

HUST

ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

ONE LOVE. ONE FUTURE.



ĐẠI HỌC
BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Conditional Diffusion Models

ONE LOVE. ONE FUTURE.

Nội dung chính

- Conditional Diffusion Models
 - Conditioning Mechanisms
 - Guidance Methods
 - Adapters/Plugins
- Stable Diffusion
 - SD1.x, SD2.x, SDXL
 - Rectified Flow, SD3, SD3.5
- FLUX: Flow Matching, FLUX.1, FLUX.2
- Hunyuan-DiT, Kandinsky
- Rectified Diffusion
- Ứng dụng trong bài toán thực tế



Conditional Diffusion Models

- Unconditional Diffusion Model: Học phân phối xác suất $P(x)$ của dữ liệu
 - Mục tiêu: Sinh ra một mẫu x bất kỳ thuộc tập dữ liệu
 - Hàm dự đoán: Nhiễu $\varepsilon(x_t, t)$
- Conditional Diffusion Model: Học phân phối xác suất có điều kiện $P(x|y)$
 - Mục tiêu: Sinh ra một mẫu x khớp với điều kiện y nào đó
 - Hàm dự đoán: Nhiễu $\varepsilon(x_t, t, y)$



Conditional Diffusion Models

- Các dạng Conditional phổ biến:
 - Class Conditional: Model sinh ảnh theo nhãn/lớp
 - Text-to-Image (T2I): Model nhận vào text và sinh ảnh từ nhiễu khớp với mô tả
 - Image-to-Image (I2I): Model nhận vào ảnh và text mà model dựa vào để điều chỉnh ảnh

cat → dog



Conditioning Mechanisms

- Conditioning Mechanisms là thuật ngữ kỹ thuật chỉ các phương pháp cụ thể để đưa thông tin điều kiện (văn bản, ảnh, nhãn,...) vào bên trong mạng nhằm kiểm soát quá trình sinh ảnh
- Các cơ chế chính được sử dụng:
 - Adaptive Normalization
 - Cross-Attention



Adaptive Normalization

- AdaNorm lần đầu được đưa vào Diffusion tháng 5/2021 để xử lý thông tin Class Label và Time step:

$$x_{norm} = \frac{x - \mu}{\sigma}$$

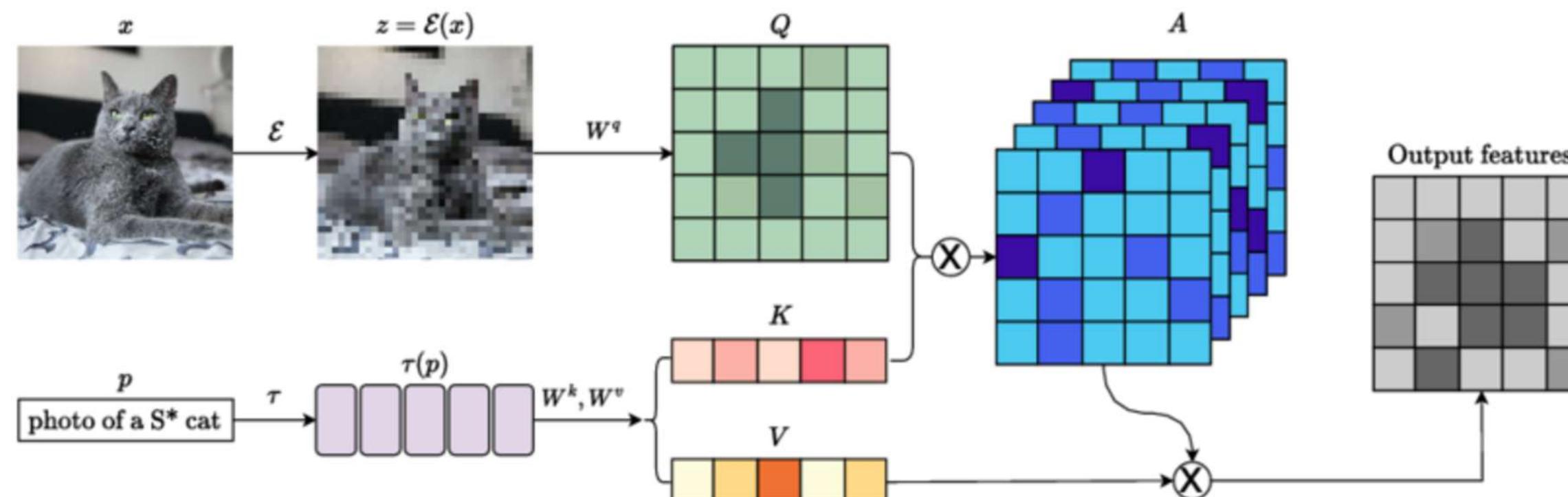
$$output = \gamma(y, t)x_{norm} + \beta(y, t)$$

- Điều kiện được vector hóa, có thể cộng hoặc concat với vector embedding của step t, đi qua một hoặc nhiều lớp Linear thành giá trị scale và shift của lớp Normalization



Cross-Attention

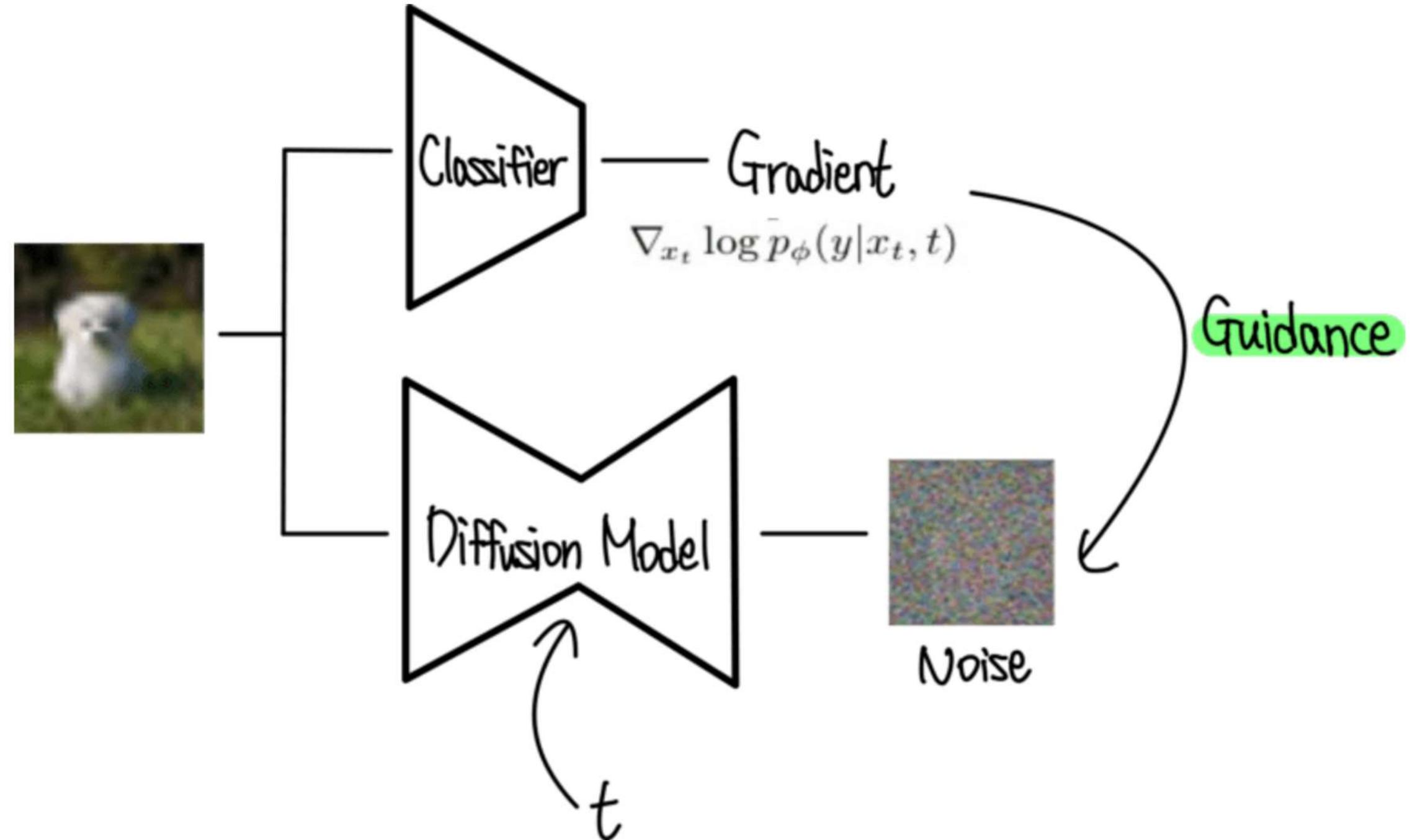
- Cross-Attention được sử dụng trong Diffusion lần đầu vào tháng 12/2021 cùng với cơ chế Self-Attention, trong đó:
 - Self-Attention: Ảnh tự nhìn chính nó, pixel này nhìn các pixel khác để hiểu ngữ cảnh
 - Cross-Attention: Ảnh nhìn sang một dữ liệu khác (Text), pixel nhìn vào các từ ngữ để biết nó cần phải hiển thị nội dung gì



- Conditioning Mechanisms là cơ chế thuộc về kiến trúc mạng, giúp đưa thông tin điều kiện vào trong quá trình huấn luyện để dự đoán nhiễu
- Tuy nhiên trong quá trình sampling, mô hình có xu hướng “trung bình hóa” bởi kiến thức nền, lờ đi các chi tiết cụ thể trong prompt
- Guidance Methods buộc mô hình di chuyển về phía những vùng khớp với prompt nhất

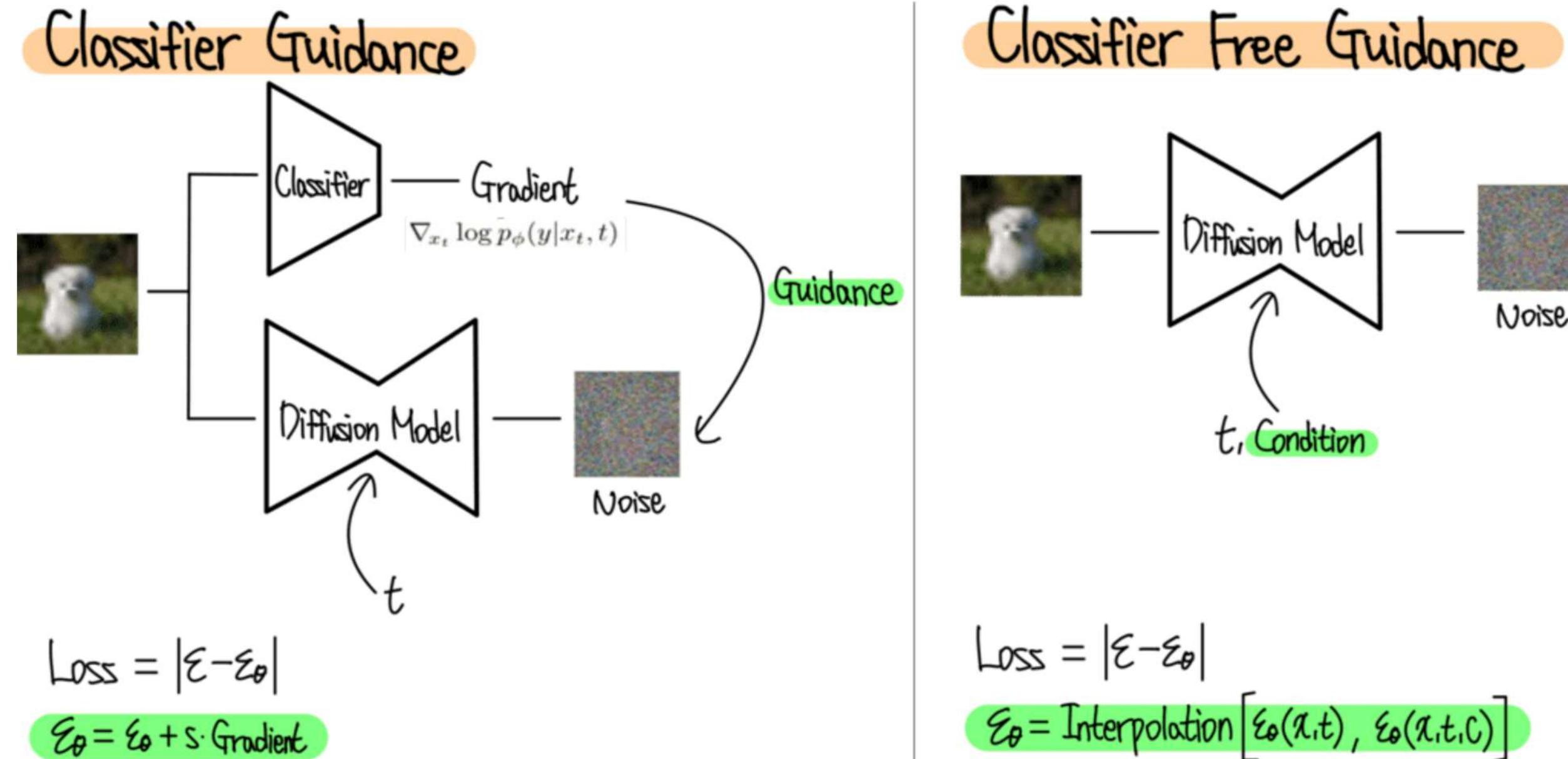
Classifier Guidance

- Classifier Guidance
ra đời tháng
5/2021 dựa trên
việc kết hợp hai
mô hình riêng biệt
hoạt động cùng lúc
trong quá trình
sinh ảnh



