Image Generation

Neil Gong

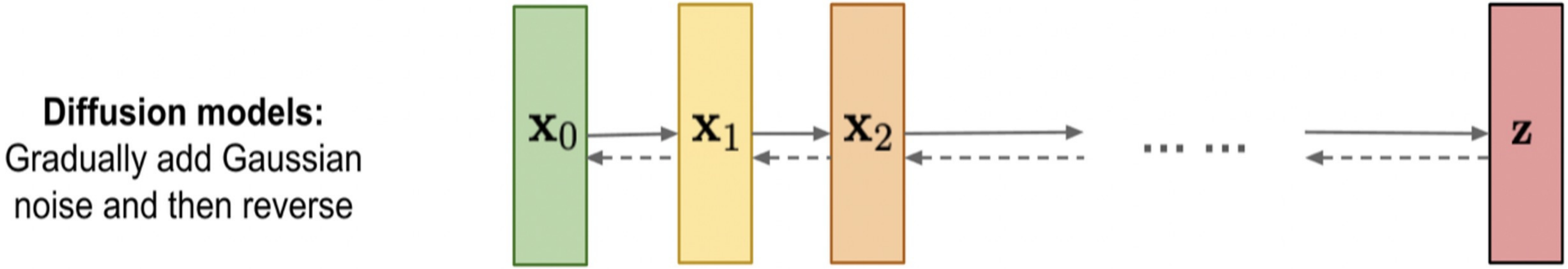
Slides adapted from Sarah Jabbour, Yiming Dou, Daniel Geng

Image generation methods

- Variational autoencoder
- GAN
- Diffusion model
- Visual AutoRegressive modeling

Diffusion

- Idea: Estimating and analyzing small step sizes is more tractable/easier than a single step from random noise to the learned distribution
- Convert a well-known and simple *base distribution* (like a Gaussian) to the *target (data) distribution* iteratively, with small step sizes, via a Markov chain:



- Markov chain: outlines the probability associated with a sequence of events occurring based on the state in the previous event.

Forward Process

- Noise added can be parameterized by:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}) \quad q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \quad \{\beta_t \in (0,1)\}_{t=1}^{T}$$

  Vary the parameters of the Gaussian according to a *noise schedule*

- You can prove with some math that as T approaches infinity, you eventually end up with an Isotropic Gaussian (i.e. pure random noise)
- Note: forward process is fixed

Reparameterization trick

Do you *have* to add noise *iteratively* to get to some timestep t? Nope!

Reverse process can be written in one step:

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

$$\alpha_t = 1 - \beta_t$$
$$\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$$

This will be useful during training!

Implementing Forward Process

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

$$\alpha_t = 1 - \beta_t$$
$$\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$$

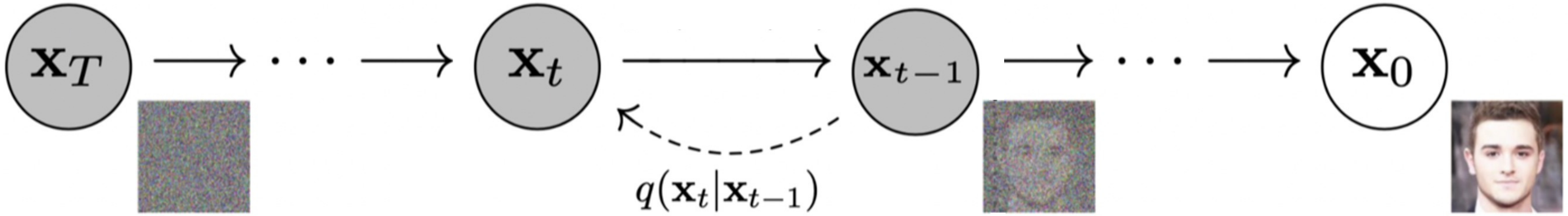
1. Sample an image from the dataset: 
2. Sample noise $\epsilon \sim N(0, \mathbf{I})$ (from a **standard** normal distribution)
3. Scale the image by $\sqrt{\overline{\alpha_t}}$: $\sqrt{\overline{\alpha_t}}\, x_0$

where
$$\alpha_t = 1 - \beta_t$$
$$\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$$

