Report paper: Denoising Diffusion Probabilistic Models

Giáp Ngọc Hiệu

**Tóm tắt nội dung**

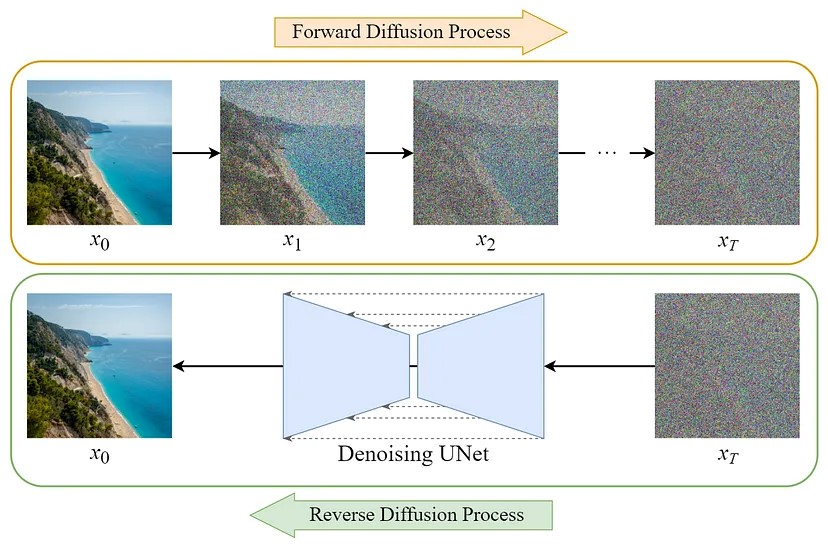
Bài report này là phần tóm tắt lại ý tưởng, nội dung chính, những điều làm được của paper