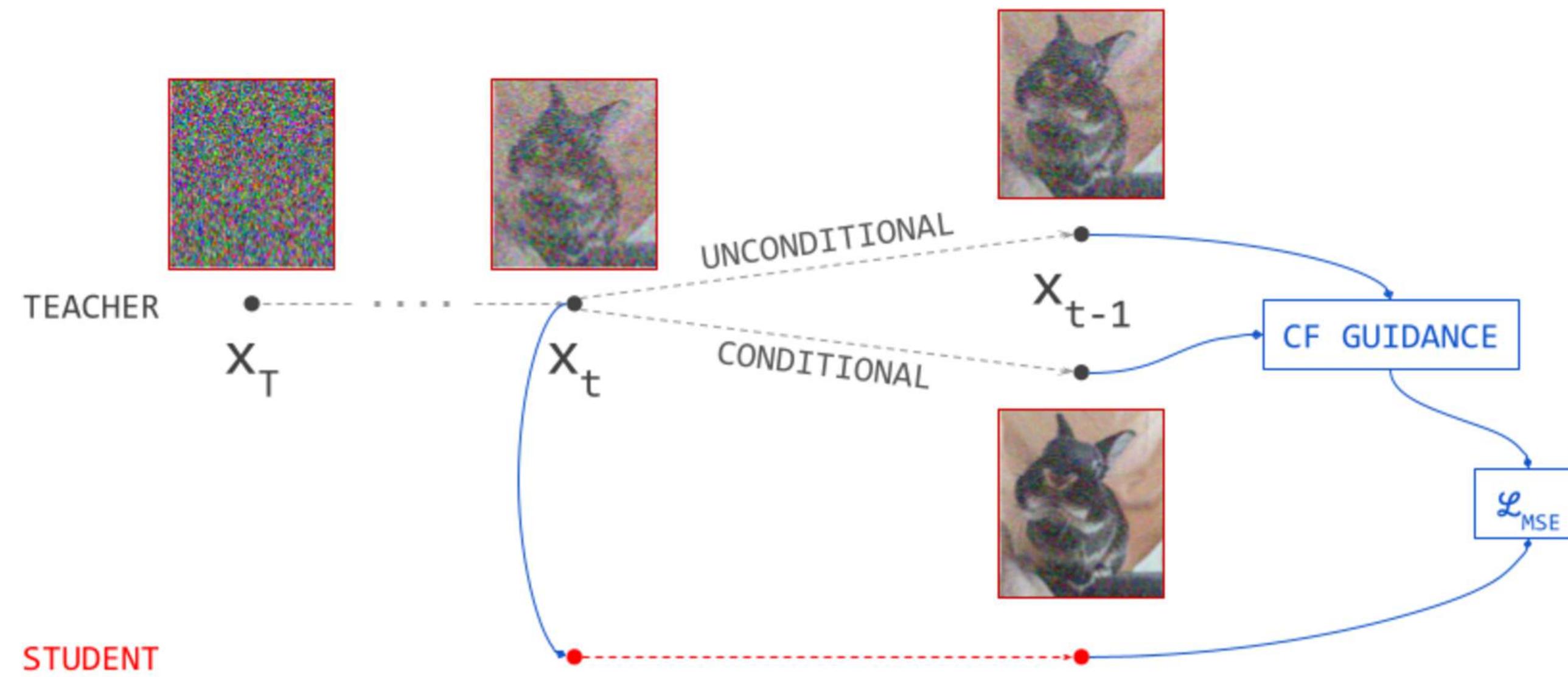
Classifier Free Guidance

- Giới thiệu lần đầu tháng 7/2022, Classifier Free Guidance xóa bỏ mô hình Classifier và để mô hình Diffusion tự biên tự diễn



Guidance Distillation

- Guidance Distillation ra đời tháng 10/2022 khắc phục hạn chế tốc độ của CFG bằng cơ chế Teacher-Student



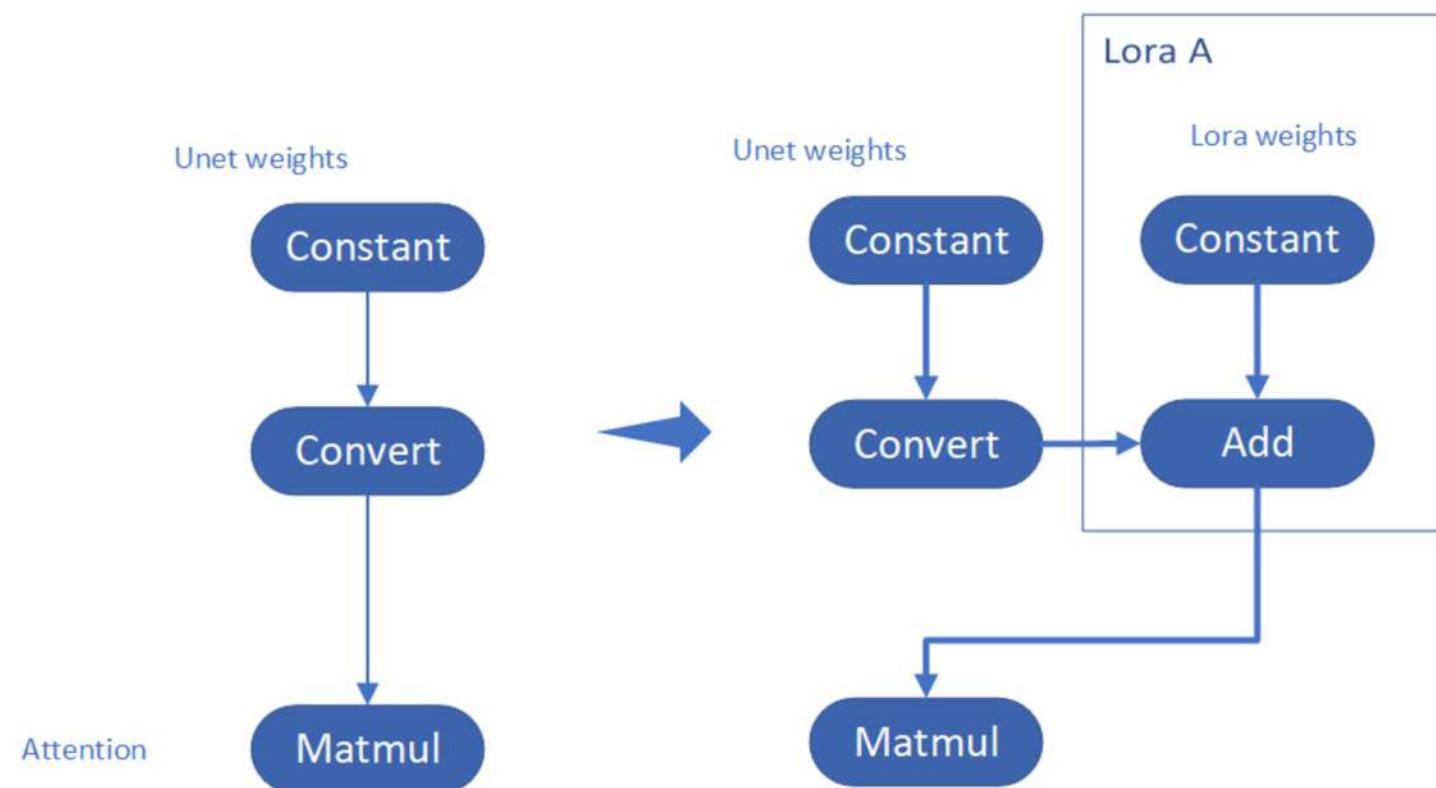
Adapters/Plugins

- Adapters (Bộ chuyển đổi) là các mô hình không thay đổi cấu trúc cốt lõi của mô hình Diffusion gốc mà chỉ "ký sinh" hoặc "gắn kèm" vào để thay đổi hành vi của mô hình trong quá trình Sampling
- Adapters chia làm 3 nhóm đại diện chính:
 - Can thiệp vào Trọng số (Weight Injection)
 - Can thiệp vào Luồng dữ liệu (Feature Injection)
 - Can thiệp vào Cơ chế Chú ý (Attention Injection)



- LoRA (Low-Rank Adaptation) được giới thiệu tháng 6/2021 vốn dùng cho LLM, đến cuối năm 2022 mới được sử dụng cho Diffusion
- Các mô hình Diffusion hiện đại có hàng tỷ tham số. Việc đào tạo lại với một bộ dữ liệu mới là không khả thi:
 - Tốn kém: Cần lượng VRAM khổng lồ để lưu trữ gradient của tất cả tham số
 - Dư thừa: Khi mô hình học một khái niệm mới, sự thay đổi thực sự của các trọng số diễn ra ở một không gian có chiều thấp

- LoRA: Thay vì cập nhật toàn bộ ma trận W thì học sự thay đổi bằng việc phân rã ma trận



$$h = Wx + \Delta Wx = Wx + (B \cdot A)x$$

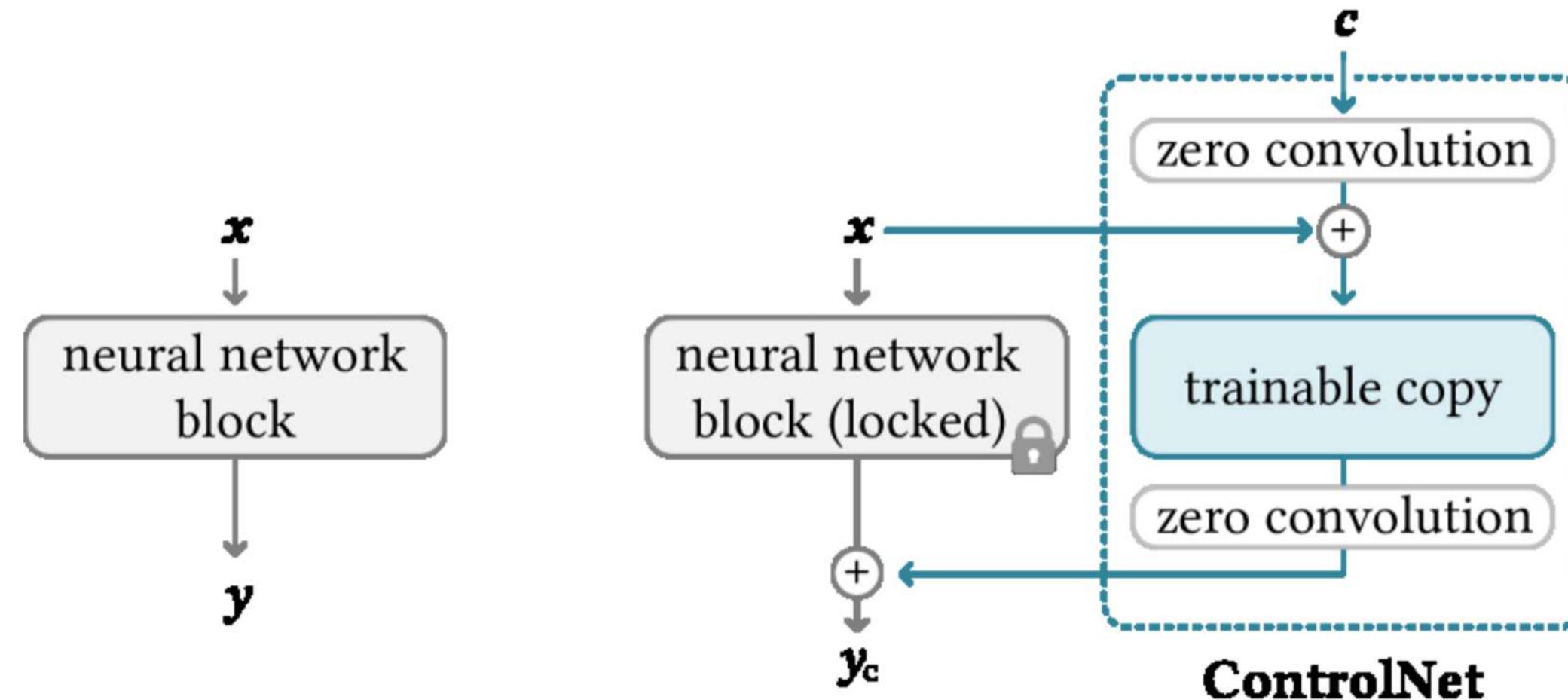
$$W \in R^{m \times n}, A \in R^{r \times n}, B \in R^{m \times r}$$