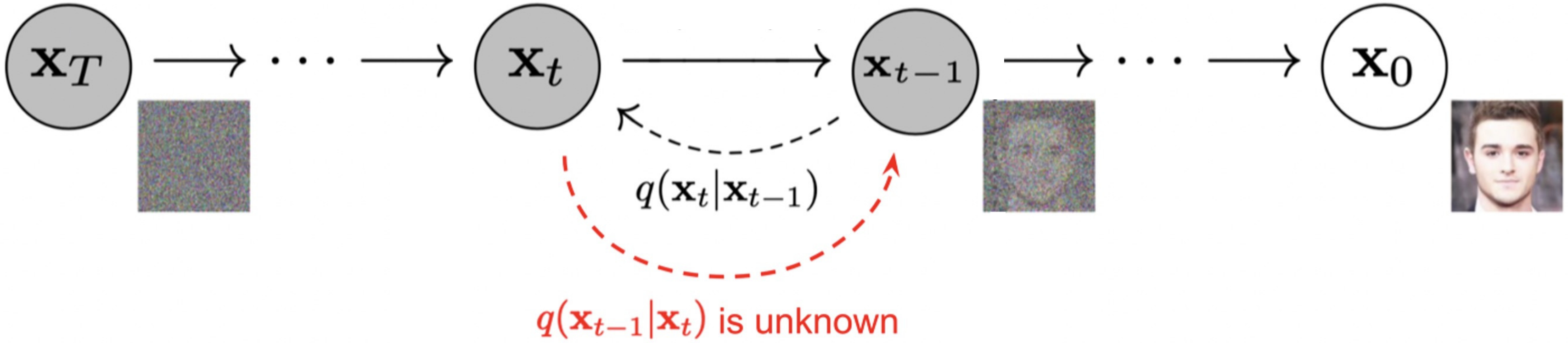
4. Add $\sqrt{1-\overline{\alpha_t}}\,\epsilon$: $\sqrt{\overline{\alpha_t}}\, x_0 + \sqrt{1-\overline{\alpha_t}}\,\epsilon$ $\longrightarrow$

