# How to AI (Almost) Anything

## Lecture 10 – Modern Generative AI

**Chanakya Ekbote**
Multisensory Intelligence Group
MIT Media Lab
Adapted From: https://diffusion.csail.mit.edu
Thanks to XKCD for the comics

mit media lab

MIT EECS

# Administrative Details

1. Reading Assignments due Tomorrow
    1. LLaMA-Adapter: Efficient Fine-tuning of Language Models with Zero-init Attention
    2. Cobra: Extending Mamba to Multi-Modal Large Language Model for Efficient Inference

2. Presentations are due this Thursday.

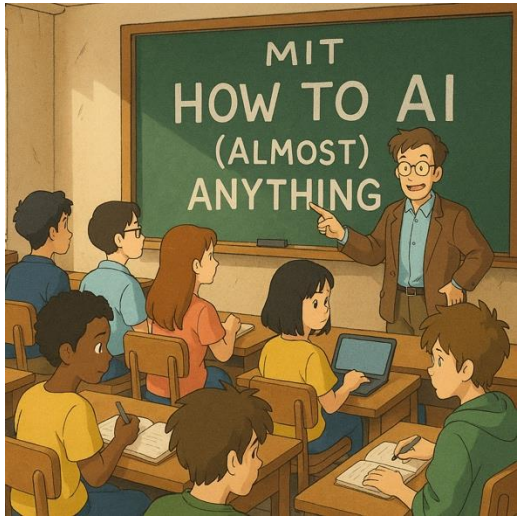
3. Hope you are all on track for your projects (talk to us in case of any blockers).

# Todays Lecture

1. What are Generative Models?
2. Current State of the Art (Flow Matching)
3. Conditional Generation
4. Architectures
5. Tips to Train these Models

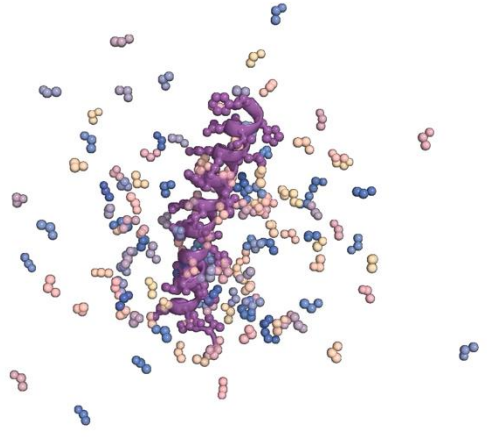
# Generative Models

> "Creating noise from data is easy; creating data from noise is generative modeling.
>
> Song et *al.* [30]




OpenAI GPT 4o

OpenAI Sora

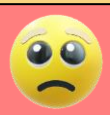


RF Diffusion

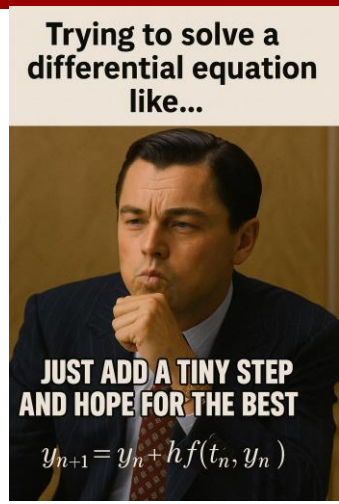
# Prerequisites – Solving Differential Equations

$$\frac{dx}{dt} = t \quad \rightarrow \quad \int dx = \int t dt \quad \rightarrow \quad x = \frac{t^2}{2} + c$$

$$\frac{dx}{dt} = u_t(x) \quad \rightarrow \quad \text{Mostly Unsolvable}$$



# Prerequisites – Euler Integration



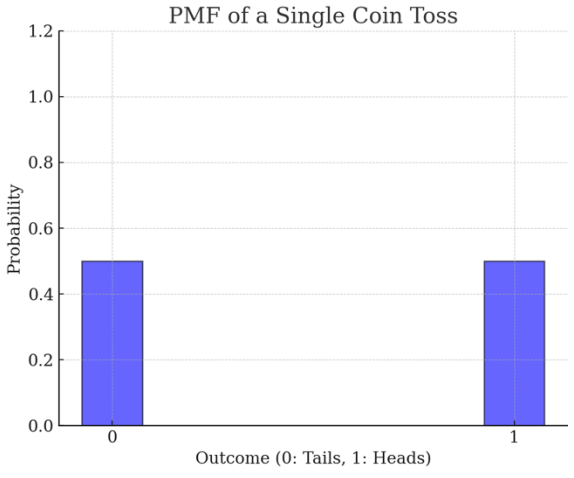

$$\frac{dx}{dt} = u_t(x) \rightarrow \frac{dx}{dt} \approx \frac{x_{t+h} - x_t}{h} = u_t(x_t) \rightarrow x_{t+h} = x_t + h\,u_t(x_t)$$

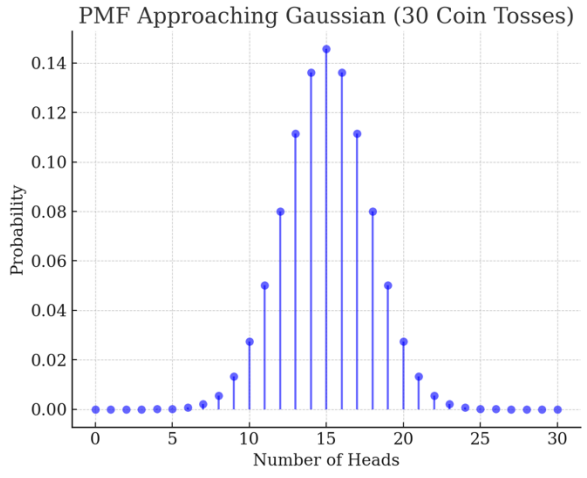
# Prerequisites - Probability

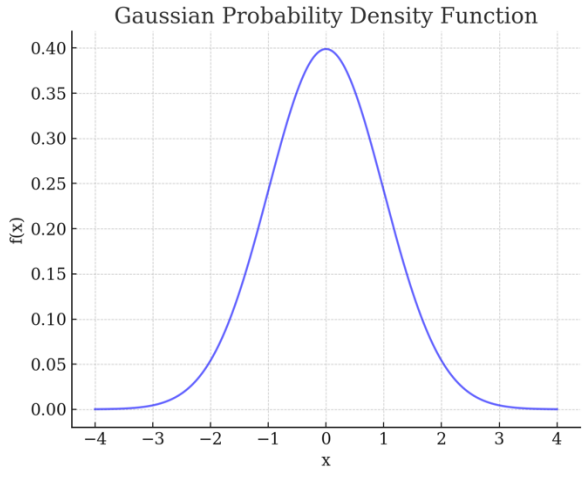
MODIFIED BAYES' THEOREM:

$$P(H|X) = P(H) \times \left(1 + P(C) \times \left(\frac{P(X|H)}{P(X)} - 1\right)\right)$$