- Trong đó r là rank của ma trận (nhỏ hơn nhiều so với m, n)

- DoRA (Tháng 2/2024) cải tiến LoRA bằng áp dụng công thức phân rã trọng số:

$$W = m \frac{V}{\|V\|}$$

- ControlNet (Tháng 2/2023) giúp kiểm soát cấu trúc ảnh mô hình sinh ra một cách tuyệt đối với kiến trúc Zero Convolution
- ControlNet giải quyết vấn đề: Làm sao thêm điều kiện đầu vào (như nét vẽ, pose) mà không phải train lại mô hình từ đầu?



- ControlNet cần đầu vào là một Tensor không gian hay có tính tương quan về không gian, đảm bảo trả lời một câu hỏi:
"Tại vị trí pixel (x, y) này có cái gì?"
- Tuy nhiên nguồn tạo ra đầu vào đó không nhất thiết phải là "ảnh". Các dữ liệu đầu vào thực tế mà ControlNet có thể xử lý gồm:
 - Dạng hình học: bản đồ các đường biên, nét vẽ, độ sâu (cho 3D)
 - Dạng dữ liệu trừu tượng: segmentation map, human pose
 - Dạng dữ liệu "Phi hình ảnh":
 - Audio: Biến file âm thanh thành hình ảnh Spectrogram (biểu đồ tần số theo thời gian)
 - Text (Typography)

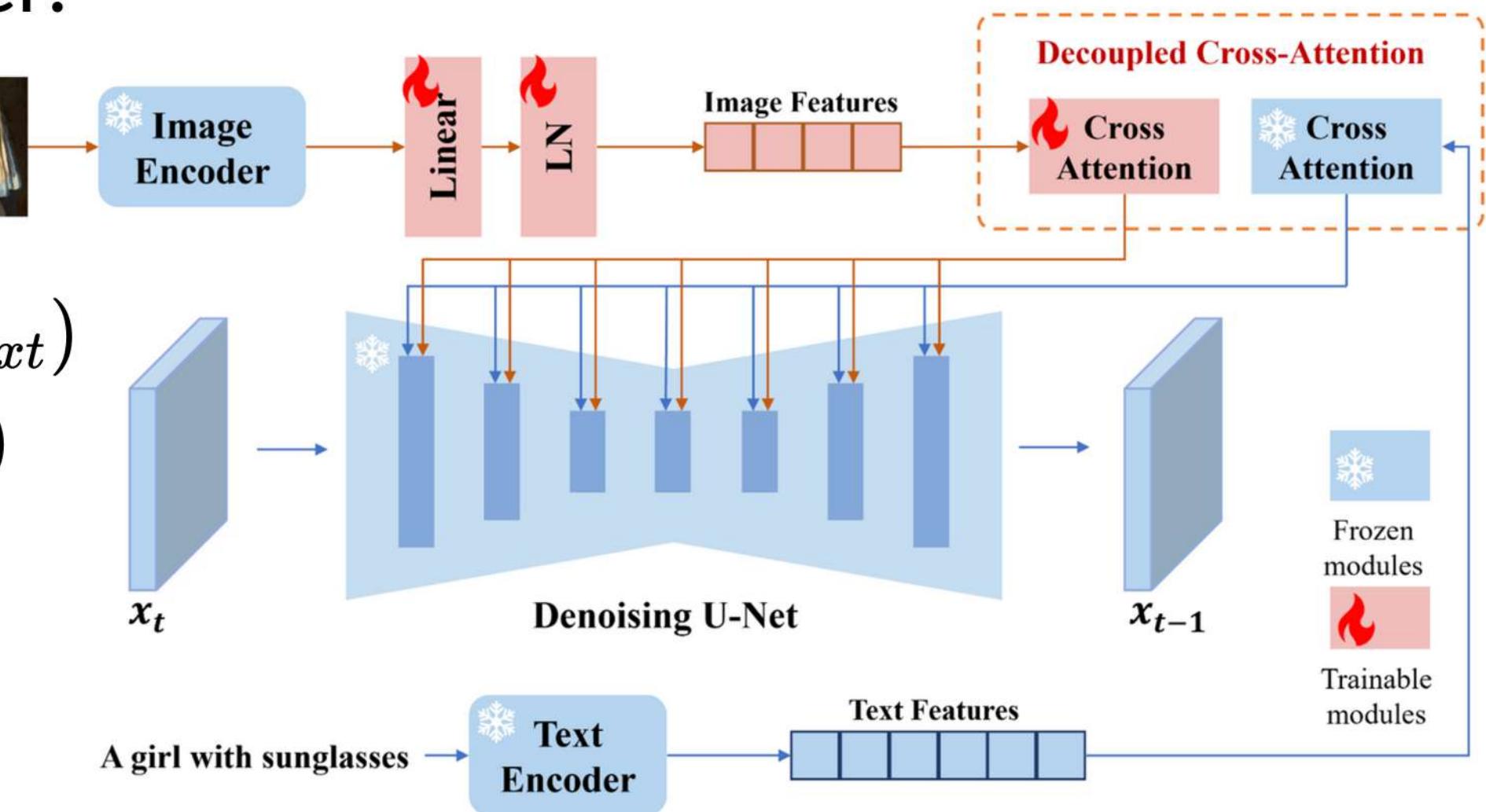
IP-Adapter

- IP-Adapter (Image Prompt Adapter) ra mắt tháng 8/2023
- Cross-Attention: $Z = \text{Attention}(Q, K_{text}, V_{text})$
- Cross-Attention + IP Adapter:

$$Z_{text} = \text{Attention}(Q, K_{text}, V_{text})$$

$$Z_{img} = \text{Attention}(Q, K_{img}, V_{img})$$

$$Z = Z_{text} + \lambda Z_{img}$$



- Cơ chế Decoupled Cross-Attention:

- Bảo toàn kiến thức gốc: Attention Text tính riêng biệt, không bị nhiễu bởi dữ liệu ảnh. Ảnh chỉ đóng vai trò là "tín hiệu bổ sung"
- Hiệu quả Fine-tuning: Khi train IP-Adapter, chỉ train K_{img} và V_{img} , hội tụ nhanh hơn so với việc phải "dạy" lại hàm Softmax cách chia sẻ sự chú ý giữa Text và Ảnh nếu concat cả hai vào làm một
- Linh hoạt: Vì chỉ cộng tuyến tính $Z = Z_{text} + \lambda Z_{img}$, có thể dễ dàng thay đổi ảnh hưởng của ảnh mà không làm ảnh hưởng đến Text, nếu thực hiện concat thì việc thay đổi sẽ khó hơn

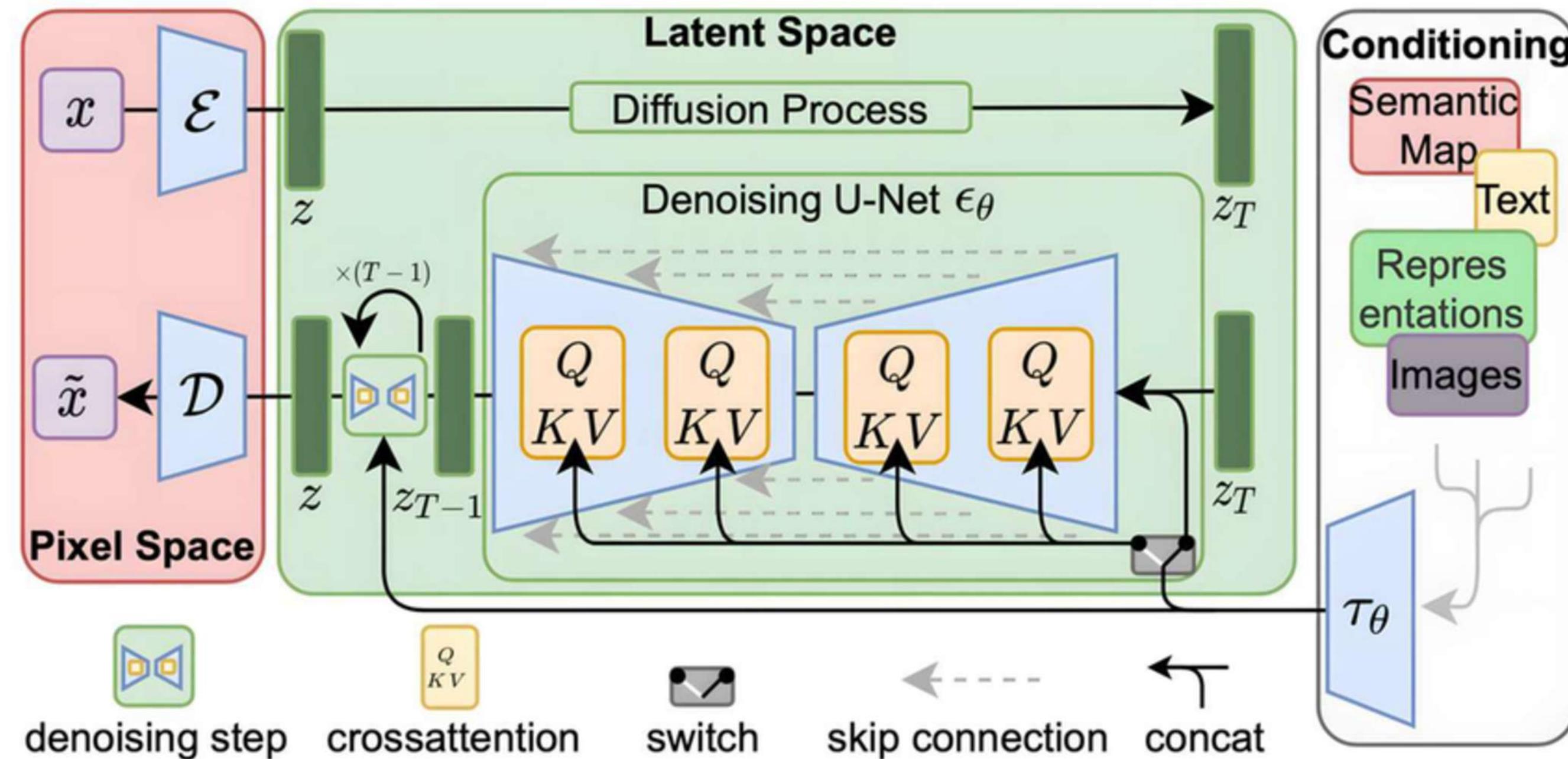
Conditional Diffusion Models

- Conditional Diffusion Models chia làm hai phần là Open Source và Closed Source
- Nhánh Open Source:
 - Stable Diffusion
 - FLUX
 - Hunyuan-DiT, Kandinsky
- Nhánh Close Source:
 - Midjourney
 - DALL-E
 - Imagen