x_t

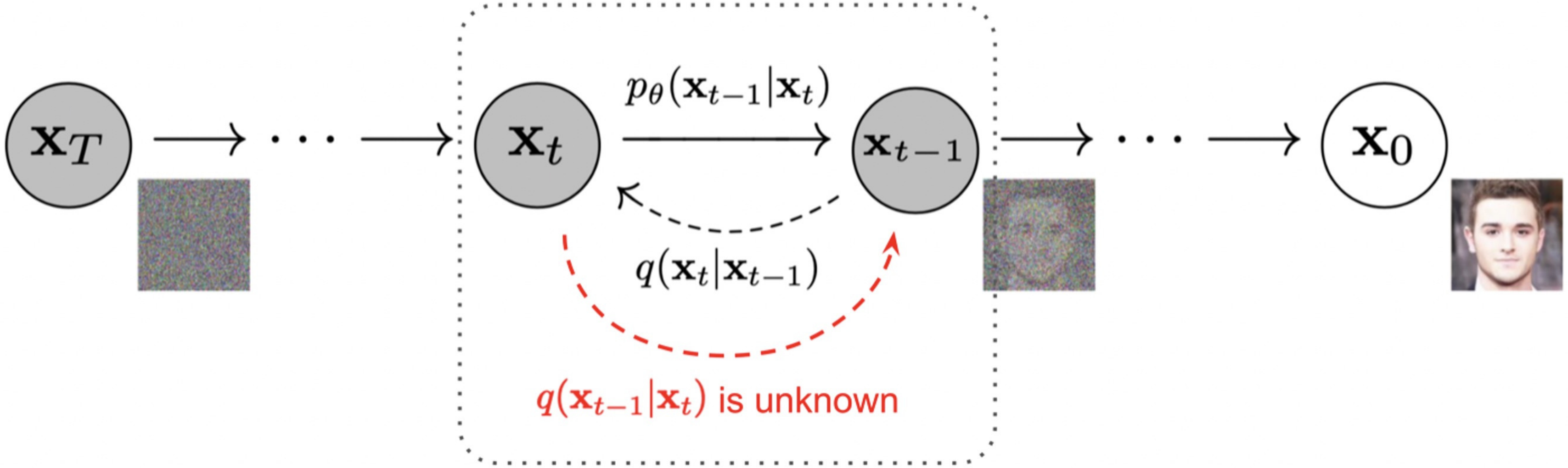
Reverse Process



Reverse Process

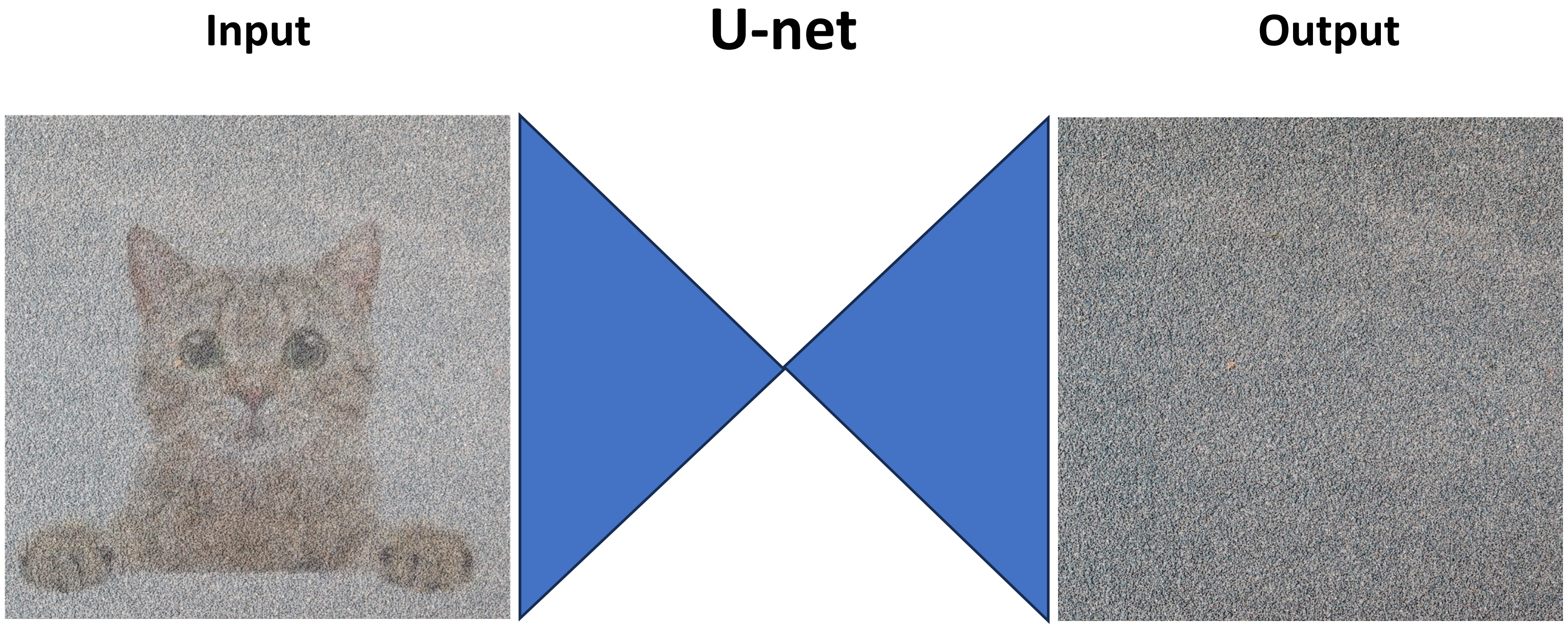


Reverse Process



- The goal of a diffusion model is to **learn** the reverse *denoising* process to iteratively **undo** the forward process
- In this way, the reverse process appears as if it is generating new data from random noise!

