$\mathbb{E}[-\log p_{\theta}(\mathbf{x}_0)] \leq \mathbb{E}_q \left[ -\log \frac{p_{\theta}(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T} | \mathbf{x}_0)} \right] = \mathbb{E}_q \left[ -\log p(\mathbf{x}_T) - \sum_{t \geq 1} \log \frac{p_{\theta}(\mathbf{x}_{t-1} | \mathbf{x}_t)}{q(\mathbf{x}_t | \mathbf{x}_{t-1})} \right] =: L$$

Các phương sai của quá trình khuếch tán  $\beta_t$  có thể được học thông qua **tái tham số hóa (reparameterization)** hoặc được giữ cố định như là siêu tham số, và khả năng biểu diễn của quá trình ngược được đảm bảo một phần bởi việc lựa chọn các điều kiện Gaussian trong  $p_{\theta}(\mathbf{x}_{t-1} | \mathbf{x}_t)$ , vì

cả hai quá trình đều có cùng dạng hàm số khi  $\beta_t$  nhỏ. Một thuộc tính đáng chú ý của quá trình tiến là nó cho phép lấy mẫu  $\mathbf{x}_t$  tại một bước thời gian bất kỳ  $t$  theo dạng đóng: sử dụng ký hiệu  $\alpha_t := 1 - \beta_t$  và  $\bar{\alpha}_t := \prod_{s=1}^t \alpha_s$ , ta có

$$q(\mathbf{x}_t | \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t} \mathbf{x}_0, (1 - \bar{\alpha}_t) \mathbf{I})$$

Huấn luyện hiệu quả do đó có thể thực hiện được bằng cách tối ưu các hạng tử ngẫu nhiên của  $L$  bằng phương pháp gradient descent ngẫu nhiên. Cải thiện thêm nữa đến từ việc giảm phương sai bằng cách viết lại  $L$  như sau:

$$\mathbb{E}_q \left[ \underbrace{D_{\text{KL}}(q(\mathbf{x}_T | \mathbf{x}_0) \| p(\mathbf{x}_T))}_{L_T} + \sum_{t>1} \underbrace{D_{\text{KL}}(q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) \| p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t))}_{L_{t-1}} - \underbrace{\log p_\theta(\mathbf{x}_0 | \mathbf{x}_1)}_{L_0} \right]$$

Phương trình trên sử dụng phân kỳ KL để trực tiếp so sánh  $p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)$  với các hậu nghiệm của quá trình tiến, mà có thể truy vấn được khi điều kiện hóa trên  $\mathbf{x}_0$ :

$$q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t \mathbf{I})$$

trong đó

$$\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0) := \frac{\sqrt{\bar{\alpha}_{t-1}} \beta_t}{1 - \bar{\alpha}_t} \mathbf{x}_0 + \frac{\sqrt{\alpha_t} (1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} \mathbf{x}_t$$

và

$$\tilde{\beta}_t := \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t} \beta_t$$

Do đó, tất cả các phân kỳ KL trong phương trình trên đều là các so sánh giữa các Gaussian, vì vậy chúng có thể được tính toán theo cách Rao-Blackwellized với các biểu thức dạng đóng thay vì ước lượng Monte Carlo với phương sai cao.

## Tóm tắt

Mô hình gồm 2 quá trình:

### 1. Quá trình tiến (Quá trình phân tán)

- Quá trình này phá hủy thông tin trong dữ liệu bằng cách thêm nhiễu Gaussian dần dần. Bắt đầu với một hình ảnh rõ ràng và dần dần thêm nhiễu tính cho đến khi có nhiễu hoàn toàn.
- Nó được điều khiển bởi một **lịch trình phương sai (variance schedule)**  $\beta_t$  cho các bước thời gian  $t = 1, \dots, T$ , trong đó  $T$  là tổng số bước phân tán. Bài báo sử dụng một lịch trình tuyến tính đơn giản:  $\beta_T = 10^{-4}$  đến  $\beta_1 = 0.02$ . Những giá trị này là nhỏ so với dữ liệu được chuẩn hóa trong phạm vi  $[-1, 1]$ , đảm bảo rằng quá trình tiến **gần như có thể đảo ngược với các chuyển tiếp Gaussian**.