Denoising Diffusion Probabilistic Models

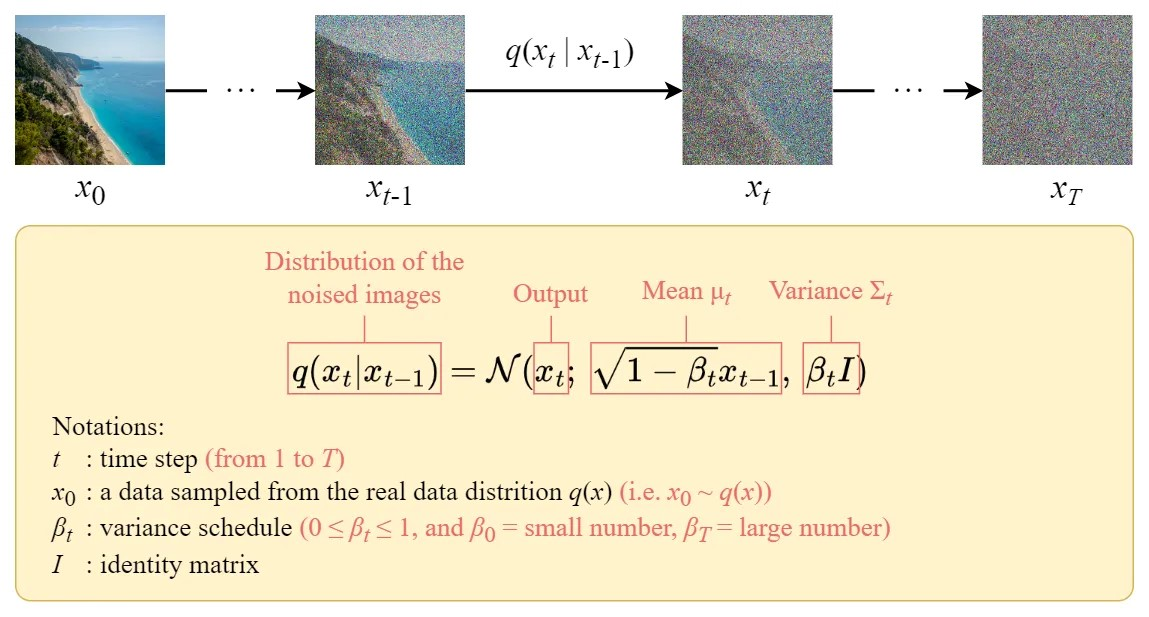
1. **Ý tưởng**

* Tác giả đề xuất và sử dụng diffusion probabilistic models để tạo ra các hình ảnh tổng hợp chất lượng cao. Diffusion probabilistic models một loại latent variable model lấy cảm hứng từ nhiệt động lực học không cân bằng.
* Trong vật lý, khuếch tán chỉ đơn giản là chuyển động tổng thể của bất cứ thứ gì ( nguyên tử, năng lượng ) từ vùng có nồng độ cao hơn đến vùng có nồng độ thấp hơn. Bây giờ hãy tưởng tượng thả một giọt sơn nhỏ vào một cốc nước, mật độ của sơn sẽ được tập trung ở một vị trí, nhưng khi thời gian trôi qua, giọt sơn sẽ khuếch tán vào nước cho đến khi nó đạt đến trạng thái cân bằng. Bây giờ sẽ rất tuyệt nếu chúng ta có thể đảo ngược quá trình này? Thật không may, điều này là không thể. Nhưng các mô hình khuếch tán cố gắng thực hiện với một mô hình với mục tiêu cuối cùng là đảo ngược quá trình này.
* Ý tưởng chính đằng sau diffusion models là phá hủy cấu trúc một cách có hệ thống và từ từ trong phân phối dữ liệu thông qua quá trình khuếch tán chuyển tiếp lặp đi lặp lại. Sau đó, tìm hiểu một quá trình khuếch tán ngược phục hồi cấu trúc trong dữ liệu, mang lại một mô hình tổng quát rất linh hoạt và dễ điều khiển của dữ liệu.

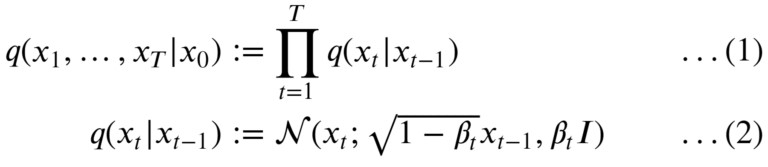
1. **Phương pháp**
   1. **Introduction**



* Diffusion probabilistic model (diffusion models) là một chuỗi Markov được tham số hóa được huấn luyện bằng cách sử dụng variational inference để tạo ra các mẫu khớp với dữ liệu sau thời gian hữu hạn.
* Các diffusion models gắng tái tạo quá trình khuếch tán bằng cách lặp lại việc thêm nhiễu vào hình ảnh gốc cho đến khi hình ảnh trở thành noise thuần túy.
* Noise được xác định bởi một chuỗi các sự kiện Markov. Chuỗi Markov là một mô hình của các sự kiện trong đó mỗi bước thời gian chỉ phụ thuộc vào bước thời gian trước đó.
* Sau đó đào tạo một mạng lưới thần kinh để loại bỏ noise đó.
  1. **Forward Diffusion Process**



* Phân phối q trong quá trình khuếch tán thuận được định nghĩa là Chuỗi Markov được đưa ra bởi:



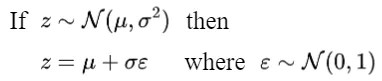
* 𝝱 (beta) được gọi là “tốc độ khuếch tán” và được tính toán trước bằng cách sử dụng “bộ lập lịch phương sai”. Thuật ngữ I là một identity matrix. Do đó, phân bố ở mỗi bước thời gian được gọi là Gaussian đẳng hướng.
* Forward Diffusion Process dần dần thêm noise Gaussian vào hình ảnh đầu vào

x ₀ từng bước một, và sẽ có tổng T các bước. Quá trình này sẽ tạo ra một chuỗi các mẫu hình ảnh noise x ₁, …, x\_T.