Neural Network that predicts noise



Algorithm 1 Training

repeat
  $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
  $t \sim \mathrm{Uniform}(\{1, \ldots, T\})$
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  Take gradient descent step on
$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$
until converged

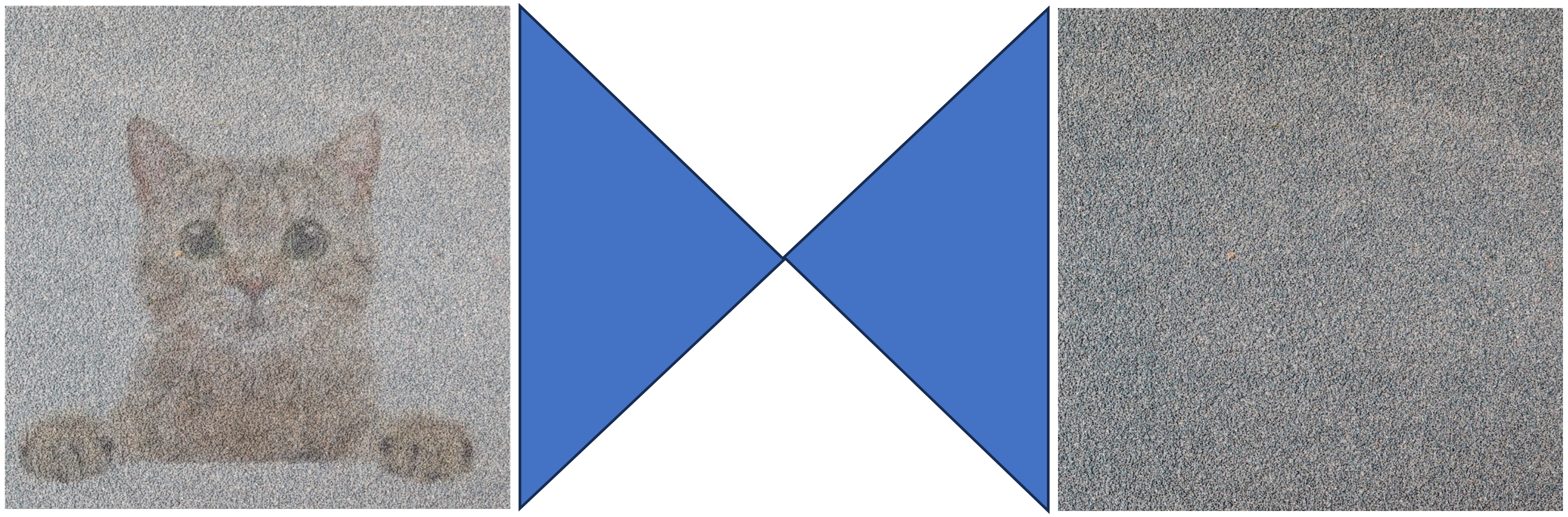
Algorithm 2 Sampling

$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, \ldots, 1$ **do**
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$$
end for
return $\mathbf{x}_0$

Denoising Diffusion Probabilistic Models (DDPM)