- Phương trình  $q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1 - \bar{\alpha}_t)\mathbf{I})$  cho phép lấy mẫu trực tiếp phiên bản nhiễu của dữ liệu  $\mathbf{x}_T$  tại bất kỳ bước thời gian  $t$  nào, dựa trên dữ liệu gốc  $\mathbf{x}_0$ . Do đó không cần phải mô phỏng toàn bộ quá trình tiến (forward process) từng bước một, giúp việc huấn luyện trở nên hiệu quả hơn.

## 2. Quá trình ngược (Khử nhiễu học được)

- Đây là nơi mô hình học cách sáng tạo! Quá trình ngược cũng là một chuỗi Markov nhưng với **các chuyển tiếp Gaussian học được**:  $p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$  trong đó  $\theta$  đại diện cho các tham số mô hình.
- Phần quan trọng nhất là xác định **giá trị trung bình  $\boldsymbol{\mu}_\theta(\mathbf{x}_t, t)$** . Các tác giả thử nghiệm với hai cách tham số hóa:
  - **Dự đoán giá trị trung bình của quá trình tiến  $\tilde{\boldsymbol{\mu}}$** : Điều này đơn giản, nhưng đòi hỏi phải huấn luyện trên toàn bộ giới hạn biến phân để có kết quả tối ưu.
  - **Dự đoán  $\epsilon$  (nhiều được thêm vào trong quá trình tiến)**: Đây là đối mới quan trọng! Dự đoán  $\epsilon$  dẫn đến một mục tiêu huấn luyện trực quan hơn, chất lượng mẫu tốt hơn và một kết nối mạnh mẽ với việc khớp điểm số khử nhiễu và động lực Langevin.

# Diffusion models và denoising autoencoders

## Quy trình khuếch tán và $L_T$

Các tác giả bỏ qua thực tế rằng các phương sai của quá trình tiến  $\beta_t$  có thể được học thông qua tái tham số hóa và thay vào đó cố định chúng thành các hằng số. Do đó, trong triển khai, hậu nghiệm xấp xỉ  $q$  không có tham số có thể học được, vì vậy  $L_T$  là một hằng số trong quá trình huấn luyện và có thể bỏ qua.

## Quy trình đảo ngược và $L_{1:T-1}$

Nói về  $p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$  với  $t > 1$

Đầu tiên, đặt  $\Sigma_\theta(x_t, t) = \sigma_t^2 \mathbf{I}$  để không phụ thuộc vào dữ liệu. Thử nghiệm cho thấy cả  $\sigma_t^2 = \beta_t$  và  $\sigma_t^2 = \tilde{\beta}_t$  mang lại kết quả tương tự. Lựa chọn đầu tiên tối ưu cho  $x_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ , và lựa chọn thứ hai là tối ưu cho  $x_t \sim q(x_t|x_0)$ . Đây là hai lựa chọn cực đoan, cho giới hạn trên và dưới về entropy của quá trình ngược với biến đồng biến chuẩn hoá  $\mathbf{I}$ .

Thứ hai, để biểu diễn giá trị kỳ vọng  $\mu_\theta(x_t)$ , đề xuất một cách tham số hoá cụ thể được thúc đẩy bởi phân tích sau:

Với  $q(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t \mathbf{I})$ , ta có thể viết:

$$L_{t-1} = \mathbb{E}_q \left[ \frac{1}{2\sigma_t^2} \|\tilde{\mu}_t(x_t, x_0) - \mu_\theta(x_t, t)\|^2 \right] + C \quad (8)$$

với  $C$  là một hằng số không phụ thuộc vào  $\mathbf{x}_0$ .