H: HYPOTHESIS
X: OBSERVATION
P(H): PRIOR PROBABILITY THAT H IS TRUE
P(X): PRIOR PROBABILITY OF OBSERVING X
P(C): PROBABILITY THAT YOU'RE USING BAYESIAN STATISTICS CORRECTLY





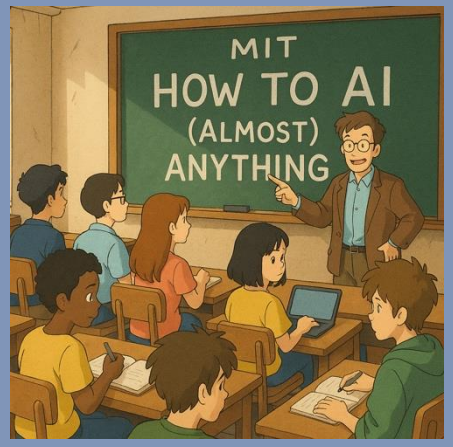
$$x \sim N(\mu, \sigma^2)$$
$$\epsilon \sim N(0, 1)$$
$$x \sim \mu + \sigma\epsilon$$

# What is Generation?

Generation implies sampling from the data distribution

$$x \sim P_{data}$$

$$x =$$



# Why Sampling is a hard problem?

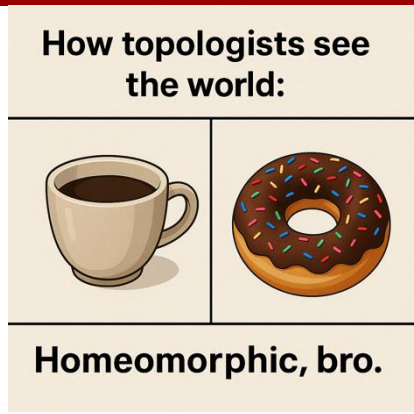
"Sampling the world exactly is impossible; approxıimations are everything."

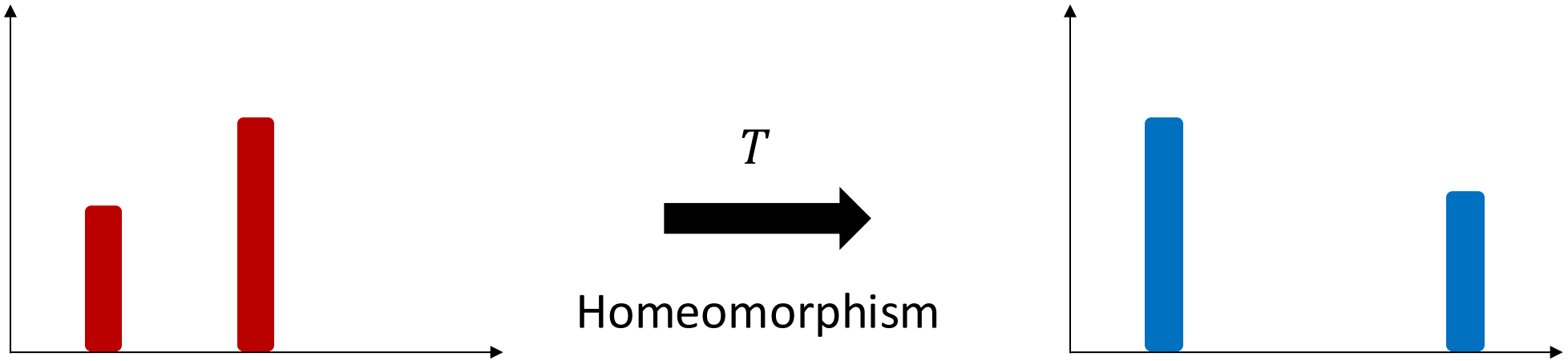
Léon Bottou

1. We don't know $P_{data}$?

2. Even if we know, $P_{data}$ we can't always sample from it.

# Warmup

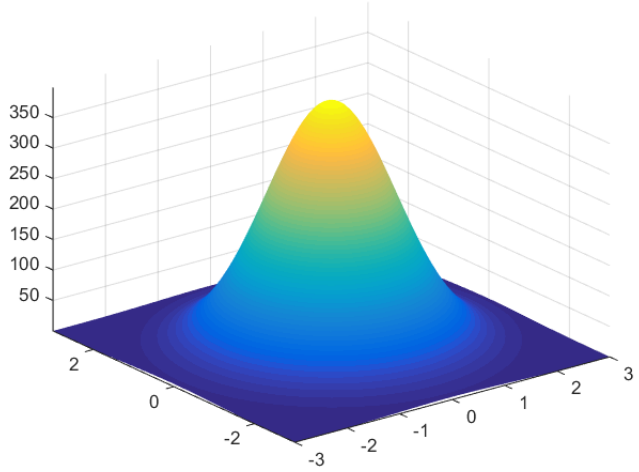

How topologists see the world:
Homeomorphic, bro.