Stable Diffusion

- Stable Diffusion là một mô hình Latent Diffusion Model (LDM)



- SD1.4 (Tháng 8/2022):
 - Backbone: U-Net (860 triệu tham số)
 - Text encoder: CLIP ViT-L/14 của OpenAI
 - Autoencoder: VAE với hệ số nén $f=8$ (Ảnh 512x512 \rightarrow 64x64)
- SD1.5 (Tháng 10/2022): Mô hình được huấn luyện lâu hơn SD1.4 gấp 2.5 lần
- SD2.0, SD2.1 (Tháng 11-12/2022):
 - Text Encoder: OpenCLIP ViT-H/14
 - Prediction: Chuyển từ dự đoán nhiễu sang dự đoán vận tốc

Velocity Prediction

- Việc tham số hóa một biến mới cho huấn luyện là vận tốc được đề xuất vào tháng 2/2022, khi nhìn Forward Process dưới góc nhìn lượng giác:

$$\begin{aligned}x_t &= \sqrt{\alpha_t}x_0 + \sqrt{1 - \alpha_t}\varepsilon, \varepsilon \sim N(0, 1) \\&= \sin(\phi_t)x_0 + \cos(\phi_t)\varepsilon\end{aligned}$$

- Vận tốc là đạo hàm của vị trí theo thời gian (hoặc tham số chuyển động):

$$v_t = \frac{dx_t}{d\phi_t} = \sqrt{\alpha_t}\varepsilon - \sqrt{1 - \alpha_t}x_0$$



- Phương trình khuếch tán:

$$x_t = \sqrt{\alpha_t}x_0 + \sqrt{1 - \alpha_t}\varepsilon \Leftrightarrow x_0 = \frac{x_t - \sqrt{1 - \alpha_t}\varepsilon}{\sqrt{\alpha_t}}$$

- Ở những step lớn $t \rightarrow T, \alpha_t \rightarrow 0$:

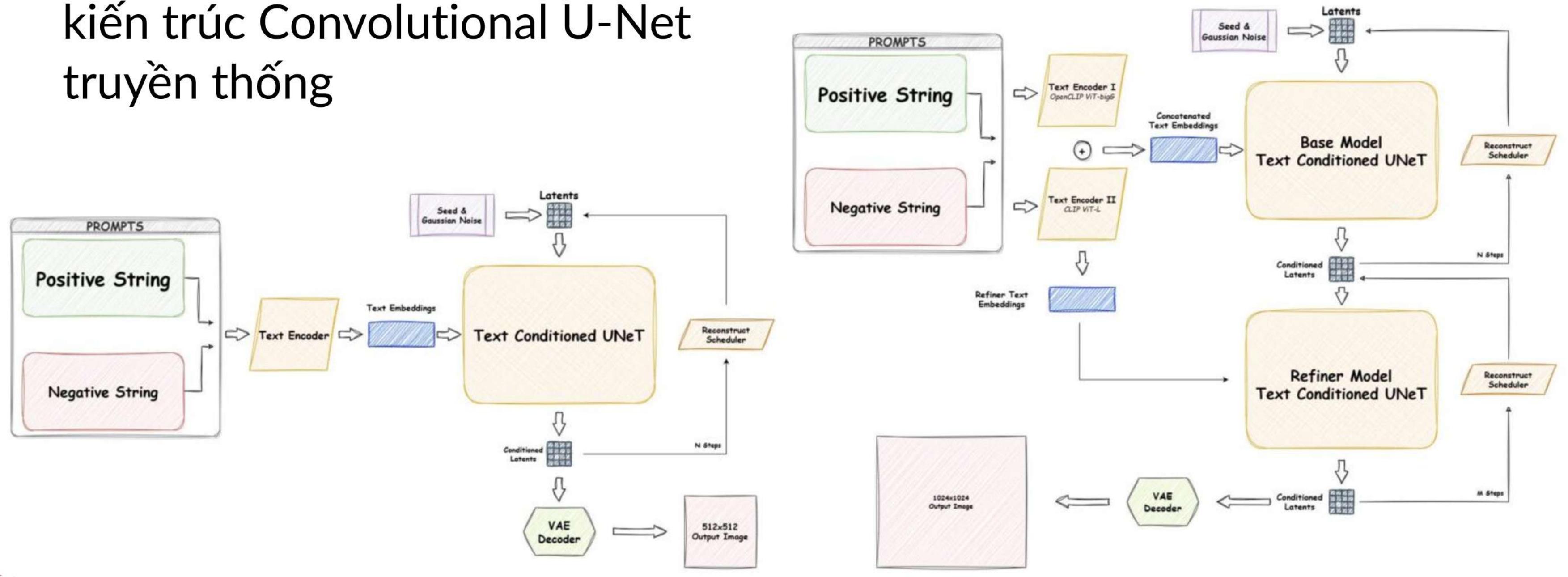
- Với mô hình dự đoán nhiễu: $x_0 = \frac{x_t - \varepsilon}{0} \rightarrow \infty$
- Với mô hình dự đoán vận tốc: $x_0 = 0.x_t - 1.v_t = -v_t$

Velocity Prediction

- Trong quá trình Sampling, khi model trả về vận tốc dự đoán, có thể suy ra cả ảnh gốc dự đoán và nhiễu dự đoán:

$$\begin{cases} x_t = \sqrt{\alpha_t}x_0 + \sqrt{1 - \alpha_t}\varepsilon \\ v_t = \sqrt{\alpha_t}\varepsilon - \sqrt{1 - \alpha_t}x_0 \end{cases}$$
$$\Rightarrow \begin{cases} x_0 = \sqrt{\alpha_t}x_t - \sqrt{1 - \alpha_t}v_t \\ \varepsilon = \sqrt{1 - \alpha_t}x_t + \sqrt{\alpha_t}v_t \end{cases}$$

- SDXL (Tháng 7/2023): Nỗ lực cuối cùng và mạnh mẽ nhất để khai thác kiến trúc Convolutional U-Net truyền thống



- SD3 (Tháng 3/2024) là một cuộc "đại phẫu" toàn diện so với các phiên bản trước đó:
 - Cơ sở toán học: sử dụng Rectified Flow
 - Text Encoder: dùng cùng lúc 3 text encoders khác nhau là OpenCLIP-ViT/G, CLIP-ViT/L and T5-xxl
 - Kiến trúc: thay thế U-Net bằng Multimodal Diffusion Transformer

- Vấn đề đối với mô hình Diffusion thời điểm đó:

$$x_t = \cos(\phi_t)x_0 + \sin(\phi_t)\varepsilon, \varepsilon \sim N(0, 1)$$

- Quỹ đạo di chuyển của Diffusion nằm trên một hypersphere, dẫn đến đường đi cong và nhiễu → Quá trình sampling chậm và tốn kém tính toán
- Rectified Flow (Tháng 9/2022) đơn giản hóa đường đi bằng cách mặc định nó nằm trên một hyperplane

$$x_t = (1 - t)x_0 + t\varepsilon, \varepsilon \sim N(0, 1), t \in [0, 1]$$

Rectified Flow

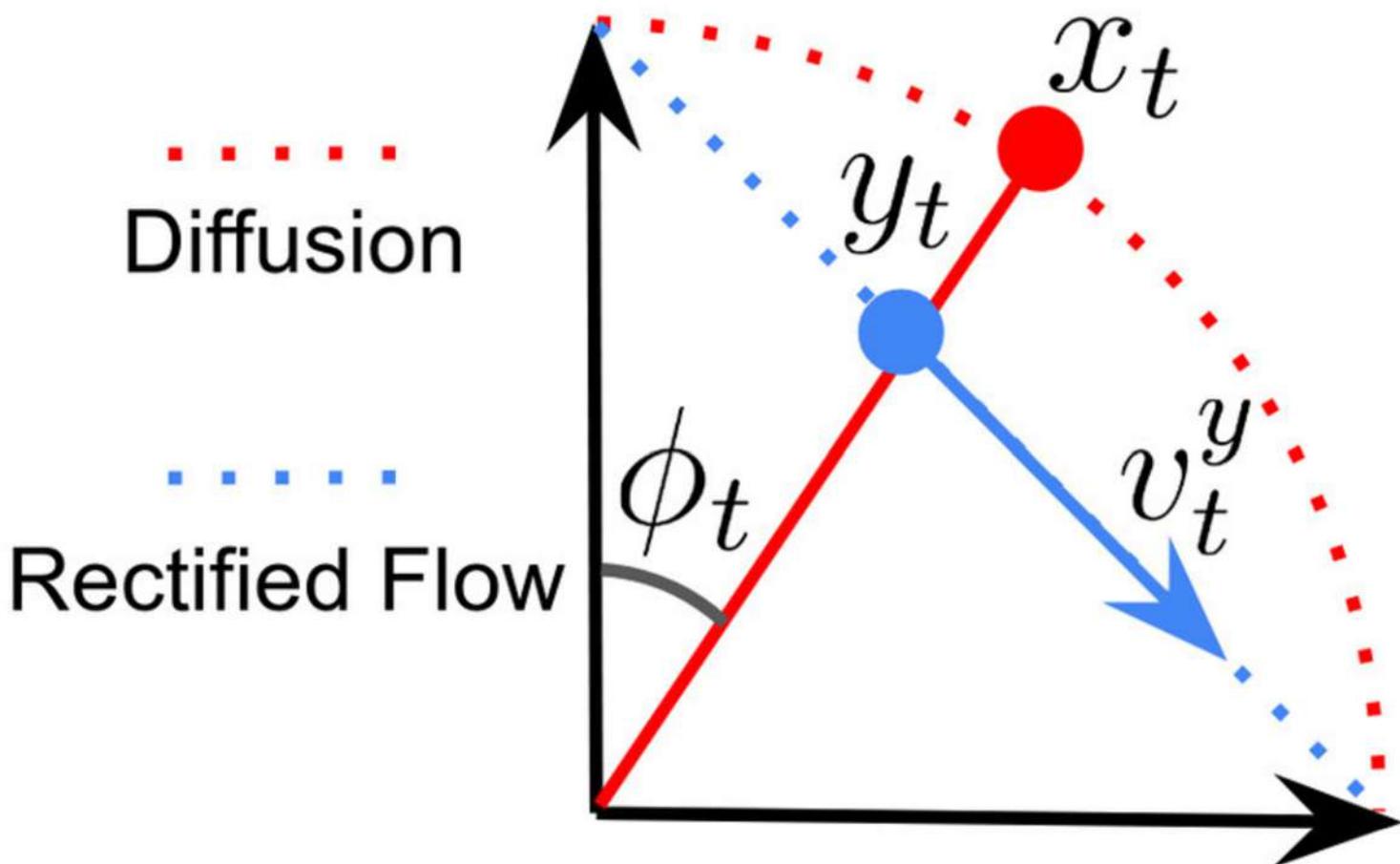
- Mô hình dự đoán vận tốc

$$v_t = \frac{dx_t}{dt} = \varepsilon - x_0$$

- Quá trình sampling

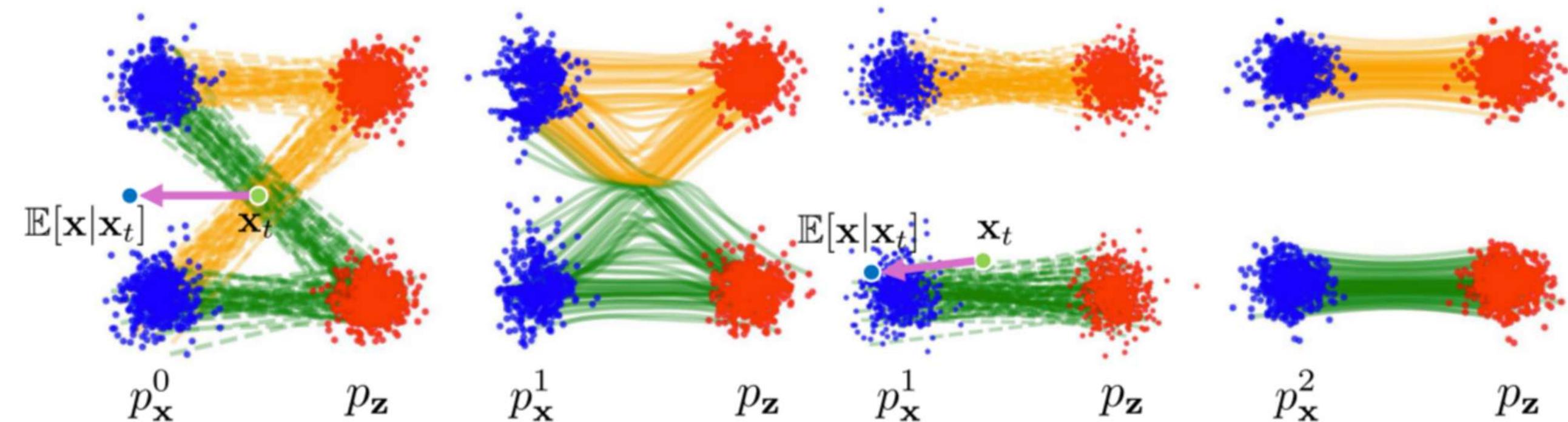
$$x_{s_1} = x_{s_2} + (s_2 - s_1)v(x_{s_2}, s_2)$$