Do đó, tham số hoá đơn giản nhất của  $\mu_\theta(x_t, t)$  là một mô hình dự đoán giá trị kỳ vọng  $\tilde{\mu}$ . Tuy nhiên, chúng ta có thể tái tham số hóa phương trình 8 như sau:

$$\begin{aligned} L_{t-1} - C &= \mathbb{E}_{\mathbf{x}_0, \epsilon} \left[ \left\| \frac{1}{2\sigma_t^2} \left\| \tilde{\mu}_t \left( \mathbf{x}_t(\mathbf{x}_0, \epsilon), \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{x}_t(\mathbf{x}_0, \epsilon) - \sqrt{1 - \bar{\alpha}_t} \epsilon \right) \right) - \mu_\theta(\mathbf{x}_t(\mathbf{x}_0, \epsilon), t) \right\|^2 \right] \\ &= \mathbb{E}_{\mathbf{x}_0, \epsilon} \left[ \left\| \frac{1}{2\sigma_t^2} \left\| \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{x}_t(\mathbf{x}_0, \epsilon) - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon \right) - \mu_\theta(\mathbf{x}_t(\mathbf{x}_0, \epsilon), t) \right\|^2 \right] \end{aligned}$$

#### Algorithm 1 Training

```
1: repeat
2:    $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ 
3:    $t \sim \text{Uniform}(\{1, \dots, T\})$ 
4:    $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
5:   Take gradient descent step on
      $\nabla_\theta \left\| \epsilon - \epsilon_\theta(\sqrt{\alpha_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, t) \right\|^2$ 
6: until converged
```

#### Algorithm 2 Sampling

```
1:  $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
2: for  $t = T, \dots, 1$  do
3:    $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  if  $t > 1$ , else  $\mathbf{z} = \mathbf{0}$ 
4:    $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$ 
5: end for
6: return  $\mathbf{x}_0$ 
```

## Quy mô dữ liệu, bộ giải mã quá trình đảo ngược và $L_0$

Giả định rằng dữ liệu hình ảnh bao gồm các số nguyên trong tập  $\{0, 1, \dots, 255\}$  được chia tỷ lệ tuyến tính thành  $[-1, 1]$ . Điều này đảm bảo rằng quá trình đảo ngược của mạng neural hoạt động trên các đầu vào được chia tỷ lệ nhất quán bắt đầu từ phân phối chuẩn prior  $p(\mathbf{x}_T)$ . Để có được log likelihood rời rạc, đặt số hạng cuối cùng của quá trình đảo ngược thành một bộ giải mã rời rạc độc lập được lấy từ phân phối Gaussian  $\mathcal{N}(\mathbf{x}_0; \mu_\theta(\mathbf{x}_1, 1), \sigma_1^2 \mathbf{I})$ :

$$p_\theta(\mathbf{x}_0 | \mathbf{x}_1) = \prod_{i=1}^D \int_{\delta_-(x_0^i)}^{\delta_+(x_0^i)} \mathcal{N}(x; \mu_\theta^i(\mathbf{x}_1, 1), \sigma_1^2) dx$$

$$\delta_+(x) = \begin{cases} \infty & \text{if } x = 1 \\ x + \frac{1}{255} & \text{if } x < 1 \end{cases}$$

$$\delta_-(x) = \begin{cases} -\infty & \text{if } x = -1 \\ x - \frac{1}{255} & \text{if } x > -1 \end{cases}$$

trong đó  $D$  là chiều của dữ liệu và chỉ số  $i$  trên cùng chỉ ra việc trích xuất một tọa độ. Tương tự như các phân phối liên tục được rời rạc hóa được sử dụng trong các bộ giải mã VAE và mô hình tự hồi quy, lựa chọn ở đây đảm bảo rằng cận biên thiên là một độ dài mã không tổn thất của dữ liệu rời rạc, mà không cần thêm nhiễu vào dữ liệu hoặc tích hợp Jacobian của phép chia tỷ lệ vào log likelihood. Vào cuối quá trình lấy mẫu, hiển thị  $\mu_\theta(\mathbf{x}_1, 1)$  không mất tính tổng quát.

## Hàm mục tiêu huấn luyện đơn giản hóa

Với quá trình đảo ngược và bộ giải mã được định nghĩa ở trên, cận biến thiên, bao gồm các số hạng được lấy từ phương trình trên, có thể vi phân rõ ràng đối với  $\theta$  và sẵn sàng được sử dụng cho huấn luyện.