$T$
Homeomorphism

# Warmup

# What if we know $P_{data}$ ?

$P_{Gaussian}$

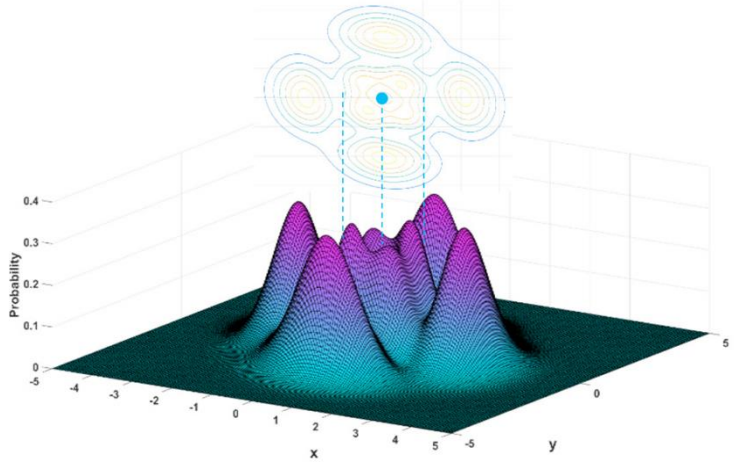



From Brilliant

$T$

Homeomorphism

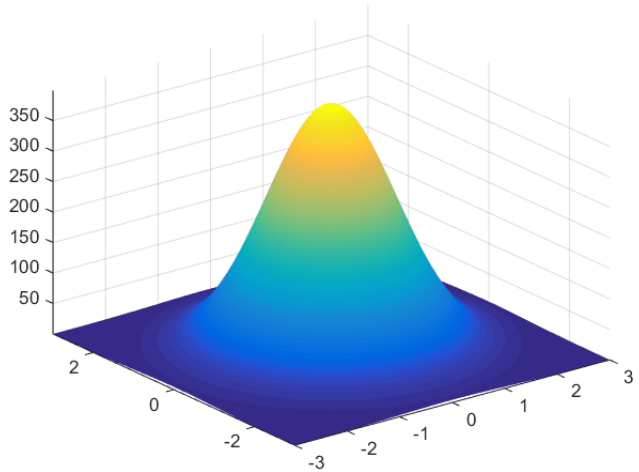
$P_{data}$




From https://arxiv.org/pdf/2103.05518

# What if we know an approximation of $P_{data}$ ?

$P_{Gaussian}$



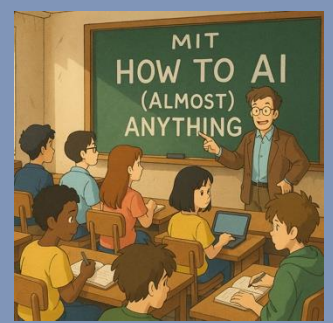

From Brilliant

$T$

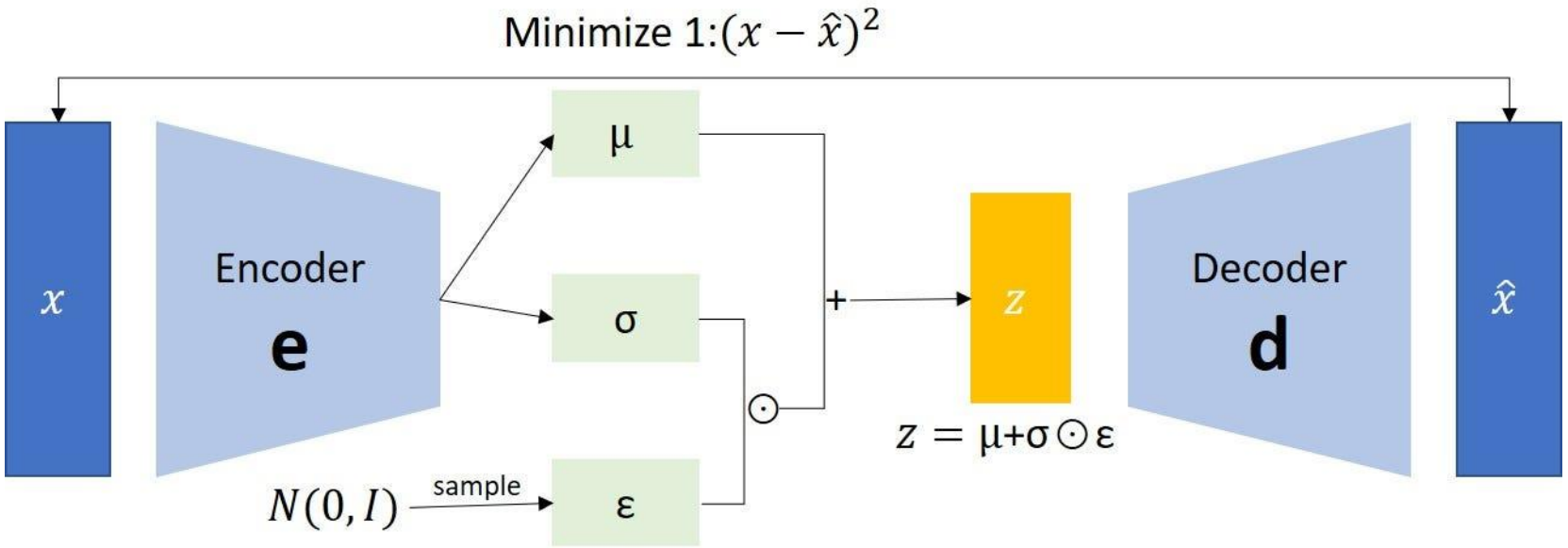
Homeomorphism

$x \sim P_{data}$

$x =$



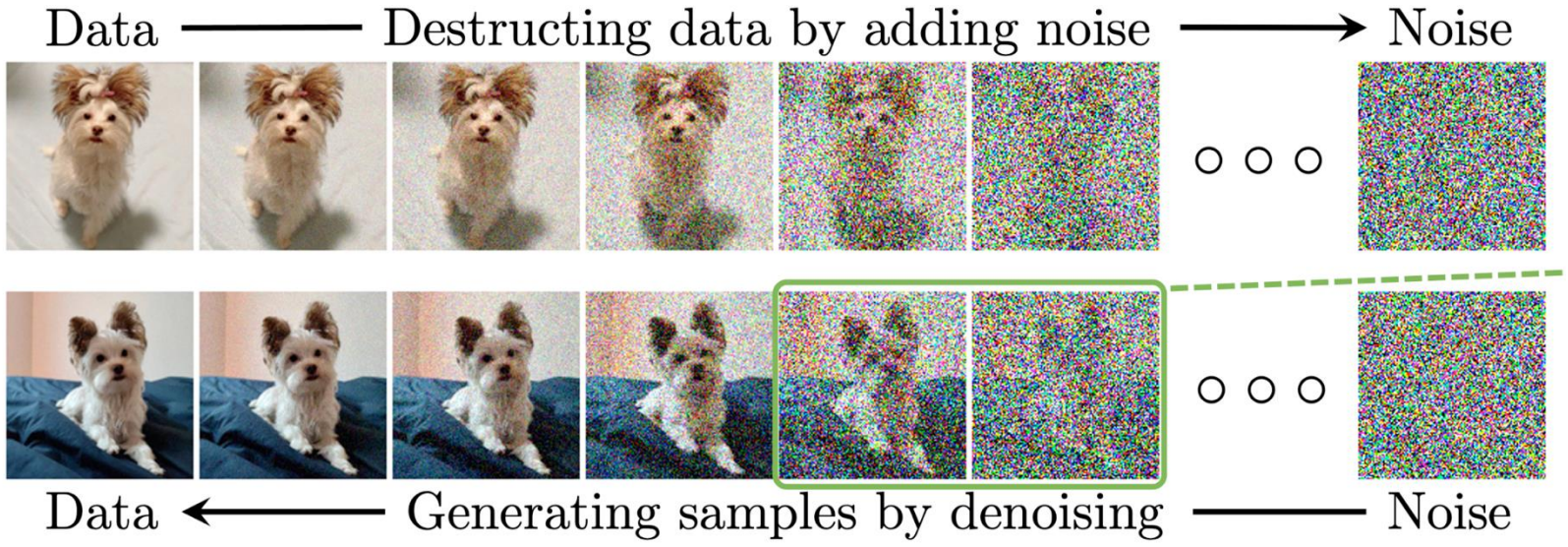
# Variational Autoencoders




Minimize 2: $\frac{1}{2}\sum_{i=1}^{N}(\exp(\sigma_i) - (1+\sigma_i) + \mu_i{}^2)$