U-net Problem

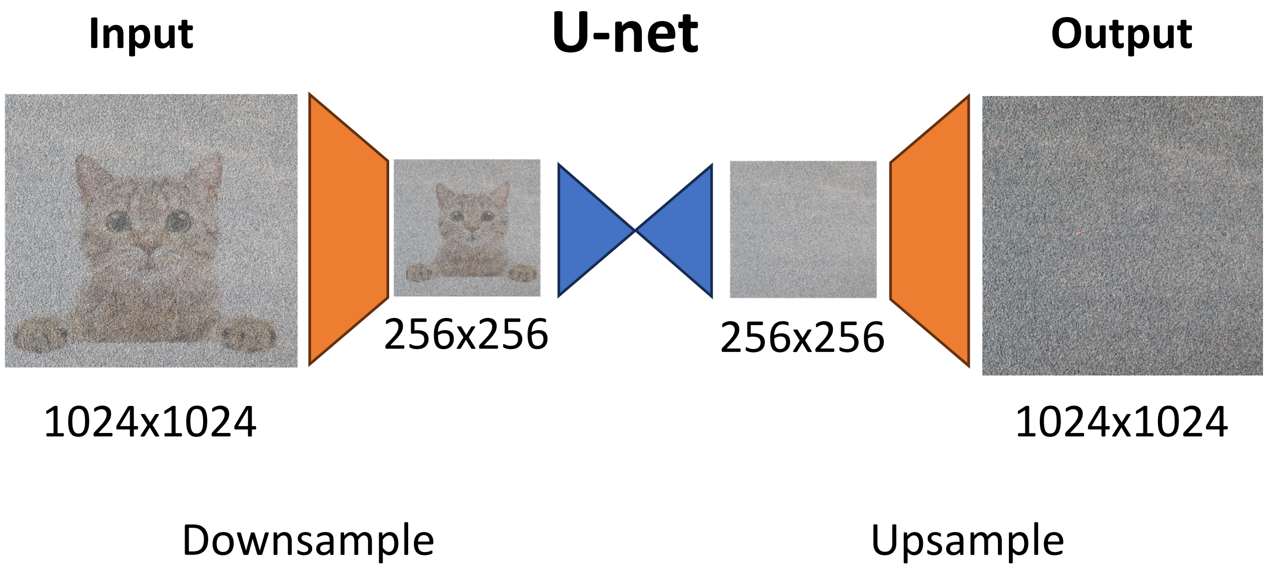
Input **U-net** **Output**



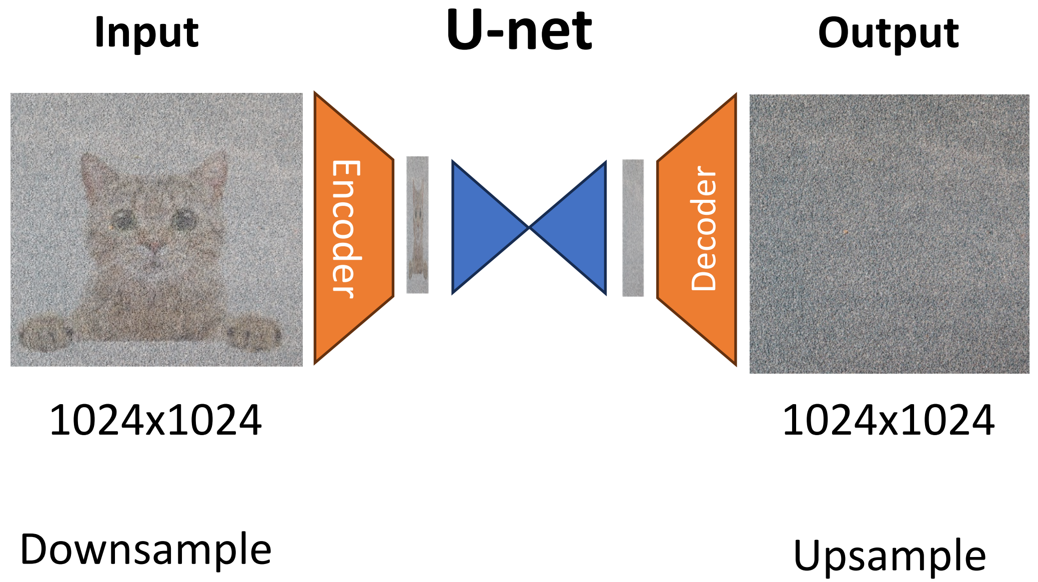
1024x1024 1024x1024

Problem: operating in the input space is very computationally expensive!