- Mặc dù trên lý thuyết, chỉ cần một bước là đi được từ nhiều về ảnh thật, tuy nhiên giá trị mô hình không hoàn toàn là hằng số dẫn tới vẫn phải chạy nhiều bước nhưng vẫn có thể giảm đáng kể thời gian (1000 bước \rightarrow 4 bước)



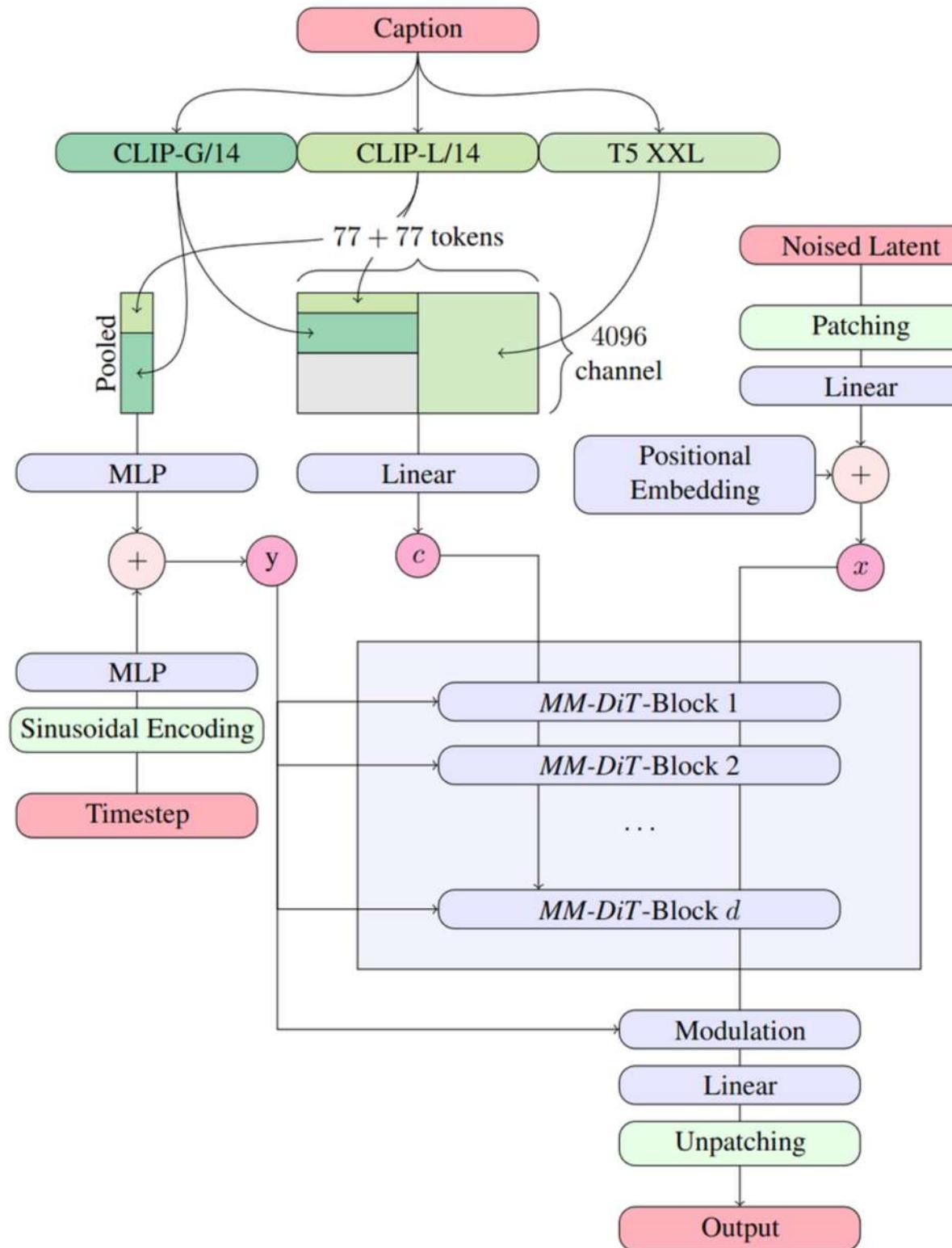
Rectified Flow

- Rectified Flow nảy sinh một vấn đề là “Crossing Paths”, để giải quyết, quá trình chia làm hai giai đoạn:
 - Giai đoạn 1: 1-Rectified Flow

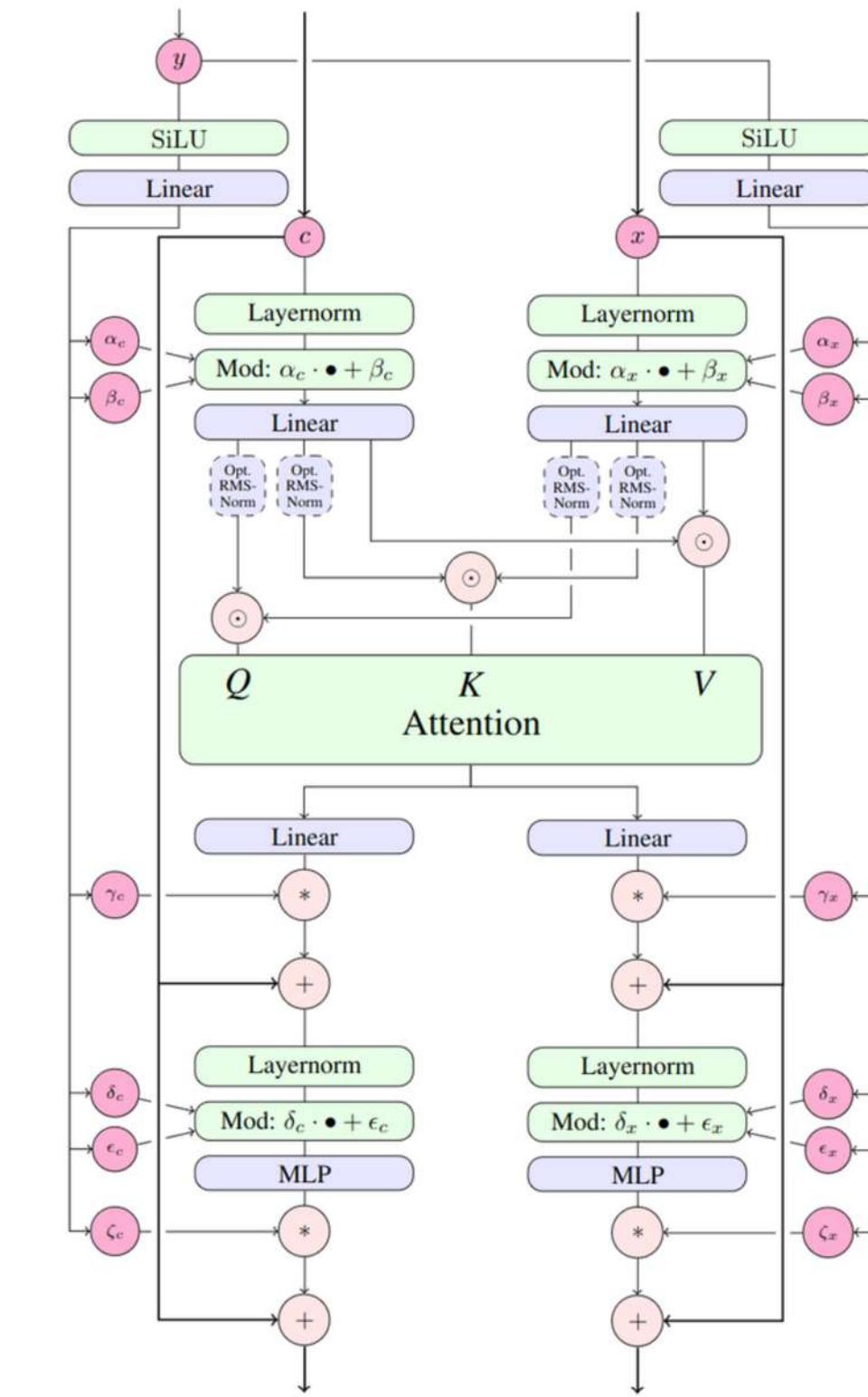


- Giai đoạn 2: Reflow

Multimodal Diffusion Transformer



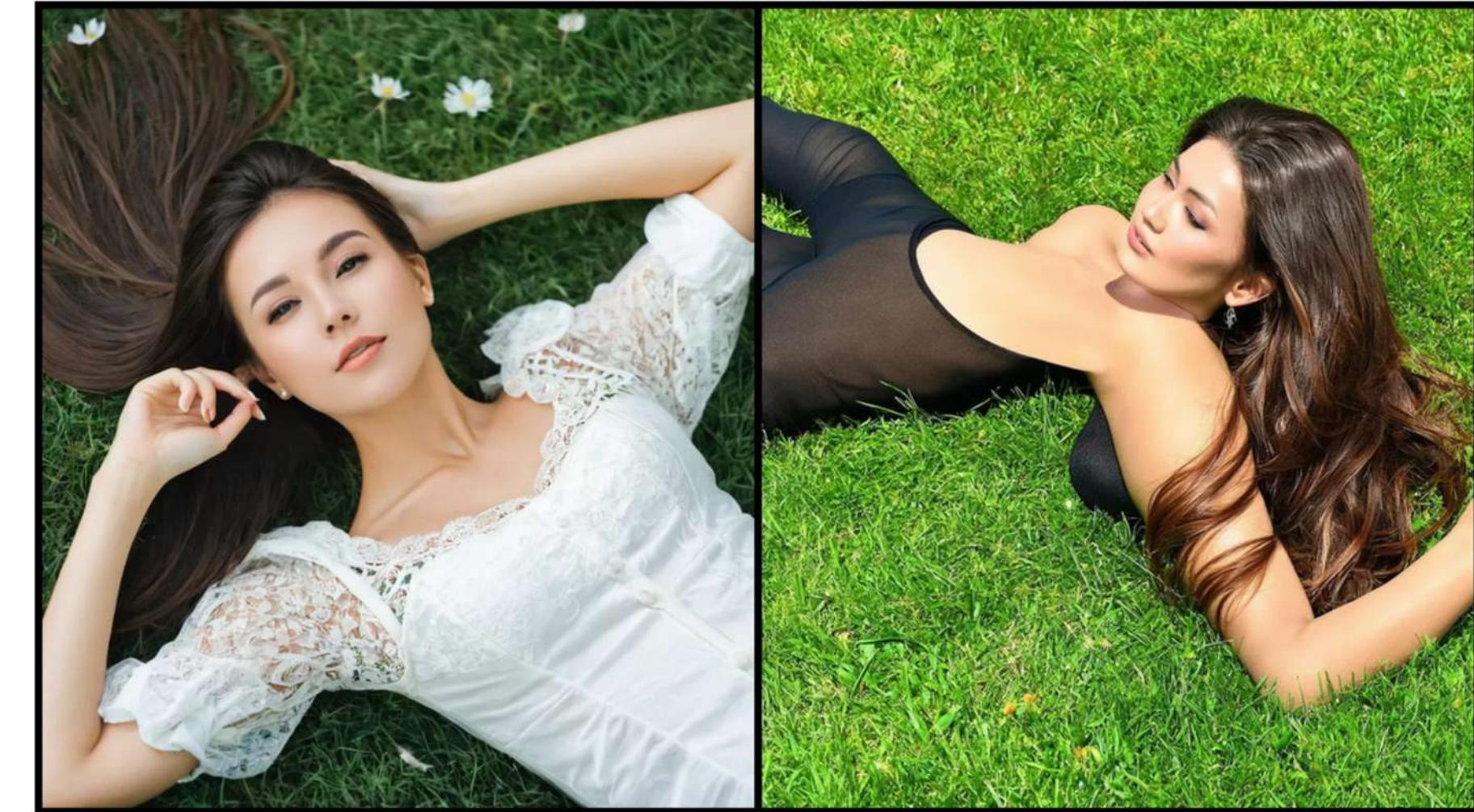
(a) Overview of all components.