Variational Autoencoders

# Denoising Diffusion Models




Diffusion Models: A Comprehensive Survey of Methods and Applications

# What methods do we know?

| Feature | VAE | Diffusion Models | Flow Matching |
|---|---|---|---|
| **Core Idea** | Encode/decode with latent noise | Add noise and learn to reverse it | Learn a continuous flow from noise |
| **Training Objective** | Minimize reconstruction + KL loss | Learn the score (gradient) of data | Match a vector field (ODE-based) |
| **Noise Handling** | Noise in latent space | Progressive noise over time | Start from noise, smooth transform |
| **Sampling Speed** | Very fast (one pass) | Slow (many denoising steps) | Faster (solving an ODE) |
| **Advantages** | Simple, fast, interpretable | Very high-quality outputs | High quality + faster than diffusion |
| **Disadvantages** | Blurry samples, limited expressiveness | Expensive, slow sampling | Newer, still developing |
| **Key Examples** | VAE (2013), β-VAE | DDPM, Stable Diffusion | Flow Matching (2023), Rectified Flow |

# Flow Models

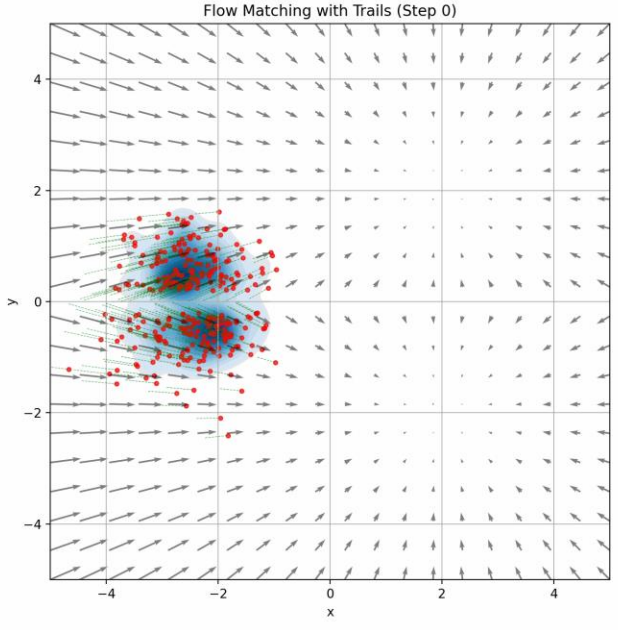
$$X_0 \sim p_{\text{init}}$$

$$\frac{\mathrm{d}}{\mathrm{d}t} X_t = u_t^\theta(X_t)$$

**Algorithm 1** Sampling from a Flow Model with Euler method

**Require:** Neural network vector field $u_t^\theta$, number of steps $n$

1: Set $t = 0$
2: Set step size $h = \frac{1}{n}$
3: Draw a sample $X_0 \sim p_{\text{init}}$
4: **for** $i = 1, \ldots, n-1$ **do**
5: $\quad X_{t+h} = X_t + h u_t^\theta(X_t)$
6: $\quad$ Update $t \leftarrow t + h$
7: **end for**
8: **return** $X_1$



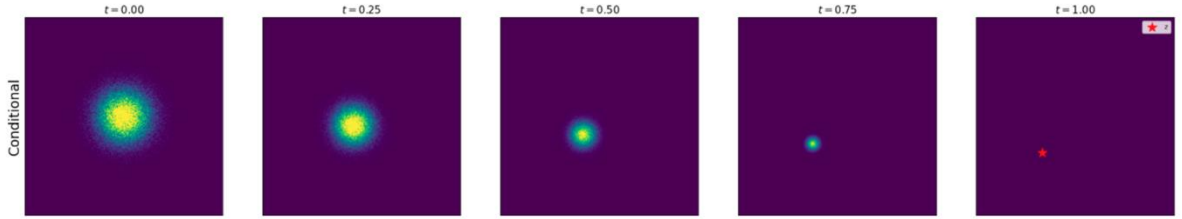
# Conditional and Marginal Probability Paths (Example)

$$\alpha_t = t, \beta_t = 1 - t$$

$$p_t(\cdot\,|z) = \mathcal{N}(\alpha_t z, \beta_t^2 I_d)$$

$$p_0(.\,|z) = \mathcal{N}(0, I_d)$$

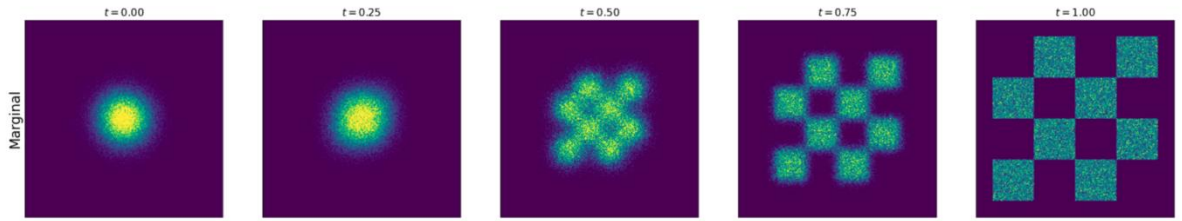
$$p_1(.\,|z) = \delta_z$$




$$z \sim p_{\text{data}}, \quad x \sim p_t(\cdot|z) \quad \Rightarrow x \sim p_t$$

$$p_t(x) = \int p_t(x|z) p_{\text{data}}(z) \mathrm{d}z$$

$$p_0 = p_{\text{init}} \quad \text{and} \quad p_1 = p_{\text{data}}.$$




$$z \sim p_{\text{data}}, \epsilon \sim p_{\text{init}} = \mathcal{N}(0, I_d) \Rightarrow x = \alpha_t z + \beta_t \epsilon \sim p_t$$