$$L_{\text{simple}}(\theta) = \mathbb{E}_{t, x_0, \epsilon} [\|\epsilon - \epsilon\theta(\sqrt{\alpha_t}\mathbf{x}_0 + \sqrt{1 - \alpha_t}\epsilon, t)\|^2]$$

trong đó  $t$  được chọn ngẫu nhiên đồng đều giữa 1 và  $T$ . Trường hợp  $t = 1$  tương ứng với  $L_0$  với tích phân trong định nghĩa bộ giải mã rời rạc được xấp xỉ bằng hàm mật độ xác suất Gaussian nhân với độ rộng bin, bỏ qua các hiệu ứng biên. Trường hợp  $t > 1$  tương ứng với phiên bản có trọng số của phương trình trên, tương tự như cách đánh trọng số mất mát được sử dụng bởi mô hình điểm số khử nhiễu NCSN. Thuật toán 1 mô tả quy trình huấn luyện hoàn chỉnh với hàm mục tiêu đơn giản hóa này.

Vì hàm mục tiêu đơn giản hóa bỏ qua việc đánh trọng số trong phương trình, nó là một cận biến thiên có trọng số nhấn mạnh các khía cạnh tái tạo khác nhau so với cận biến thiên chuẩn. Cụ thể, thiết lập quá trình khuếch tán khiến cho hàm mục tiêu đơn giản hóa giảm xuống thành mất mát bình phương tương ứng với  $L$  nhỏ. Những mất mát này huấn luyện mạng để khử nhiễu dữ liệu với một lượng nhiễu rất nhỏ, vì vậy nó có lợi để giảm trọng số của chúng để mạng có thể tập trung vào các nhiệm vụ khử nhiễu khó khăn hơn. Các tác giả thấy trong các thực nghiệm của họ rằng việc đánh trọng số lại dẫn đến chất lượng mẫu tốt hơn.

Table 1: CIFAR10 results. NLL measured in bits/dim.

Model	IS	FID	NLL Test (Train)
<b>Conditional</b>			
EBM [11]	8.30	37.9	
JEM [17]	8.76	38.4	
BigGAN [3]	9.22	14.73	
StyleGAN2 + ADA (v1) [29]	<b>10.06</b>	<b>2.67</b>	
<b>Unconditional</b>			
Diffusion (original) [53]			$\leq 5.40$
Gated PixelCNN [59]	4.60	65.93	3.03 (2.90)
Sparse Transformer [7]			<b>2.80</b>
PixelQNN [43]	5.29	49.46	
EBM [11]	6.78	38.2	
NCSNv2 [56]		31.75	
NCSN [55]	8.87 $\pm$ 0.12	25.32	
SNGAN [39]	8.22 $\pm$ 0.05	21.7	
SNGAN-DDLS [4]	9.09 $\pm$ 0.10	15.42	
StyleGAN2 + ADA (v1) [29]	<b>9.74 <math>\pm</math> 0.05</b>	3.26	
Ours ( $L$ , fixed isotropic $\Sigma$ )	7.67 $\pm$ 0.13	13.51	$\leq 3.70$ (3.69)
Ours ( $L_{\text{simple}}$ )	9.46 $\pm$ 0.11	<b>3.17</b>	$\leq 3.75$ (3.72)

Table 2: Unconditional CIFAR10 reverse process parameterization and training objective ablation. Blank entries were unstable to train and generated poor samples with out-of-range scores.

Objective	IS	FID
<b><math>\bar{\mu}</math> prediction (baseline)</b>		
$L$ , learned diagonal $\Sigma$	7.28 $\pm$ 0.10	23.69
$L$ , fixed isotropic $\Sigma$	8.06 $\pm$ 0.09	13.22
$\ \bar{\mu} - \bar{\mu}_\theta\ ^2$	—	—
<b><math>\epsilon</math> prediction (ours)</b>		
$L$ , learned diagonal $\Sigma$	—	—
$L$ , fixed isotropic $\Sigma$	7.67 $\pm$ 0.13	13.51
$\ \bar{\epsilon} - \epsilon_\theta\ ^2$ ( $L_{\text{simple}}$ )	<b>9.46 <math>\pm</math> 0.11</b>	<b>3.17</b>

## Thực nghiệm

### 1. Thiết lập Thực nghiệm

- Số bước khuếch tán  $T = 1000$
- Phương sai quá trình tiến ( $\beta$ ) tăng tuyến tính từ  $10^{-4}$  đến 0.02
- Dữ liệu được chuẩn hóa về khoảng  $[-1, 1]$
- Tỷ lệ tín hiệu/nhiều:  $L_{\text{KL}}(q(\mathbf{x}_t|\mathbf{x}_0)\|\mathcal{N}(0, \mathbf{I})) \approx 10^{-5}$  bits/chiều