(b) One *MM-DiT* block

- Một điểm yếu lớn của SDXL là hiện tượng rò rỉ thuộc tính hay Attribute Bleeding do text được xử lý cục bộ và tách rời so với ảnh
 - Ví dụ với prompt “Áo đỏ quần xanh” tách thành các từ “áo”, “đỏ”, “quần”, “xanh”, mô hình có thể gán nhầm “áo xanh” và “quần đỏ”
- SD3 sử dụng MMDiT với cơ chế Joint Attention giúp cả text lẫn ảnh nhìn nhau và nhìn chính bản thân → Thông tin văn bản không bị "đóng băng" như UNet mà được biến đổi và tinh chỉnh liên tục cùng với ảnh qua từng lớp

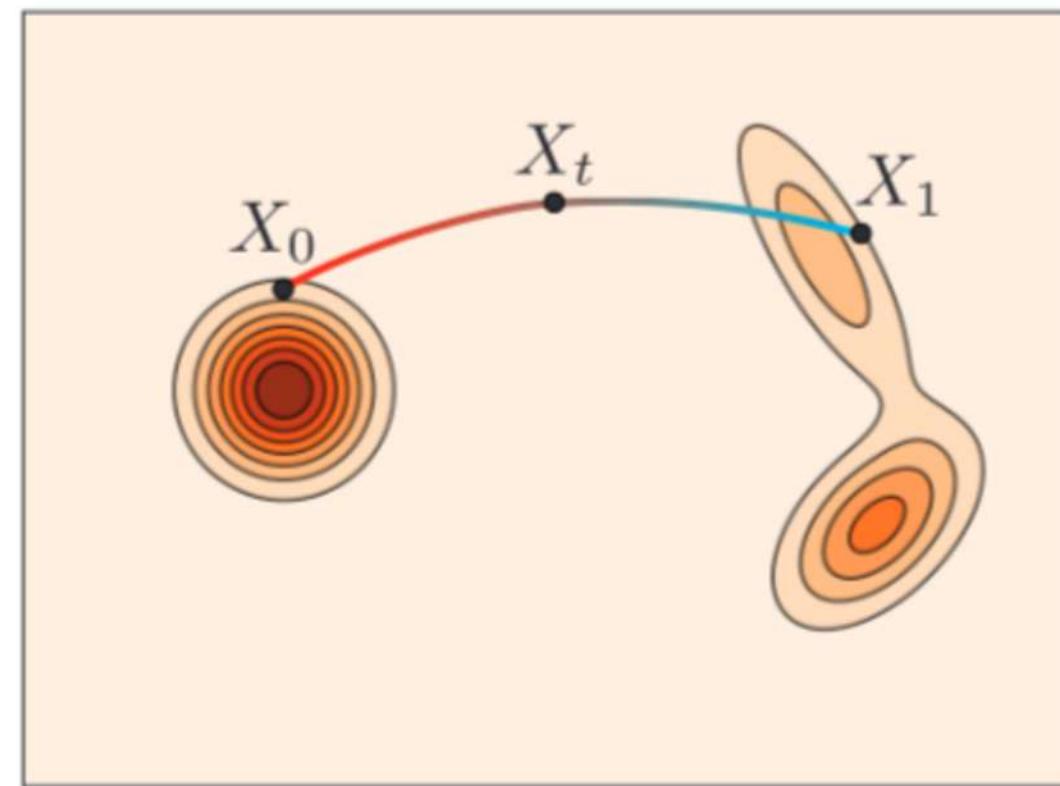
- Do quá trình lọc dữ liệu an toàn cực đoan, lọc bỏ hầu hết các ảnh người thật có tư thế nhạy cảm, vô tình làm SD3 "quên" luôn cấu trúc cơ thể người ở các tư thế đó
- SD3.5 (Tháng 10/2024) nới lỏng và cân bằng lại chiến lược dữ liệu đồng thời cải tiến kiến trúc với QK Normalization



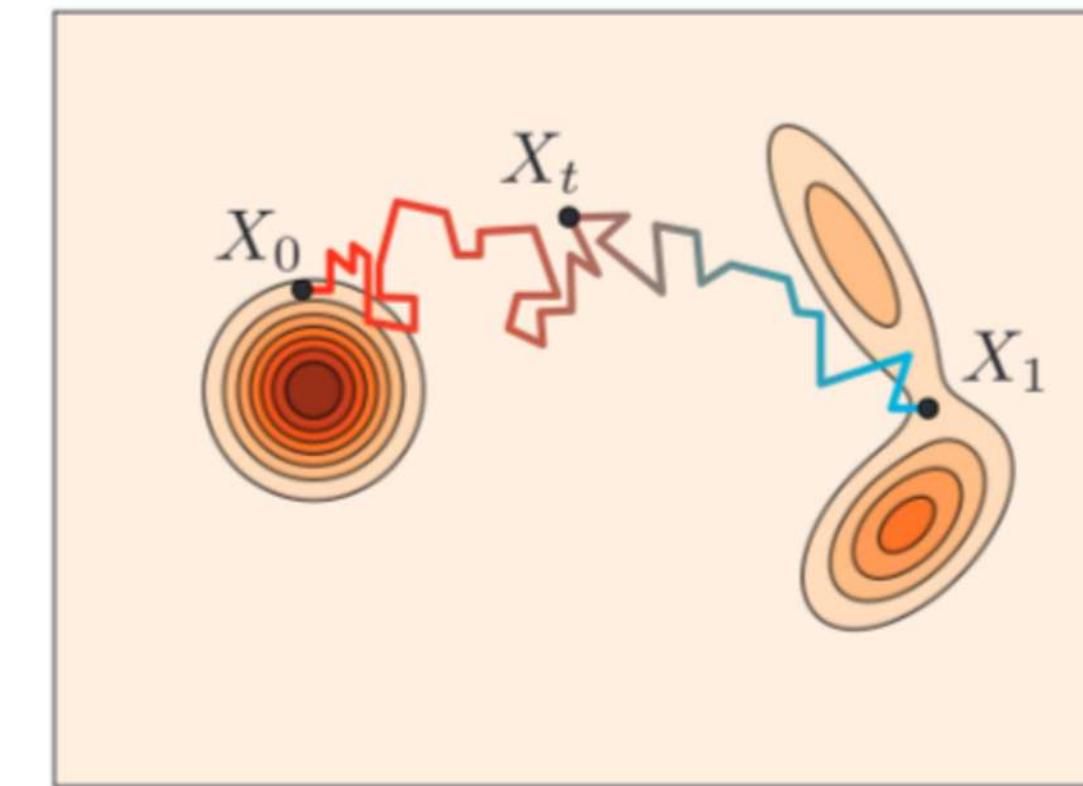
- FLUX được tạo bởi các kỹ sư cũ của Stability AI - nơi tạo ra Stable Diffusion với mục đích hoàn thiện công nghệ Rectified Flow mà SD3 còn dang dở
 - Thuật toán: Flow Matching
 - Kiến trúc: HybridDiT

Flow Matching

- Flow Matching (Tháng 10/2022) là một khung lý thuyết tổng quát, bao trùm lên nhiều phương pháp khác nhau
- Với một con đường bất kỳ, Flow Matching đều giúp đưa ra công thức tính vận tốc để huấn luyện mô hình