* Khi T → ∞, kết quả cuối cùng sẽ trở thành một hình ảnh hoàn toàn noise như thể nó được lấy mẫu từ một phân phối Gaussian đẳng hướng.
* Nhưng thay vì thiết kế một thuật toán để lặp lại thêm nhiễu cho hình ảnh, chúng ta có thể sử dụng công thức closed-form để lấy mẫu trực tiếp hình ảnh noise ở bước thời gian cụ thể t.

**Closed-Form Formula**

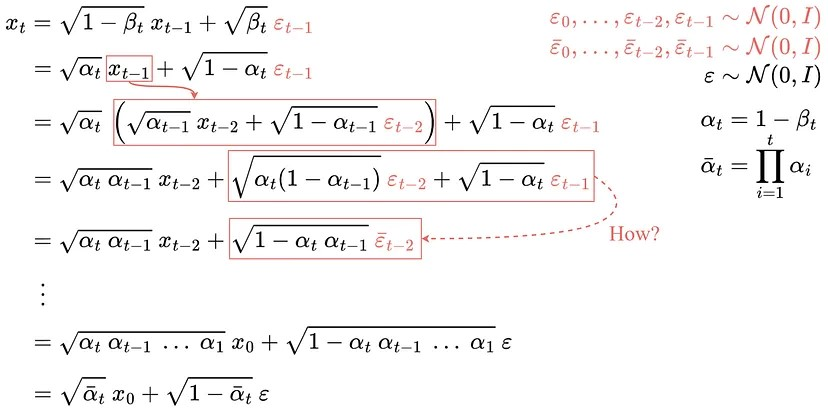
Closed-form sampling formula có thể được rút ra bằng cách sử dụng Reparameterization Trick.



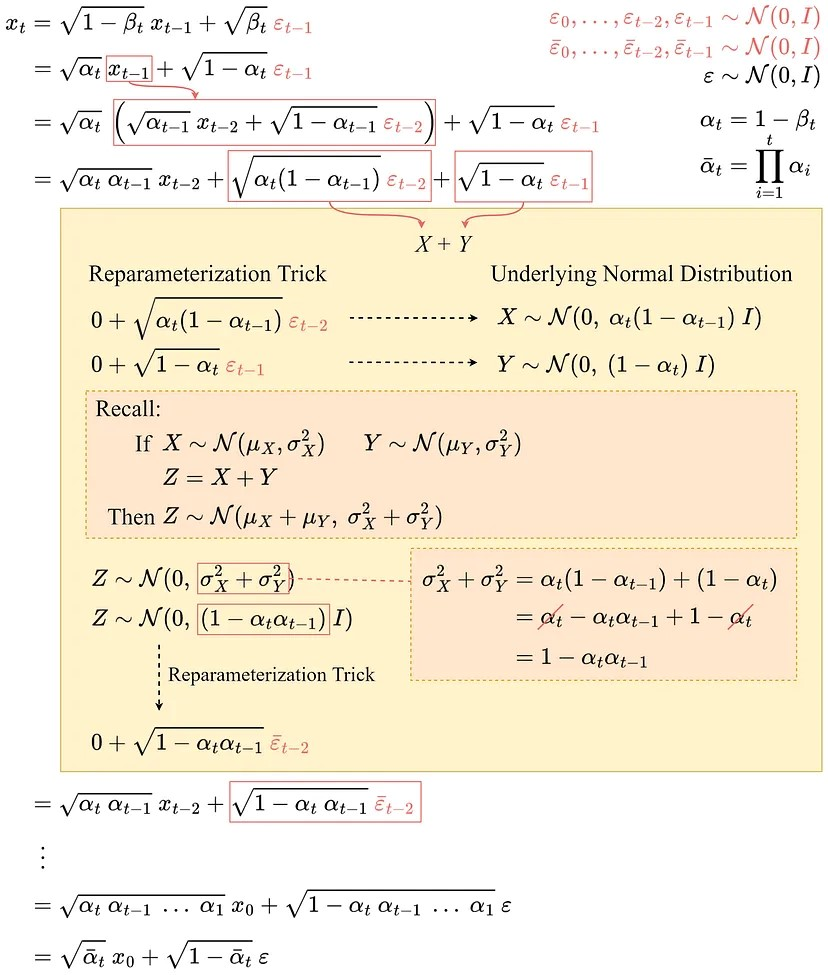
Với cách này, chúng ta có thể biểu thị hình ảnh được lấy mẫu xₜ như sau:



Sau đó, chúng ta có thể mở rộng nó một cách đệ quy để có được closed-form formula:



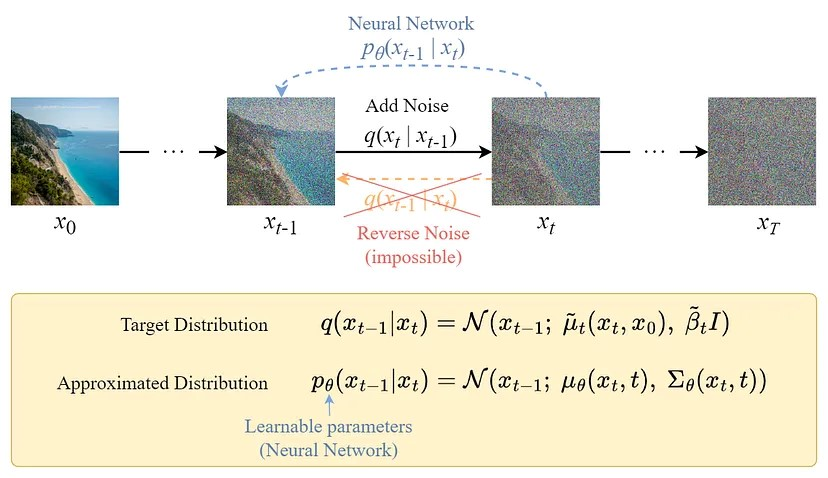
* Tất cả ε đều độc lập và có phân phối giống hệt các biến ngẫu nhiên chuẩn hóa tiêu chuẩn.
* Điều quan trọng là phải phân biệt chúng bằng các ký hiệu và chỉ mục khác nhau vì chúng độc lập và giá trị của chúng có thể khác nhau sau khi lấy mẫu.



* X và Y thể được coi là mẫu từ hai phân phối chuẩn khác nhau.
* vd. X ~ N ( 0, αt ( 1-αt ₋ ₁ ) I ) và Y ~ N ( 0, ( 1-αt ) I ).
* Tổng của hai biến ngẫu nhiên ( độc lập ) cũng được phân phối chuẩn.
* vd. nếu Z = X + Y, sau đó Z ~ N ( 0, σ ²x+ σ ²γ).
* Do đó, chúng ta có thể hợp nhất chúng lại với nhau và thể hiện phân phối chuẩn được hợp nhất ở dạng reparameterized form.
* Lặp lại các bước này sẽ cho ra công thức sau chỉ phụ thuộc vào hình ảnh đầu vào x ₀:



* Giờ đây, có thể lấy mẫu trực tiếp xₜ ở bất kỳ bước nào bằng cách sử dụng công thức này và điều này giúp Forward Diffusion Process nhanh hơn nhiều.
  1. **Reverse Diffusion Process**



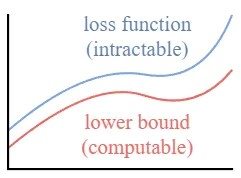
* Không giống như quy trình chuyển tiếp, chúng ta không thể sử dụng q(xₜ₋₁|xₜ) để đảo ngược nhiễu vì nó không thể xử lý được (không thể tính toán được).
* Vì vậy, chúng ta cần huấn luyện một neural network pθ(xₜ₋₁|xₜ) để xấp xỉ q(xₜ₋₁|xₜ). Giá trị gần đúng pθ(xₜ₋₁|xₜ) tuân theo phân phối chuẩn và giá trị trung bình cũng như phương sai của nó được đặt như sau:



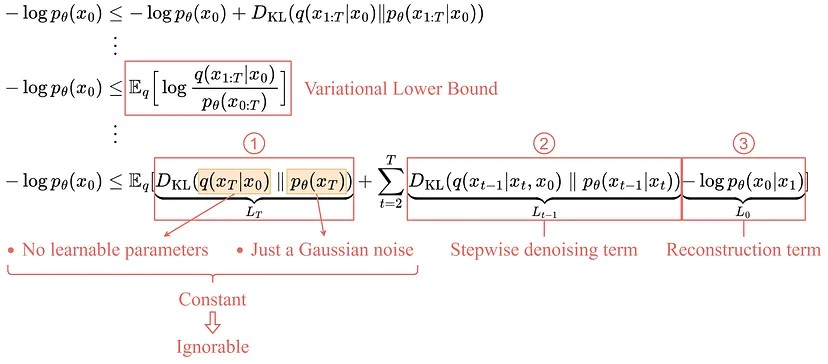
* 1. **Loss Function**
* Chúng ta có thể định nghĩa hàm loss là Negative Log-Likelihood:



* Thiết lập này rất giống với thiết lập trong VAE, thay vì tối ưu hóa chính hàm loss khó xử lí này, chúng ta có thể tối ưu hóa Variational Lower Bound.



* Bằng cách tối ưu hóa giới hạn dưới có thể tính toán được, chúng ta có thể gián tiếp tối ưu hóa hàm mất mát khó xử lí.

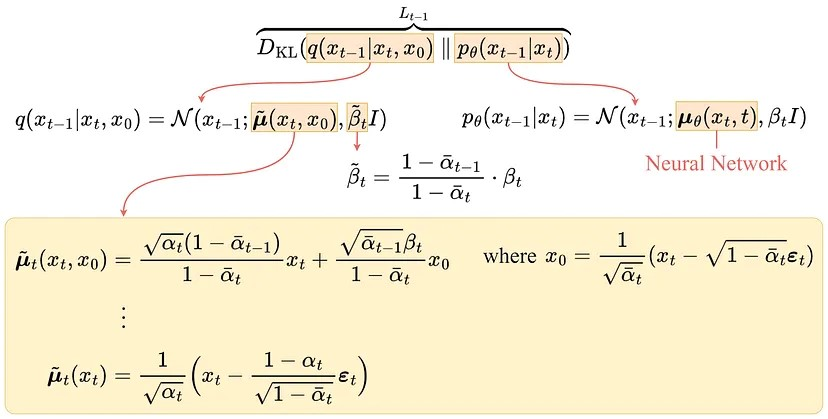


1. **L\_T**: **Constant term**

* Vì q không có tham số có thể học được và p chỉ là Gaussian noise probability nên thuật ngữ này sẽ là hằng số trong quá trình huấn luyện và do đó có thể bị bỏ qua.

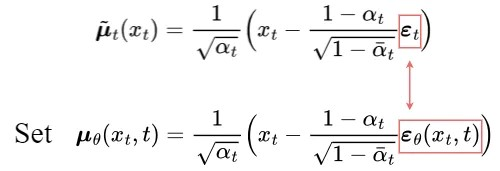
1. **Lₜ₋₁: Stepwise denoising term**

* Thuật ngữ này so sánh bước khử nhiễu mục tiêu q và bước khử nhiễu gần đúng pθ.
* Lưu ý rằng bằng cách điều hòa x₀, q(xₜ₋₁|xₜ, x₀) trở nên dễ điều khiển.