### 2. Kết quả Chính

## 2.1 Chất lượng Mẫu

- Điểm FID trên CIFAR10: 3.17 (tập huấn luyện), 5.24 (tập kiểm tra)
- Vượt trội so với hầu hết các mô hình trong tài liệu, kể cả mô hình có điều kiện

Algorithm 3 Sending $\mathbf{x}_0$	Algorithm 4 Receiving
1: Send $\mathbf{x}_T \sim q(\mathbf{x}_T \mathbf{x}_0)$ using $p(\mathbf{x}_T)$	1: Receive $\mathbf{x}_T$ using $p(\mathbf{x}_T)$
2: <b>for</b> $t = T - 1, \dots, 2, 1$ <b>do</b>	2: <b>for</b> $t = T - 1, \dots, 1, 0$ <b>do</b>
3:   Send $\mathbf{x}_t \sim q(\mathbf{x}_t \mathbf{x}_{t+1}, \mathbf{x}_0)$ using $p_\theta(\mathbf{x}_t \mathbf{x}_{t+1})$	3:   Receive $\mathbf{x}_t$ using $p_\theta(\mathbf{x}_t \mathbf{x}_{t+1})$
4: <b>end for</b>	4: <b>end for</b>
5: Send $\mathbf{x}_0$ using $p_\theta(\mathbf{x}_0 \mathbf{x}_1)$	5: <b>return</b> $\mathbf{x}_0$

## 2.2 Phân tích Phương pháp

- Dự đoán nhiễu  $\epsilon$  hiệu quả khi huấn luyện trên cận biến thiên thực
- Mã hóa tiến bộ giúp nén dữ liệu hiệu quả với tỷ lệ nén 1.78 bits/dim
- Root mean squared error: 0.95 trên thang đo 0 đến 255

## 3. Cải tiến Kỹ thuật

- Sử dụng phương pháp nén tiến bộ (progressive coding)
- Thuật toán gửi-nhận được tối ưu hóa cho truyền tải hiệu quả
- Công thức ước lượng tiến bộ:  $x_0 \approx \hat{x}_0 = (x_t - \sqrt{1 - \alpha_t}\epsilon_\theta(x_t))/\sqrt{\alpha_t}$

## 4. Kết luận

- Mô hình đạt được cả chất lượng mẫu cao và khả năng nén hiệu quả
- Phương pháp mã hóa tiến bộ mở ra hướng mới cho việc nén dữ liệu dựa trên mô hình khuếch tán
- Kết quả thực nghiệm trên các tập dữ liệu CIFAR10 và LSUN cho thấy tính hiệu quả của phương pháp
- Các mẫu được tạo ra được lưu ở dropbox trong file readme của github repository.

## Phân tích sâu về mối liên hệ với việc khớp điểm số

Việc khớp điểm số cung cấp một cách học hàm điểm (gradient của mật độ xác suất log) của phân phối dữ liệu. Mối liên hệ giữa DDPM và việc khớp điểm số bắt nguồn từ việc chọn dự đoán  $\epsilon$  trong quá trình ngược:

- **Mục tiêu khớp điểm số:** Mục tiêu khớp điểm số giảm thiểu sự khác biệt giữa điểm số dự đoán của mô hình và điểm số thực của phân phối dữ liệu.



- **Mục tiêu đơn giản hóa DDPM:** Mục tiêu đơn giản hóa (Lsimple) trong DDPM gần giống với một phiên bản có trọng số của mục tiêu khớp điểm số. Bằng cách dự đoán nhiễu  $\epsilon$ , mô hình học được một hàm tỷ lệ thuận với hàm điểm ở các mức độ nhiễu khác nhau.

Mối liên hệ này rất sâu sắc:

- Nó giải thích tại sao DDPM có thể tạo ra các mẫu chất lượng cao mặc dù có điểm số log xác suất tương đối kém: mô hình ưu tiên học hàm điểm, điều này rất quan trọng để nắm bắt cấu trúc phức tạp của phân phối dữ liệu.
- Nó gợi ý rằng DDPM có thể được coi là học một đại diện ngầm của phân phối dữ liệu thông qua hàm điểm của nó, thay vì mô hình hóa trực tiếp mật độ.