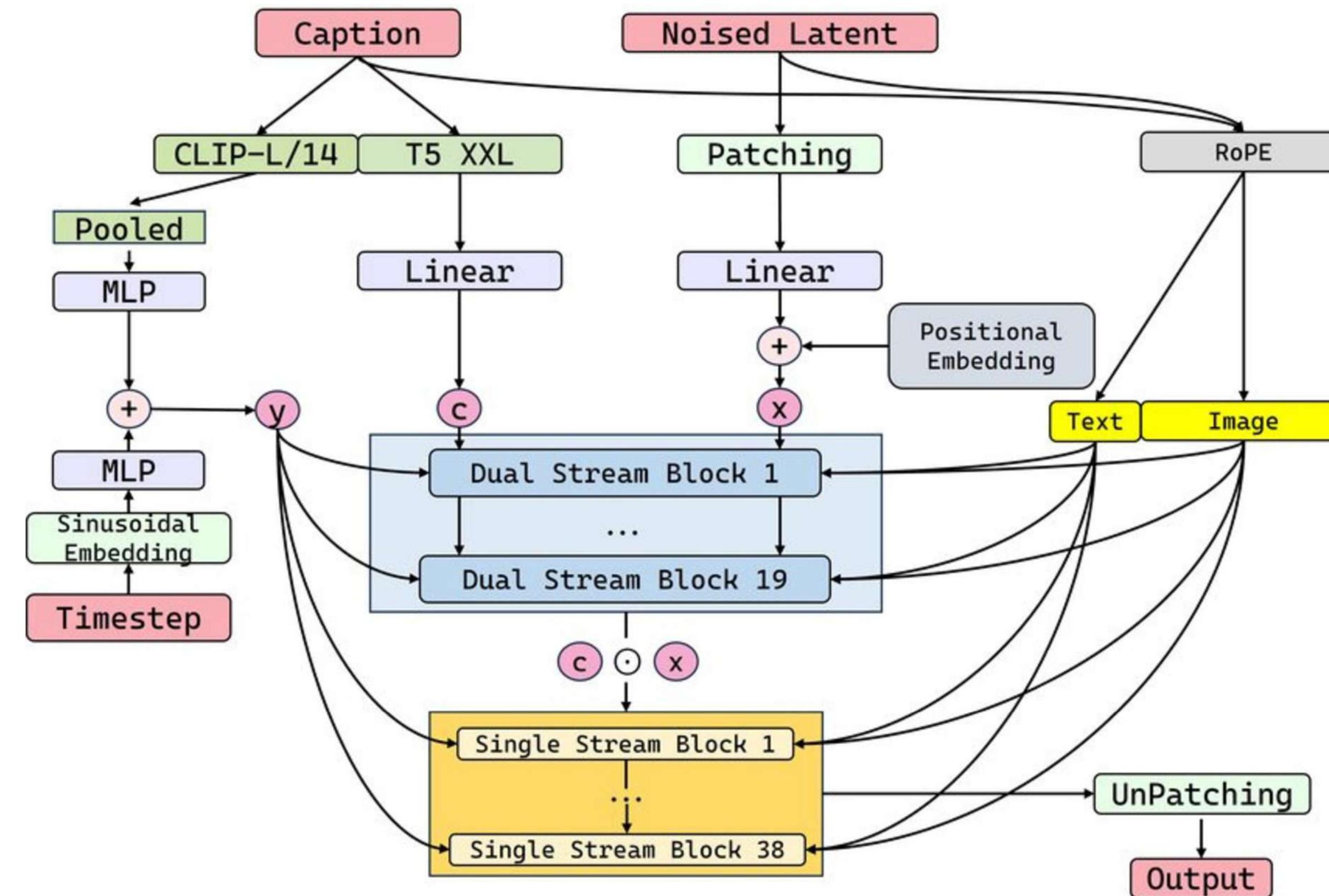


(a) Flow



(b) Diffusion

HybridDiT

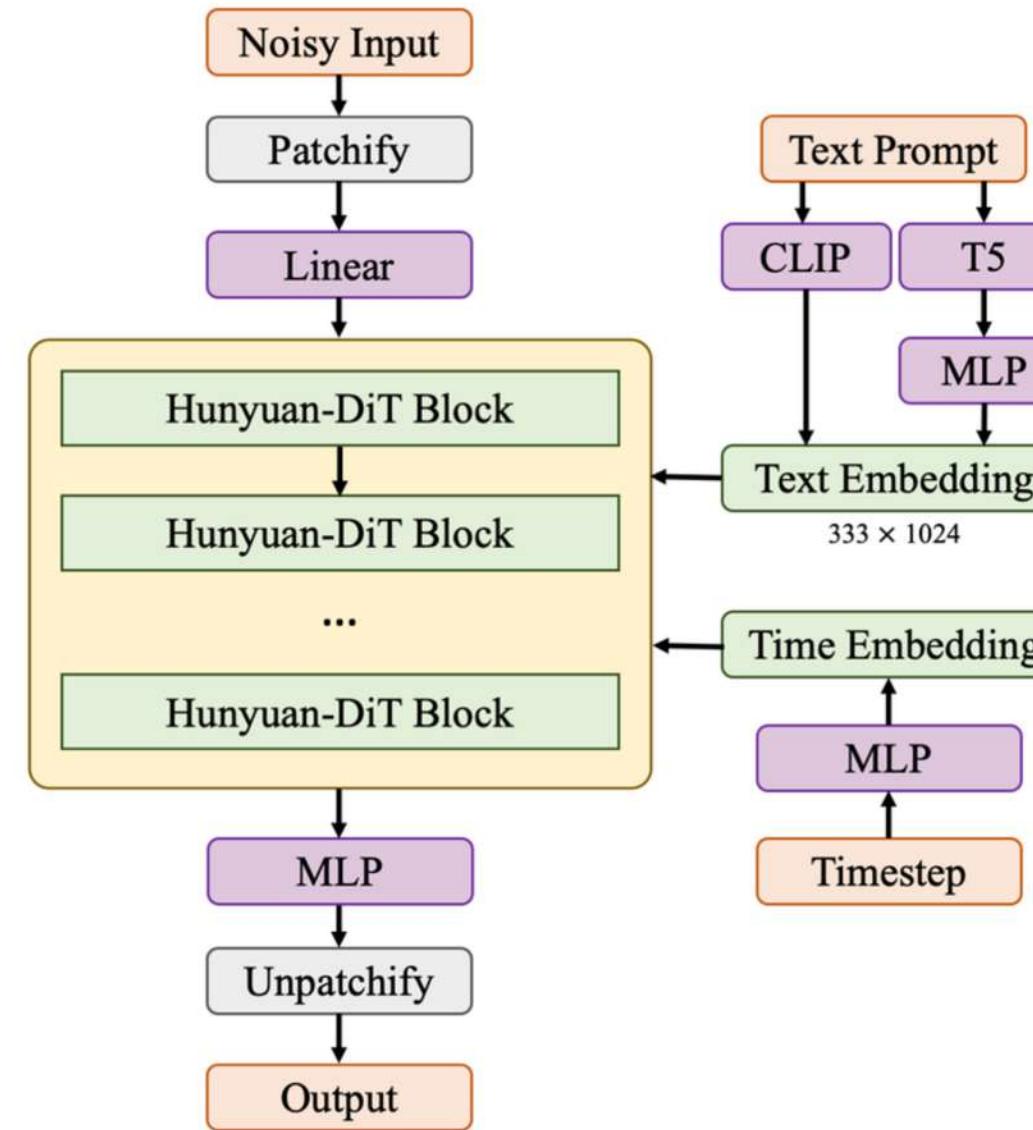


FLUX.1, FLUX.2

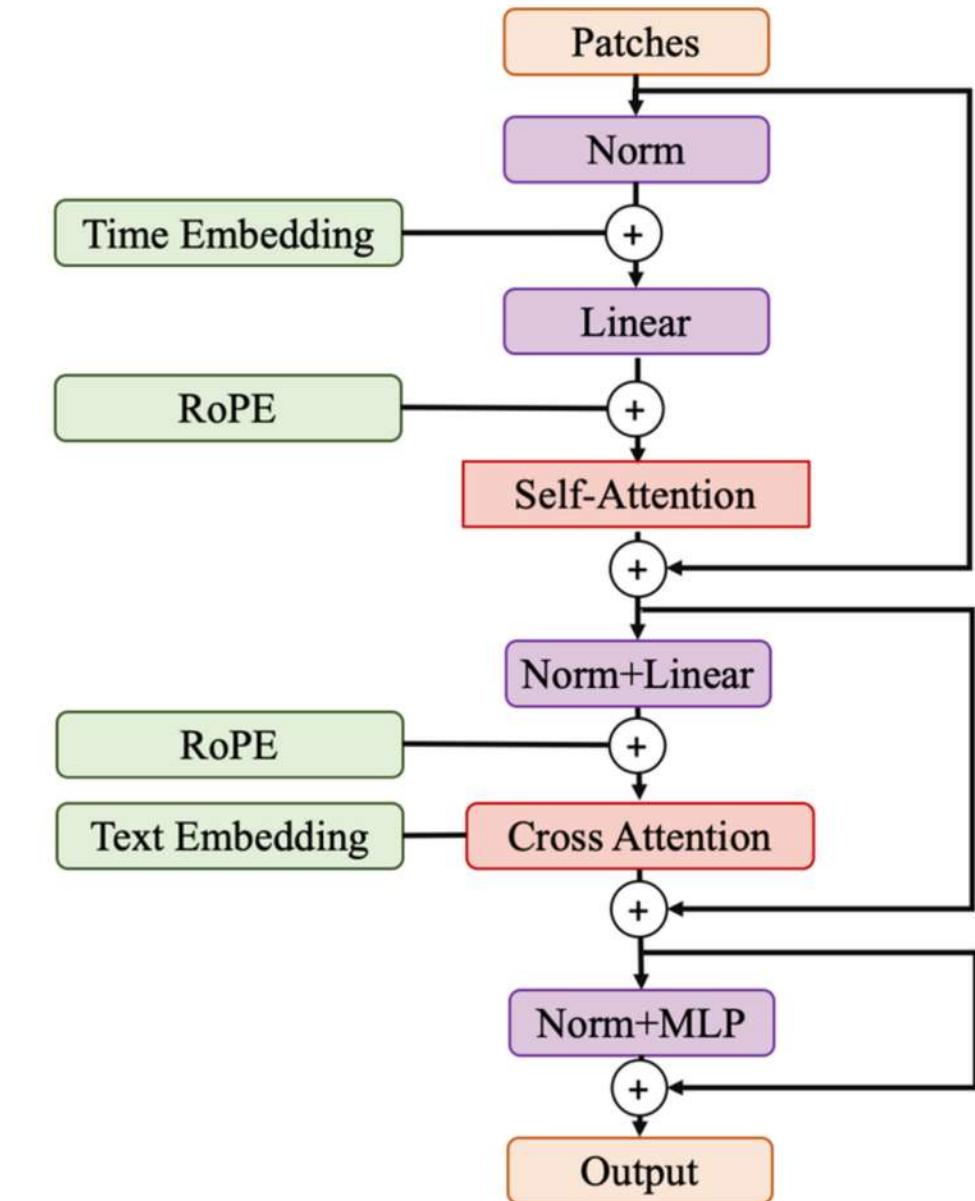
- FLUX.1 (Tháng 8/2024):
 - Kiến trúc: HybridDiT với 12 tỷ tham số
 - Text Encoders: CLIP-ViT/L and T5-xxl
- FLUX.2 (25/11/2025):
 - Kiến trúc: tăng lên 32 tỷ tham số



- Hunyuan-DiT
(Tháng 5/2024):
 - Kiến trúc: DiT
 - Bilingual Text Encoders: CLIP (ViT-L/14) hiểu tiếng Anh và mT5 để hiểu tiếng Trung