Option #1: Generate Low-Resolution + Upsample



Option #2: Generate in Latent Space

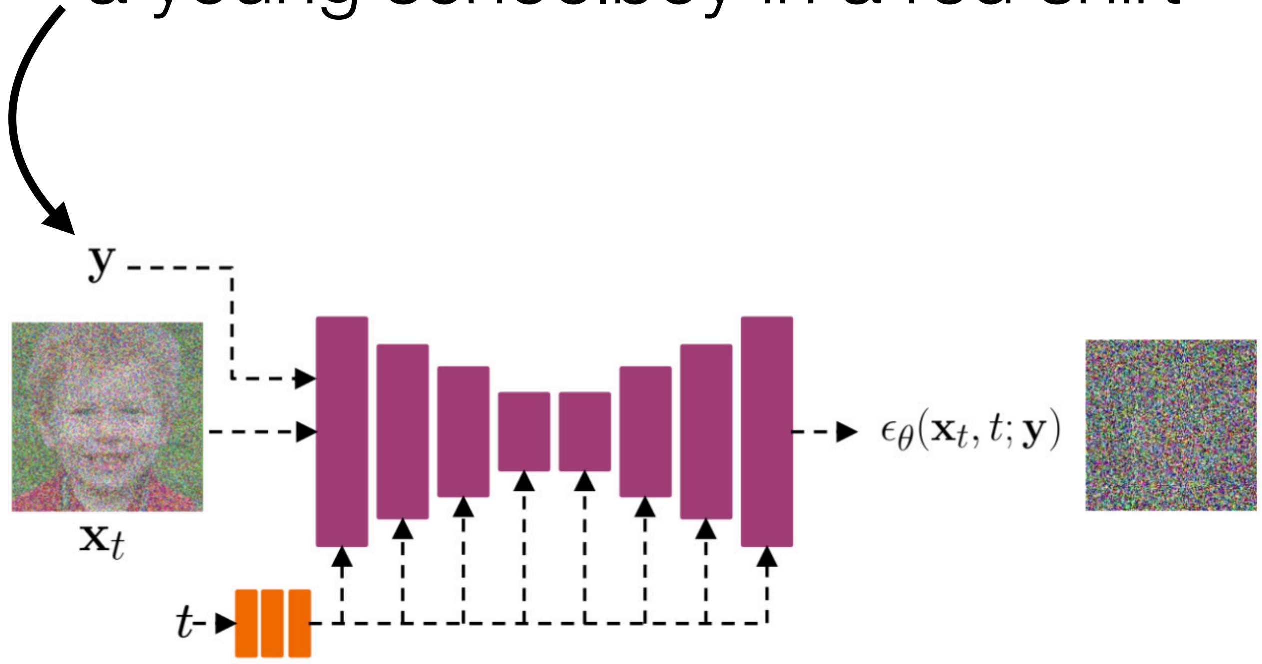


Controlling diffusion model

- Explicit conditioning
- Classifier-free guidance

Explicit Conditioning

“a young schoolboy in a red shirt”



Slide Credit: CVPR 2023 Diffusion Models Tutorial

Explicit Conditioning

How do we train this?

Use an Image-Text dataset (for example, LAION 5B)