## Khám phá cơ chế giải mã tiến triển

Vẻ đẹp của DDPM nằm ở **tính chất giải mã tiến triển** của chúng:

- **Tạo ra từ thô đến tinh:** Trong quá trình lấy mẫu, mô hình bắt đầu bằng việc loại bỏ nhiễu ở mức độ cao nhất, làm lộ ra các tính năng hình ảnh quy mô lớn. Khi quá trình ngược tiếp tục, các chi tiết tinh vi dần xuất hiện. Điều này được chứng minh thực nghiệm trong Hình 6 của bài báo.
- **Trao đổi tỷ lệ-méo:** Thuật toán 3 và 4 trình bày một sơ đồ nén mất mát tiến triển, nhấn mạnh rằng hầu hết thông tin trong không gian ẩn được dành cho những chi tiết tinh vi, thường không thể nhận thấy. Điều này giải thích chất lượng mẫu cao mà DDPM đạt được ngay cả khi log xác suất của chúng không phải là xuất sắc nhất.

## Hiểu mối liên hệ với các mô hình tự hồi quy

Các mô hình tự hồi quy tạo ra dữ liệu một phần tử tại một thời điểm, điều kiện hóa mỗi phần tử trên các phần tử đã tạo ra trước đó. Mặc dù có vẻ khác biệt, DDPM có thể được hiểu là một sự mở rộng của các mô hình tự hồi quy:

- **Sắp xếp bit tổng quát:** Trong các mô hình tự hồi quy, thứ tự tạo ra các phần tử là cố định. Các mô hình phân tán, thông qua quá trình tiêm nhiễu Gaussian, tạo ra một "sắp xếp bit" linh hoạt hơn, nơi thông tin bị phá hủy và phục hồi theo cách phân phối trên toàn bộ mẫu dữ liệu.
- **Liên tục so với rời rạc:** Các mô hình tự hồi quy phù hợp tự nhiên với dữ liệu rời rạc. Các mô hình phân tán mở rộng ý tưởng này với dữ liệu liên tục bằng cách sử dụng nhiễu Gaussian và hàm điểm như một hướng dẫn cho việc tạo ra.

## Những điểm quan trọng và ý nghĩa

- **Đảo ngược quá trình phân tán:** DDPM xuất sắc trong việc tạo ra dữ liệu bằng cách học đảo ngược một quá trình thêm nhiễu dần dần. Quá trình này cho phép mô hình dần dần tinh chỉnh một mẫu, bắt đầu từ nhiễu thuần túy và tiến đến một điểm dữ liệu thực tế.

- **Mối liên hệ với việc khớp điểm số:** Việc chọn dự đoán nhiễu ( $\epsilon$ ) trong quá trình ngược tạo ra một mối liên kết mạnh mẽ với việc khớp điểm số. Điều này giải thích chất lượng mẫu cao mà DDPM đạt được và gợi ý rằng chúng học một đại diện ngầm của phân phối dữ liệu thông qua hàm điểm của nó.
- **Giải mã tiến triển:** DDPM có một quá trình tạo ra tự nhiên từ thô đến tinh, lộ ra các tính năng quy mô lớn trước và dần dần thêm chi tiết. Hiểu biết này có ý nghĩa đối với việc hiểu trao đổi tỷ lệ-méo của mô hình và mối liên hệ của nó với nén mất mát.
- **Mở rộng các mô hình tự hồi quy:** DDPM có thể được coi là mở rộng ý tưởng tự hồi quy đối với dữ liệu liên tục và đưa ra một "sắp xếp bit" linh hoạt hơn thông qua quá trình nhiễu Gaussian.

Các mô hình phân tán là một lớp mô hình sinh mạnh mẽ và đa năng, đạt được kết quả ấn tượng trong việc tạo hình ảnh. Mối liên hệ của chúng với việc khớp điểm số, giải mã tiến triển và các mô hình tự hồi quy mang đến những hiểu biết quý giá về cơ chế hoạt động của chúng và cung cấp nền tảng cho việc khám phá và phát triển thêm.