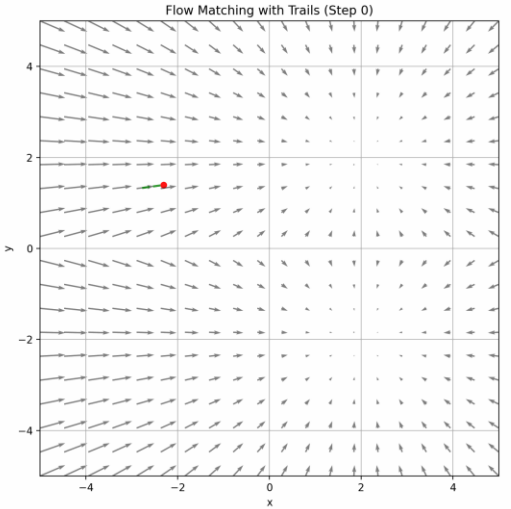
# Conditional Vector Field

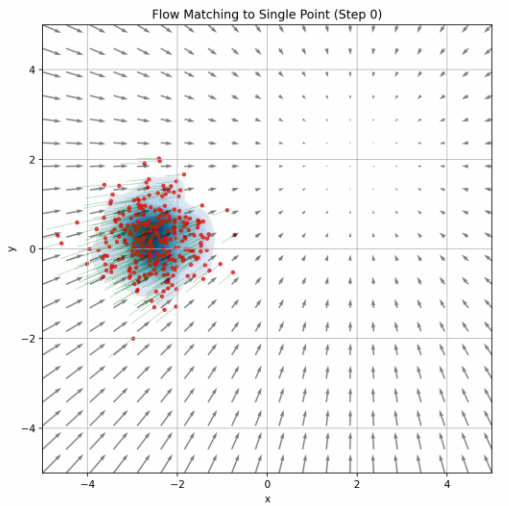
$$X_0 \sim P_{init} \qquad \frac{dX_t}{dt} = u_t^{target}(X_t|z) \implies X_t \sim P_t(\cdot\,|z)$$

$$P_t(x_t|z) = \mathcal{N}(x_t;\ \alpha_t z, \beta_t^2 I_d) \quad \Rightarrow \quad u_t^{target}(x_t|z) = \left(\dot{\alpha}_t - \frac{\dot{\beta}_t}{\beta_t}\alpha_t\right) z + \frac{\dot{\beta}_t}{\beta_t} x_t$$

# Conditional Vector Field (Intuition)

$$u_t^{target}(x_t|z) = \left(\dot{\alpha}_t - \frac{\dot{\beta}_t}{\beta_t}\alpha_t\right)z + \frac{\dot{\beta}_t}{\beta_t}x_t$$



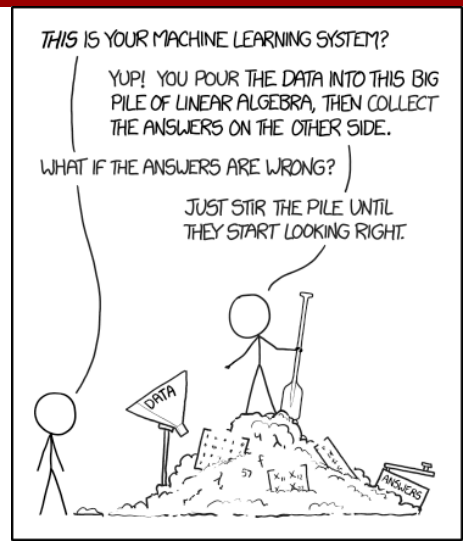
# Conditional Vector Field

$$P_t(x_t|z) = \mathcal{N}(x_t;\ \alpha_t z, \beta_t^2 I_d) \quad \Rightarrow \quad u_t^{target}(x_t|z) = \left(\dot{\alpha}_t - \frac{\dot{\beta}_t}{\beta_t}\alpha_t\right) z + \frac{\dot{\beta}_t}{\beta_t} x_t$$

$$\alpha_t = t, \beta_t = 1 - t$$
$$P_t(x_t|z) = \mathcal{N}(x_t; tz, (1-t)^2 I_d) \quad \Rightarrow \quad u_t^{target}(x_t|z) = \frac{1+t}{1-t} z - \frac{1}{1-t} x_t$$

$$x_t = t\, z + (1-t)\epsilon \quad \Rightarrow \quad u_t^{target}(x|z) = z - \epsilon$$

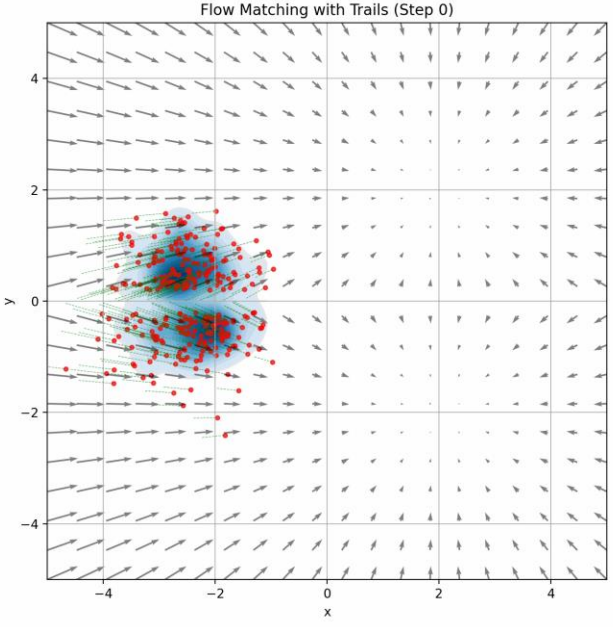
# Loss




$$\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}_{z\sim p_{\mathrm{data}}, x\sim p_t(\cdot|z)}[\|u_t^\theta(x) - u_t^{\mathrm{target}}(x|z)\|^2].$$

# Loss (Intuition)

$$\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}_{z\sim p_{\mathrm{data}}, x\sim p_t(\cdot|z)}[\|u_t^\theta(x) - u_t^{\mathrm{target}}(x|z)\|^2].$$



# Algorithm

**Algorithm 3** Flow Matching Training Procedure (here for Gaussian CondOT path $p_t(x|z) = \mathcal{N}(tz, (1-t)^2)$)

**Require:** A dataset of samples $z \sim p_{\text{data}}$, neural network $u_t^\theta$

1: **for** each mini-batch of data **do**
2: Sample a data example $z$ from the dataset.
3: Sample a random time $t \sim \text{Unif}_{[0,1]}$.
4: Sample noise $\epsilon \sim \mathcal{N}(0, I_d)$
5: Set $x = tz + (1-t)\epsilon$ (General case: $x \sim p_t(\cdot|z)$)
6: Compute loss

$$\mathcal{L}(\theta) = \|u_t^\theta(x) - (z - \epsilon)\|^2 \qquad \text{(General case: } = \|u_t^\theta(x) - u_t^{\text{target}}(x|z)\|^2\text{)}$$

7: Update the model parameters $\theta$ via gradient descent on $\mathcal{L}(\theta)$.
8: **end for**