Algorithm 1 Training

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

Unconditional

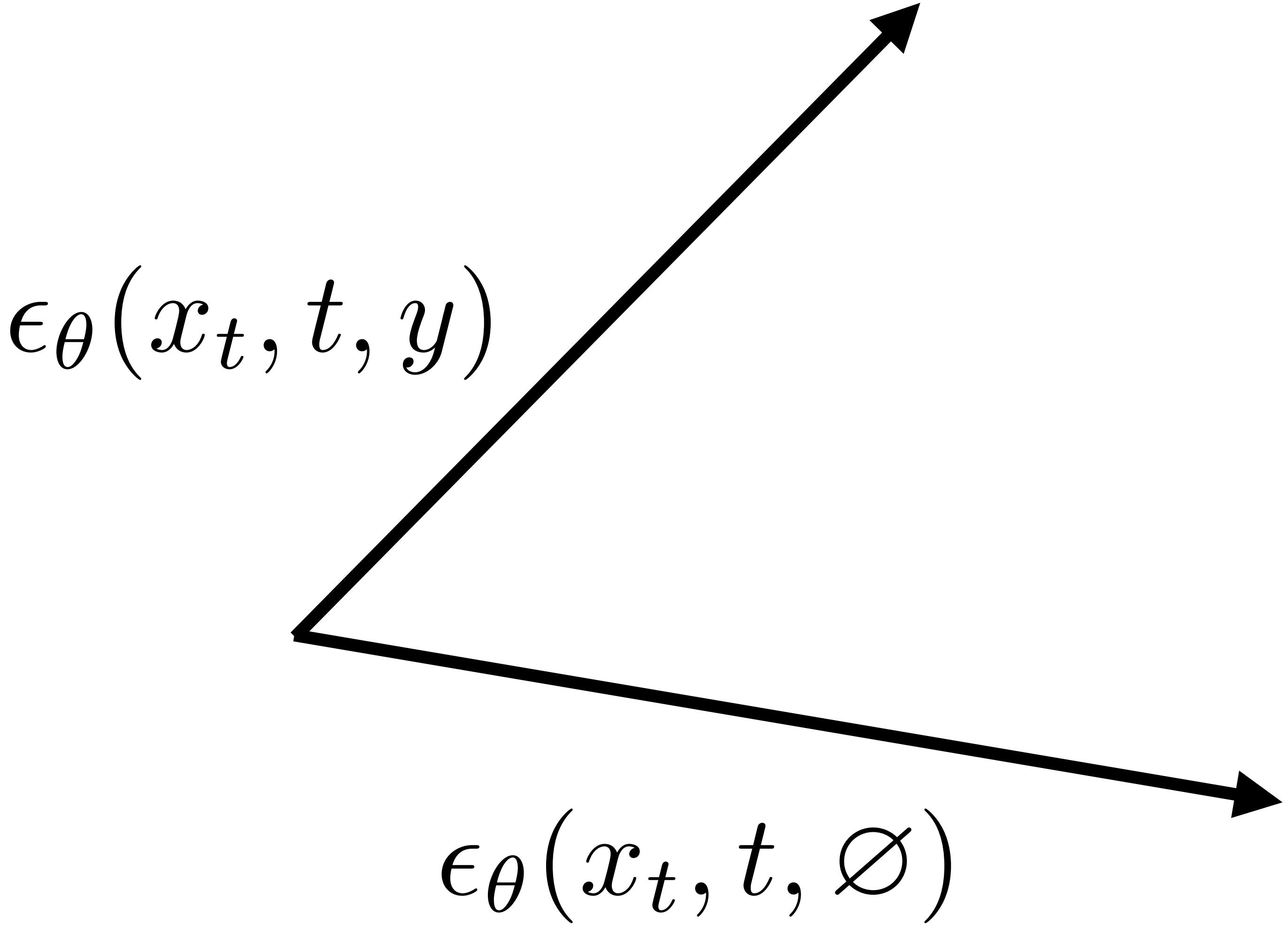
Classifier Free Guidance

Train an explicitly conditioned diffusion model: $\epsilon_\theta(x_t, t, y)$

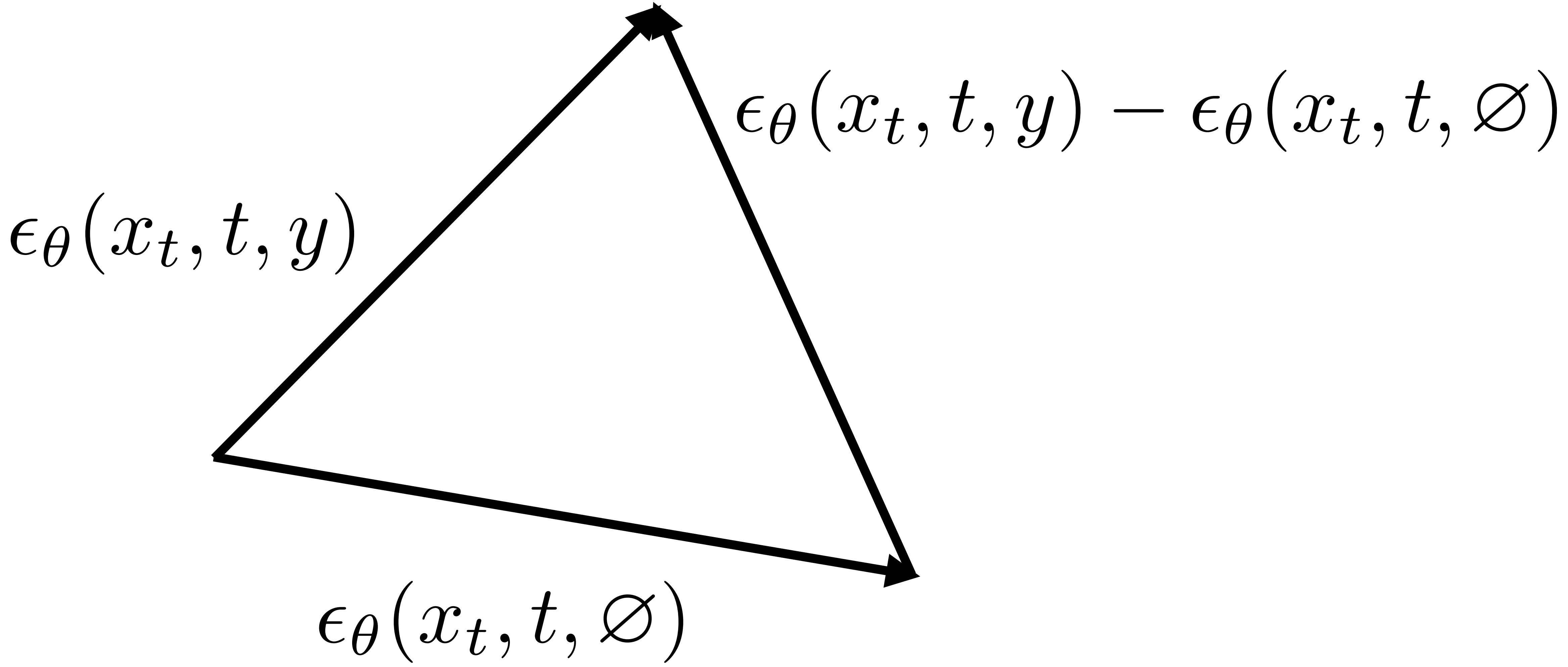
But also train it to be **unconditional**