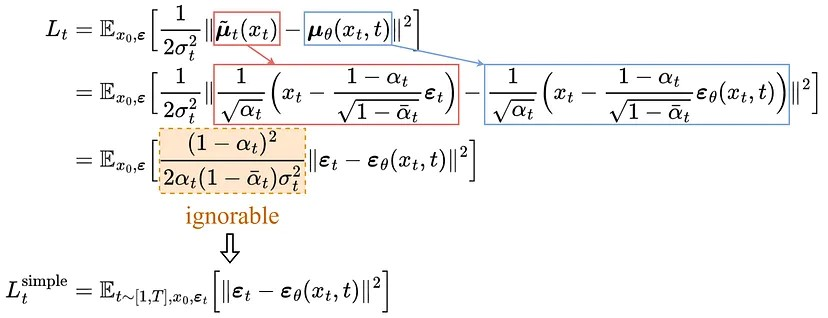




* Để ước tính bước khử nhiễu mục tiêu q, chúng ta chỉ cần ước tính giá trị trung bình của nó bằng cách sử dụng neural network. Vì vậy, ta đặt giá trị trung bình gần đúng μθ có cùng dạng với giá trị trung bình mục tiêu μ̃ₜ (với neural network có thể học được εθ):



* Việc so sánh giữa target mean và approximated mean có thể được thực hiện bằng cách sử dụng MSE:



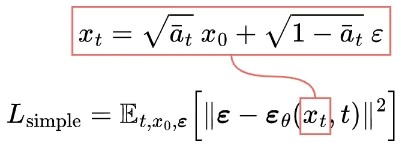
* Về mặt thực nghiệm, có thể đạt được kết quả tốt hơn bằng cách bỏ qua weighting term và chỉ cần so sánh target và predicted noises với MSE.

1. **L₀: Reconstruction term**

* Đây là reconstruction loss của bước khử nhiễu cuối cùng và nó có thể bị bỏ qua trong quá trình huấn luyện vì những lý do sau:
  + Nó có thể được ước tính gần đúng bằng cách sử dụng cùng một neural network trong Lₜ₋₁.
  + Bỏ qua nó sẽ làm cho chất lượng mẫu tốt hơn và làm cho việc thực hiện đơn giản hơn.

**Simplified Loss**

* Vì vậy, mục tiêu đào tạo đơn giản hóa cuối cùng như sau:

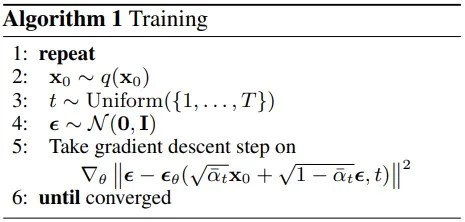


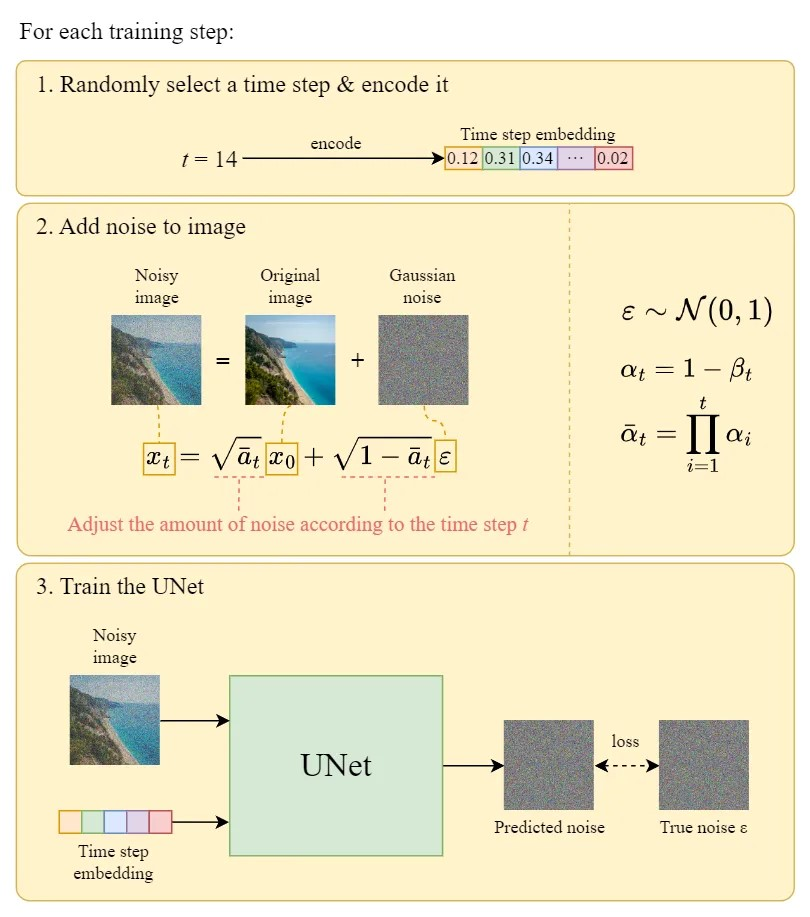
* Tác giả thấy rằng việc đào tạo các mô hình về variational bound thực sự mang lại độ dài mã tốt hơn so với đào tạo về simplified objective, như mong đợi, nhưng mục tiêu sau mang lại chất lượng mẫu tốt nhất
  1. **The U-Net Model**