# Conditional Generation Loss

$$\mathcal{L}_{\text{CFM}}^{\text{guided}}(\theta) = \mathbb{E}_{(z,y)\sim p_{\text{data}}(z,y),\, t\sim\text{Unif}[0,1),\, x\sim p_t(\cdot|z)} \|u_t^\theta(x|y) - u_t^{\text{target}}(x|z)\|^2.$$

$$\mathcal{L}_{\text{CFM}}^{\text{CFG}}(\theta) = \mathbb{E}_{\square} \|u_t^\theta(x|y) - u_t^{\text{target}}(x|z)\|^2$$

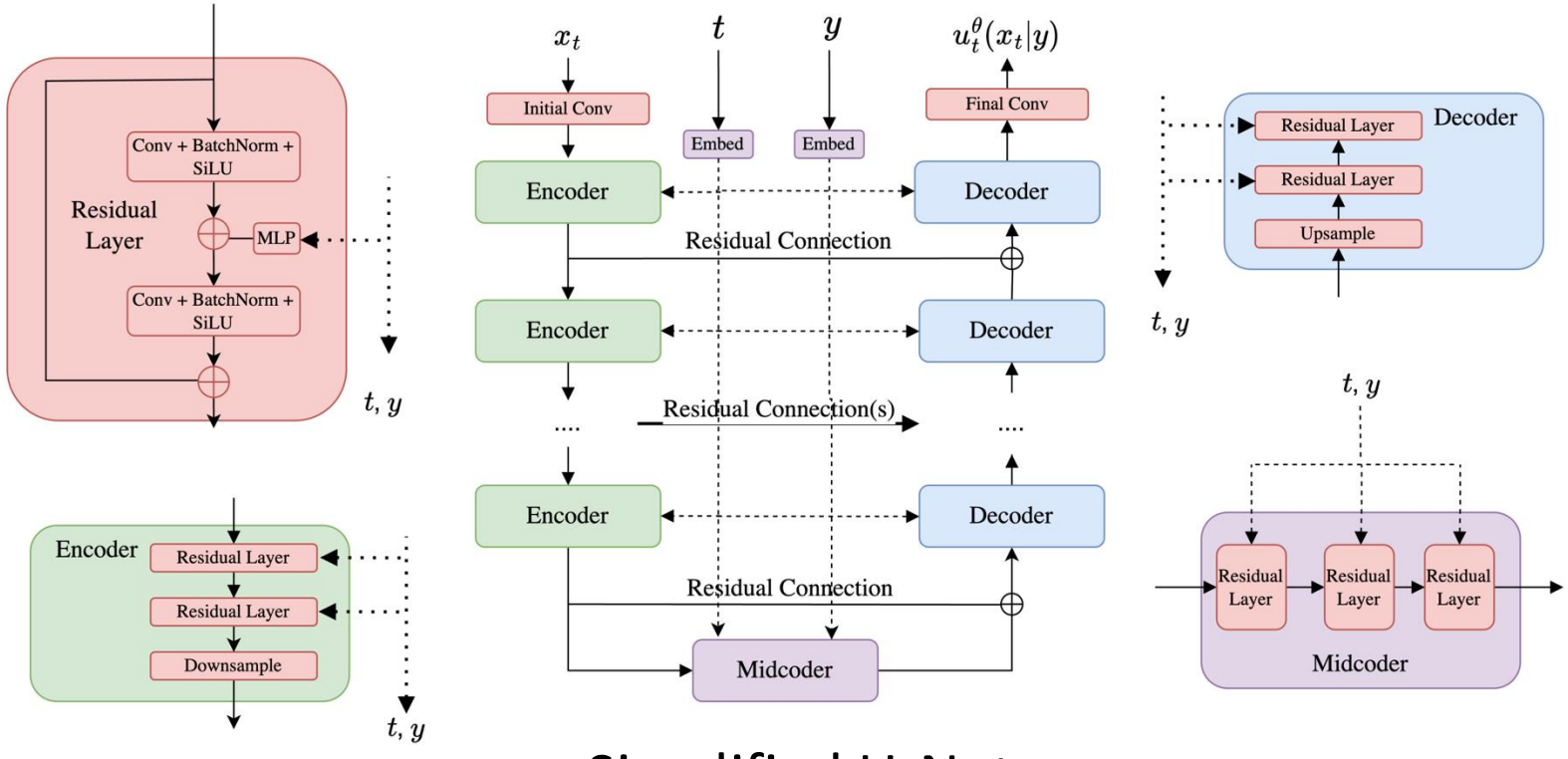
$$\square = (z,y) \sim p_{\text{data}}(z,y),\, t \sim \text{Unif}[0,1),\, x \sim p_t(\cdot|z), \text{replace } y = \varnothing \text{ with prob. } \eta$$

$$\tilde{u}_t(x|y) = (1-w)u_t^{\text{target}}(x|\varnothing) + w u_t^{\text{target}}(x|y).$$

# Metrics

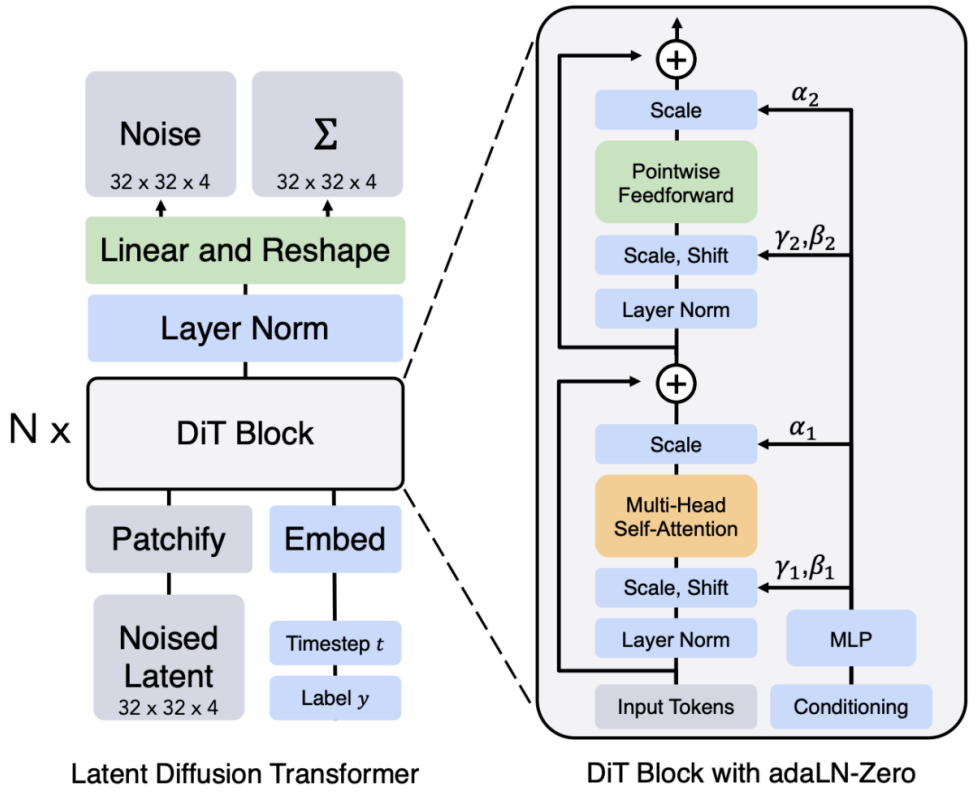
| Metric | What It Measures | Intuition | Good Value |
|---|---|---|---|
| **FID** (Fréchet Inception Distance) | How close generated images are to real images (quality + diversity). | "How much the fake images *feel* like real ones." | Lower is better (e.g., FID < 10 is strong). |
| **CLIP Score** | How well the generated image matches the text prompt. | "Did the model generate *what I asked for*?" | Higher is better. |
| **Precision / Recall** | Precision = image realism. Recall = variety compared to real images. | "Are the images realistic (precision) and varied (recall)?" | High for both. |
| **Aesthetic Score** | How beautiful or professional the images look. | "Would a human think this looks good?" | Higher is better. |