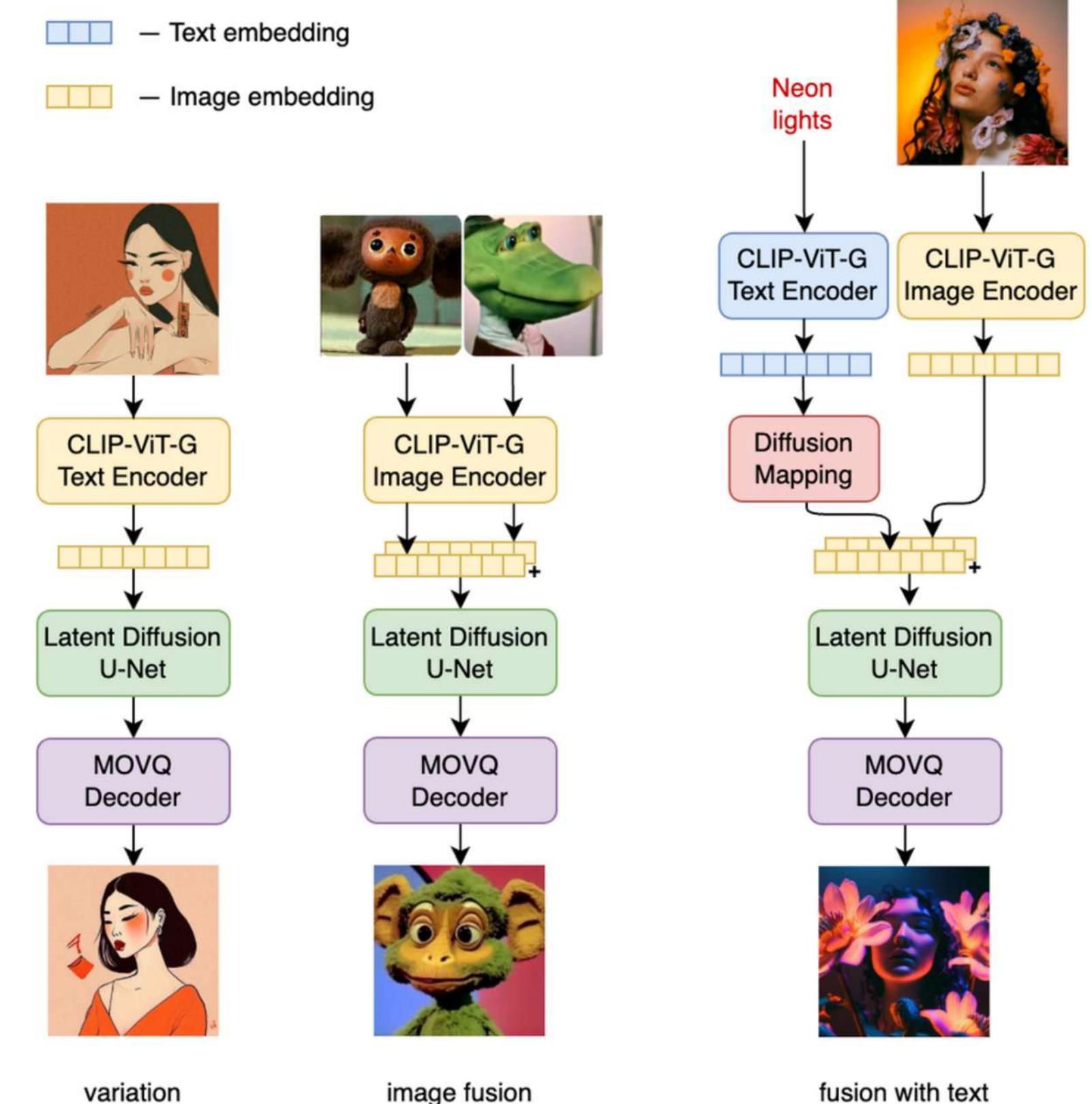


(a) Hunyuan-DiT



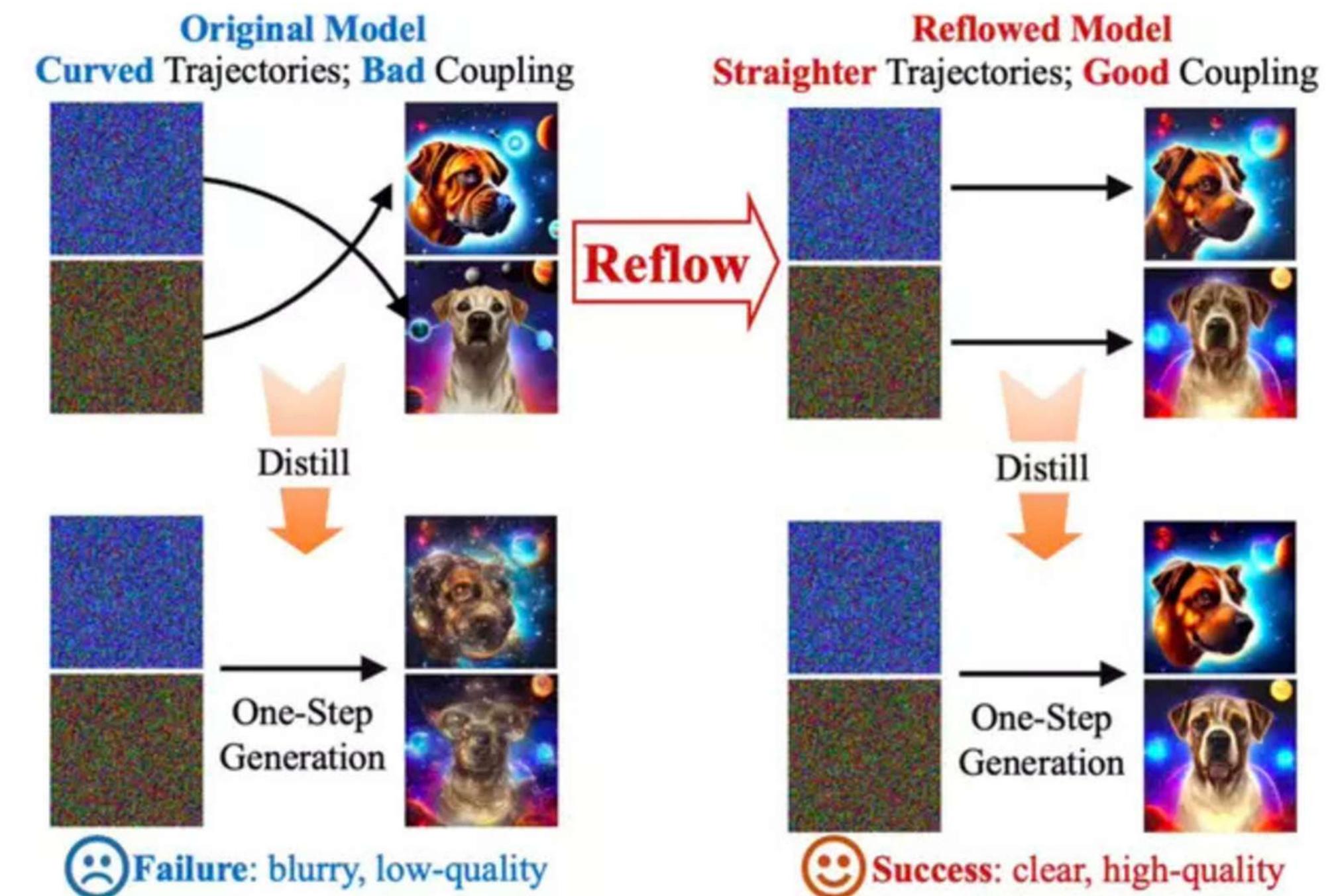
(b) Hunyuan-DiT Block

- Kandinsky 2.1 (Tháng 4/2023)
 - Kiến trúc: Two-Stage Latent Diffusion
 - Text Encoder: XLM-RoBERTa
 - Autoencoder: MoVQ-GAN
- Kandinsky 3.0 (Tháng 11/2023)
 - Kiến trúc: U-Net với 11.9 tỷ tham số
 - Text Encoder: Flan-UL2 của Google



Rectified Diffusion

- InstaFlow (Tháng 9/2023) áp dụng lý thuyết Rectified Flow để "ép" mô hình SD1.5 thành quỹ đạo thẳng



Ứng dụng bài toán thực tế

- MS-COCO 2017 Validation Set bao gồm 5000 ảnh, mỗi ảnh chứa 5 câu mô tả khác nhau (tổng 25014 mô tả)



- Các chỉ số dùng để đánh giá SD1.5, SDXL, SD3, SD3.5, FLUX và Hunyuan:
 - FID Score (Càng thấp càng tốt): Đo khoảng cách giữa tập ảnh sinh ra và tập ảnh thật
 - CLIP Score (Càng cao càng tốt): Đo mức độ khớp nhau giữa văn bản mô tả và ảnh
 - Aesthetic Score (Càng cao càng tốt): Đánh giá mức độ “đẹp” của ảnh sinh ra
 - Inference Time: Trung bình thời gian sinh một ảnh

Ứng dụng bài toán thực tế

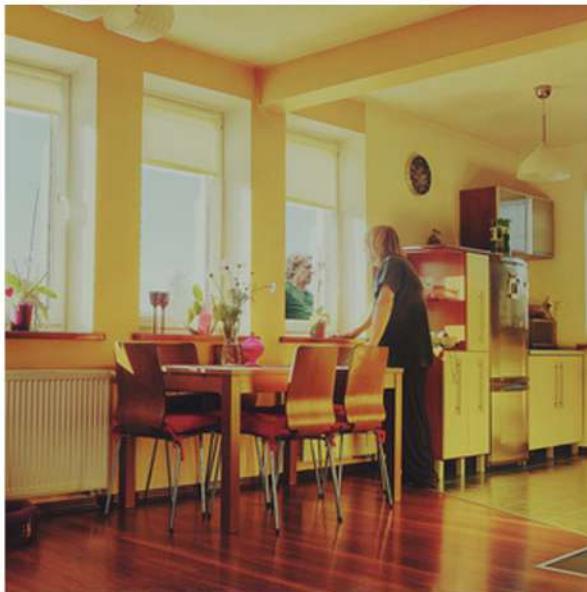
Model	FID	CLIP	Aesthetic	Inference Time (s)
SD1.5	156.48	27.76	5.35	1.05
SDXL	154.20	28.38	5.54	6.57
SD3	148.47	28.14	5.36	12.04
SD3.5	146.23	28.24	5.34	14.8
FLUX.1	154.19	28.18	5.43	36.65
Hunyuan-DiT	159.25	27.85	5.84	18.31
Kandinsky	154.49	28.54	5.65	5.10



Ứng dụng bài toán thực tế

“A woman stands in the dining area at the table”

Real



SD1.5



SDXL



SD3



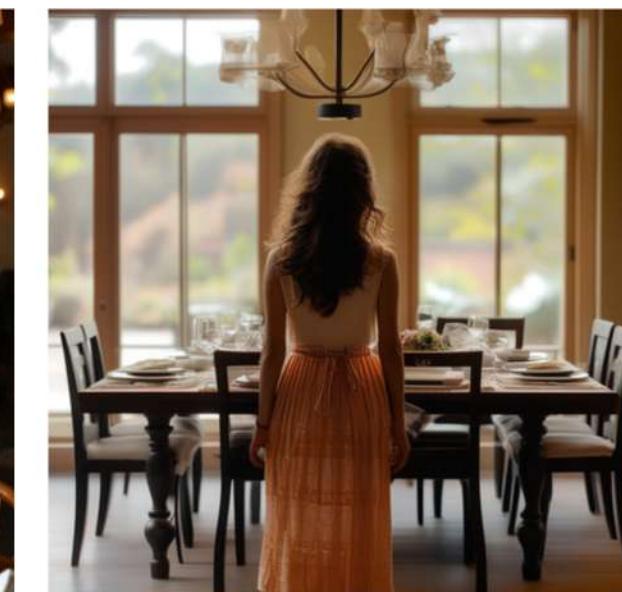
SD3.5



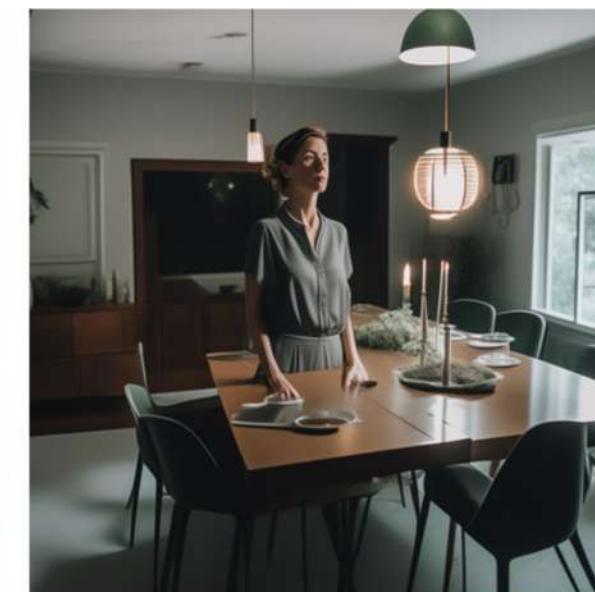
FLUX



Hunyuan



Kadinsky



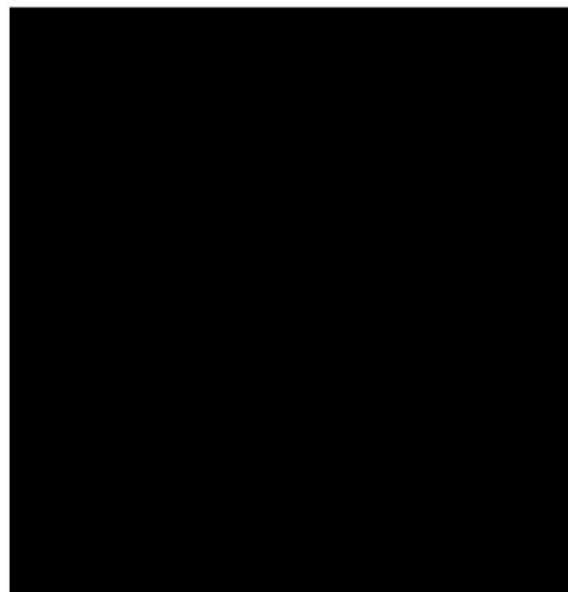
Ứng dụng bài toán thực tế

“A stop sign installed upside down on a street corner”

Real



SD1.5



SDXL



SD3



SD3.5



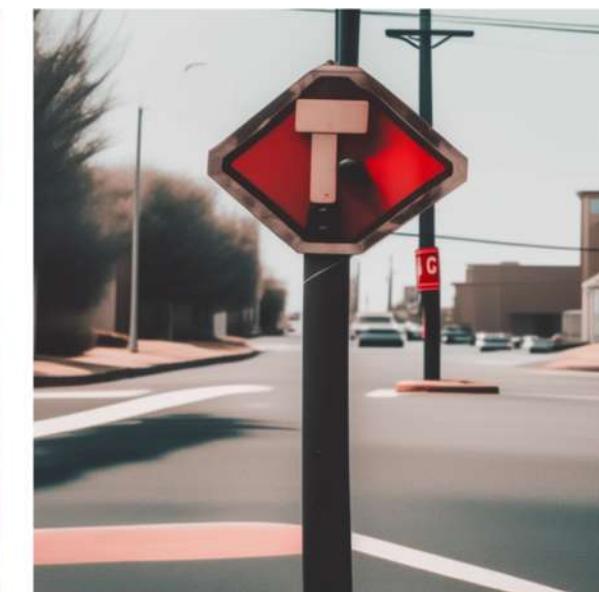
FLUX



Hunyuan



Kadinsky



Ứng dụng bài toán thực tế

“A black and white photo of a group of kids”

Real



SD1.5



SDXL



SD3



SD3.5



FLUX



Hunyuan



Kadinsky



A large, faint watermark of the HUST logo is visible across the entire background of the slide.

HUST

THANK YOU !