**2.5.1 Dataset**

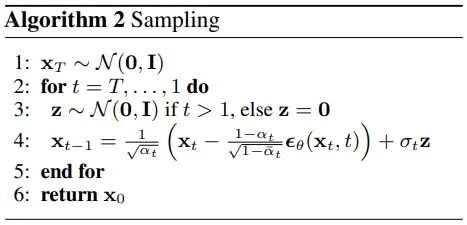
* Trong mỗi epoch:
  + Time step ngẫu nhiên t sẽ được chọn cho mỗi training sample (image).
  + Áp dụng Gaussian noise (tương ứng với t) cho mỗi hình ảnh.
  + Convert time steps to embeddings (vectors).

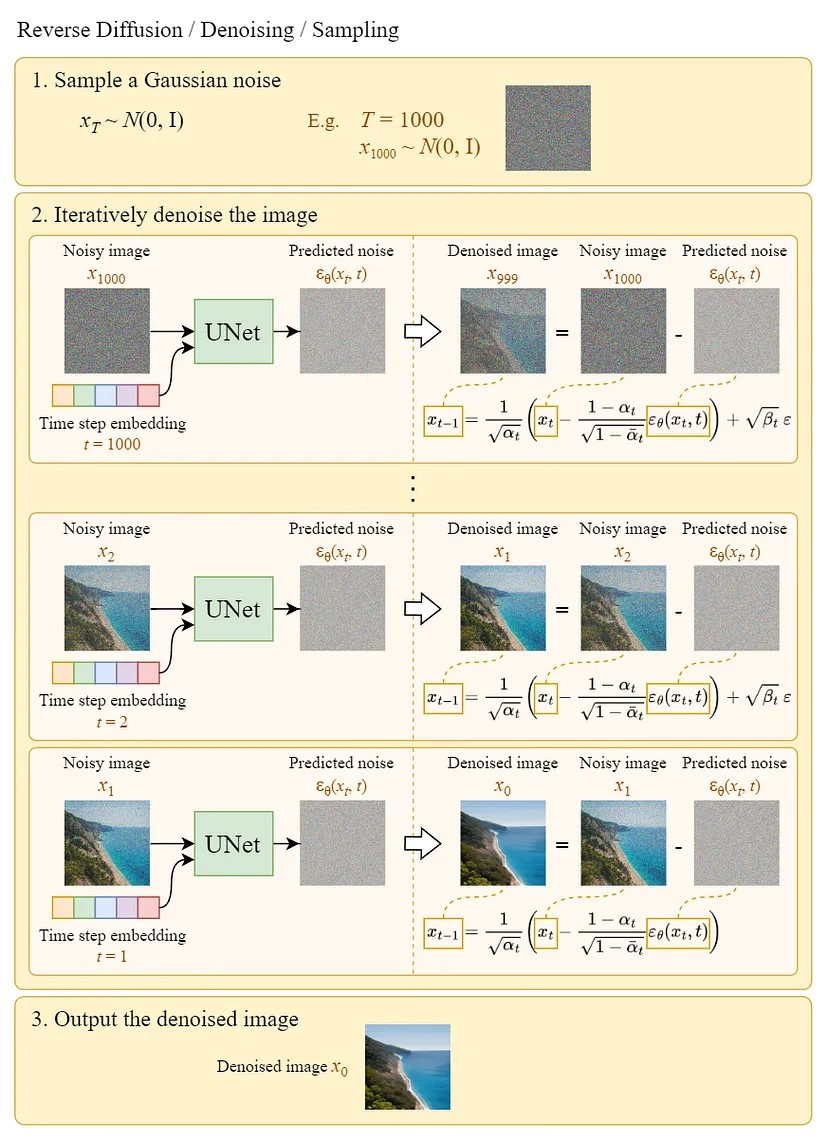
**2.5.2 Training**





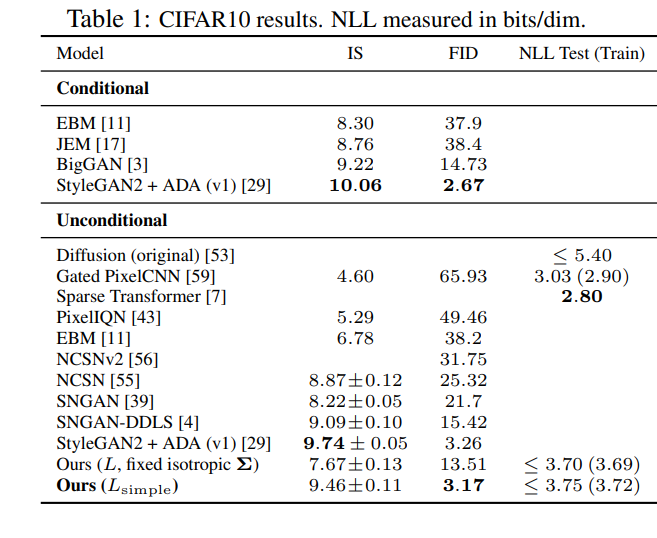
**2.5.3 Reverse Diffusion**





* Lưu ý rằng ở bước cuối cùng, chúng ta chỉ cần xuất learned mean μθ(x₁, 1) mà không thêm noise vào nó**.**

1. **Experimental & Results**
   1. **Sample quality**

****

* Với FID score là 3,17, mô hình vô điều kiện đạt được chất lượng mẫu tốt hơn hầu hết các mô hình khác, bao gồm cả mô hình có điều kiện.

1. **Note**

* Ưu điểm:
  + Khả năng tạo ra hình ảnh có độ phân giải cao, đa dạng và chi tiết.
  + Có thể chống lại tình trạng overfitting do quá trình khử nhiễu được sử dụng trong đào tạo..
  + Linh hoạt, có thể ứng dụng trên nhiều lĩnh vực khác nhau.
  + Có thể tìm hiểu các patterns và mối quan hệ phức tạp trong dữ liệu đầu vào.
* Nhược điểm:
  + Thời gian đào tạo và tạo ra kết quả dài hơn, khiến chúng tốn kém về mặt tính toán.
  + Khó khăn trong việc xử lý dữ liệu dựa trên văn bản so với dữ liệu hình ảnh.
  + Dễ tạo ra hình ảnh không thực tế nếu thuật toán khử nhiễu không thành công.
* Ứng dụng:
  + Tạo và xử lý hình ảnh.
  + Tổng hợp kết cấu cho môi trường ảo và trò chơi điện tử.
  + Style transfer và sáng tạo nội dung cho nghệ thuật và thiết kế.