# Model Architectures



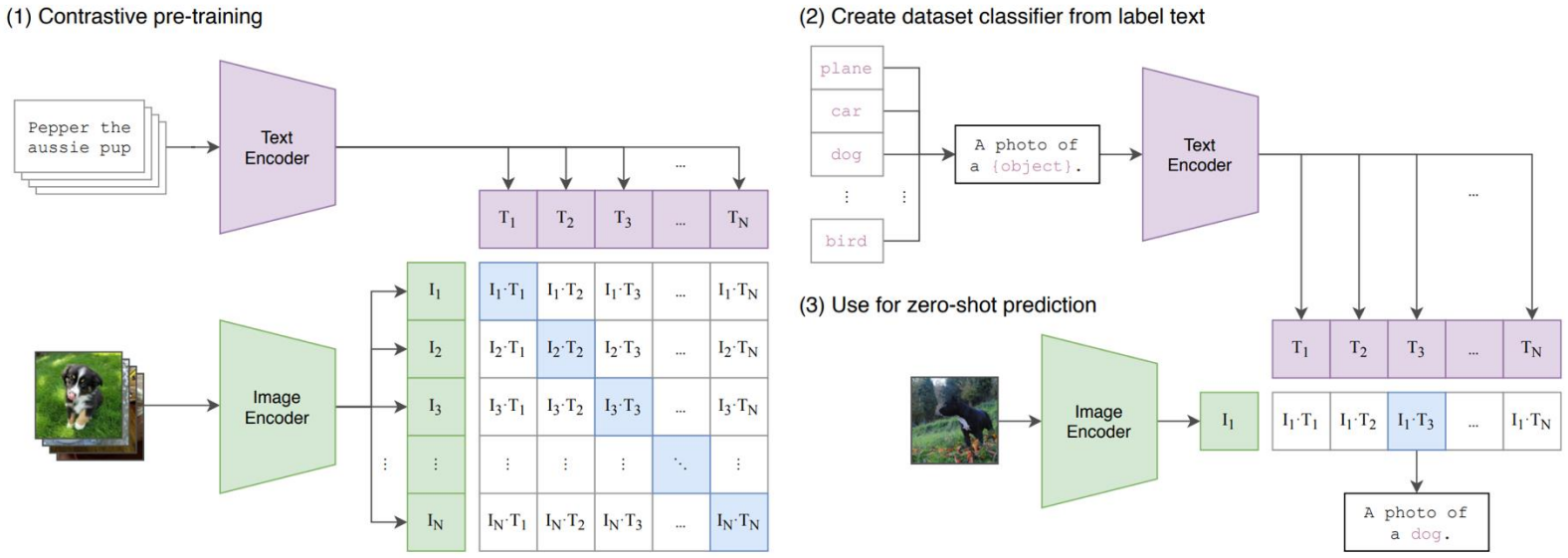

Simplified U-Net

# Diffusion Transformer



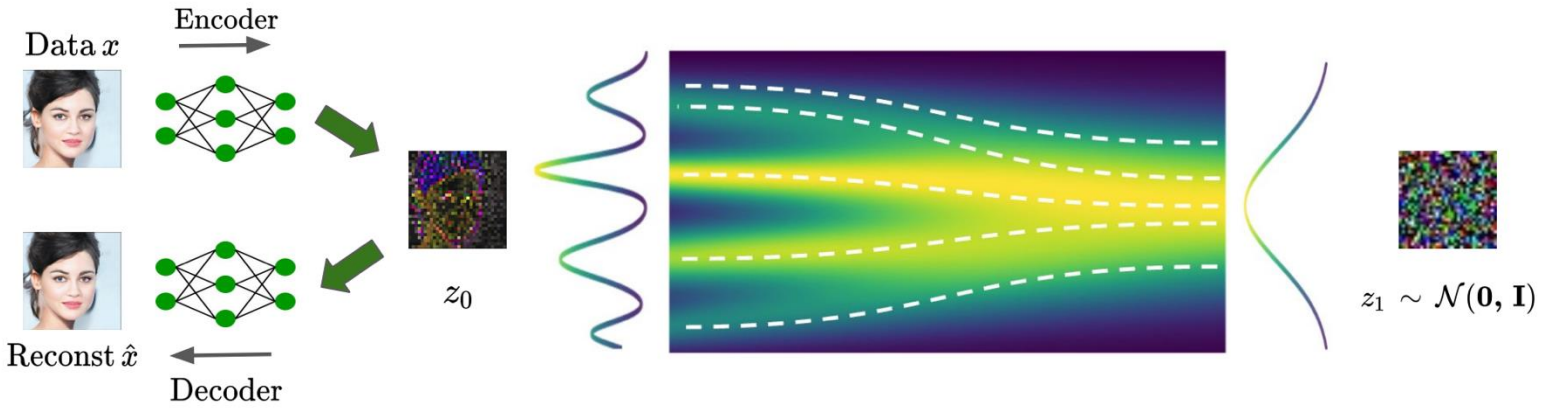

Scalable Diffusion Models with Transformers

# Conditioning with Language



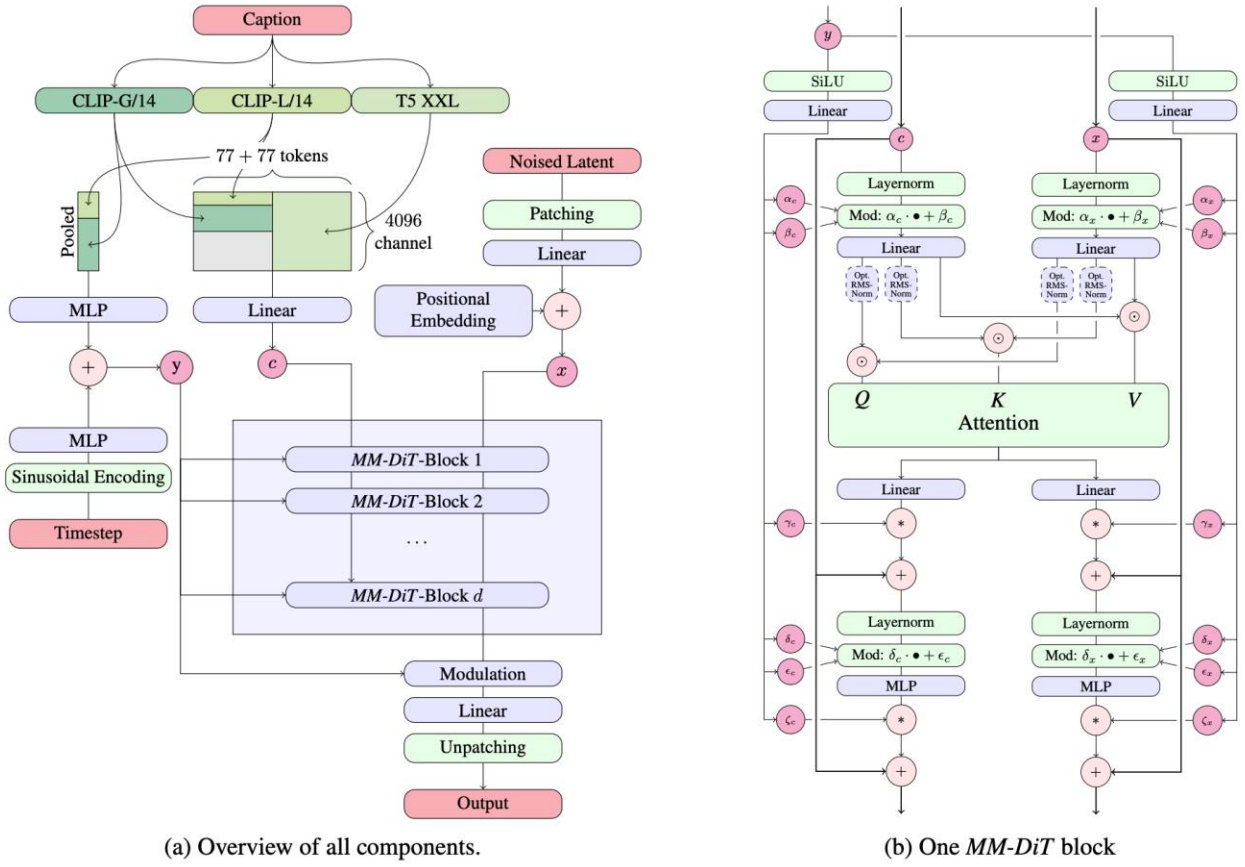

Learning Transferable Visual Models From Natural Language Supervision

# Diffusion in the Latent Space




Flow Matching in Latent Space

# Stable Diffusion 3

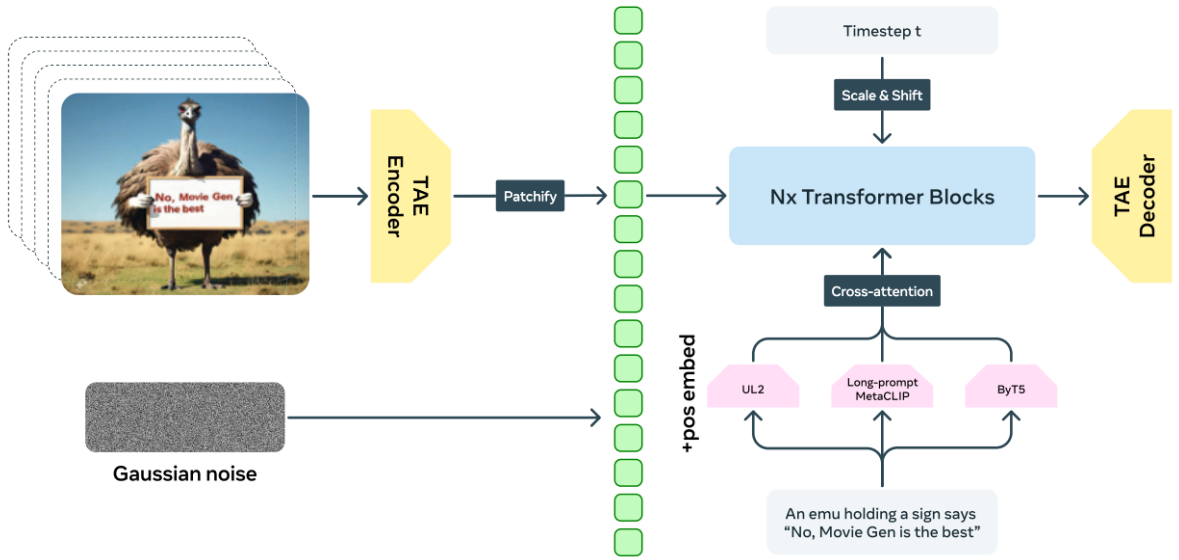



(a) Overview of all components.

(b) One *MM-DiT* block

Scaling Rectified Flow Transformers for High-Resolution Image Synthesis

# Meta: MovieGen




Movie Gen: A Cast of Media Foundation Models

# Tips to Train Flow Matching Models

1. Pick the right Loss Functions (L2 can be unstable)
2. Gradient Clipping maybe required
3. Use a Learning Rate Schedular (Liner Warmup + Cosine Decay)
4. Larger Batch Size is Preferred (Makes it more Stable)
5. Regularize

# Tips to Debug Flow Matching Models

1. Visualize Trajectories if Possible (Test on Toy examples)
2. Does time (t) change anything?
3. Overfit on one sample and then on one batch
4. Check Loss Value on different examples
5. Check the norms of the output vector fields
6. Add small noise during validation (check if output changes wildly)

# Further Reading


1. Flow Matching for Generative Modelling
2. Dreamcraft3d: Hierarchical 3d generation with bootstrapped diffusion prior
3. Score-based generative modeling through stochastic differential equation
4. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformer
5. Diffusion Models for Multi-Modal Generative Modelling
6. Multi-Track MusicLDM: Towards Versatile Music Generation with Latent Diffusion Model
7. https://lilianweng.github.io/posts/2021-07-11-diffusion-models/
8. https://yang-song.net/blog/