We can do this with *conditioning dropout:* $\epsilon_\theta(x_t, t, \varnothing)$

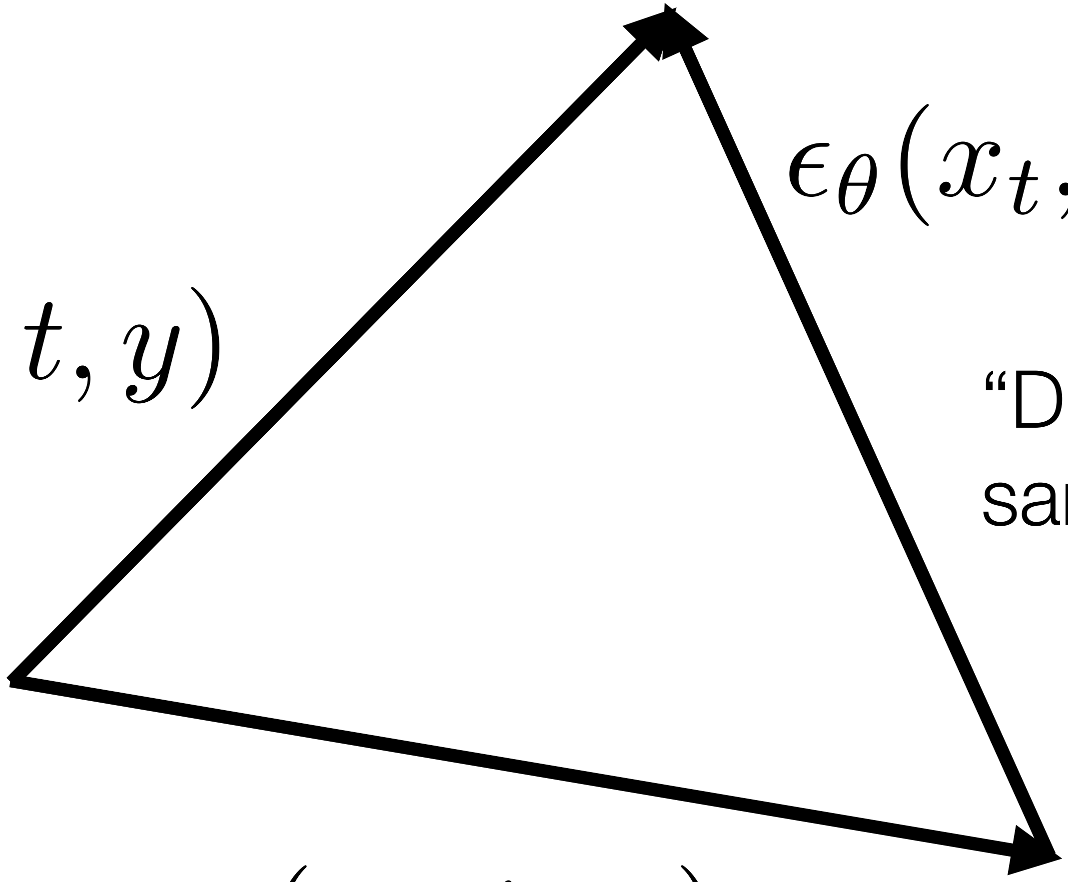
Classifier Free Guidance



Classifier Free Guidance

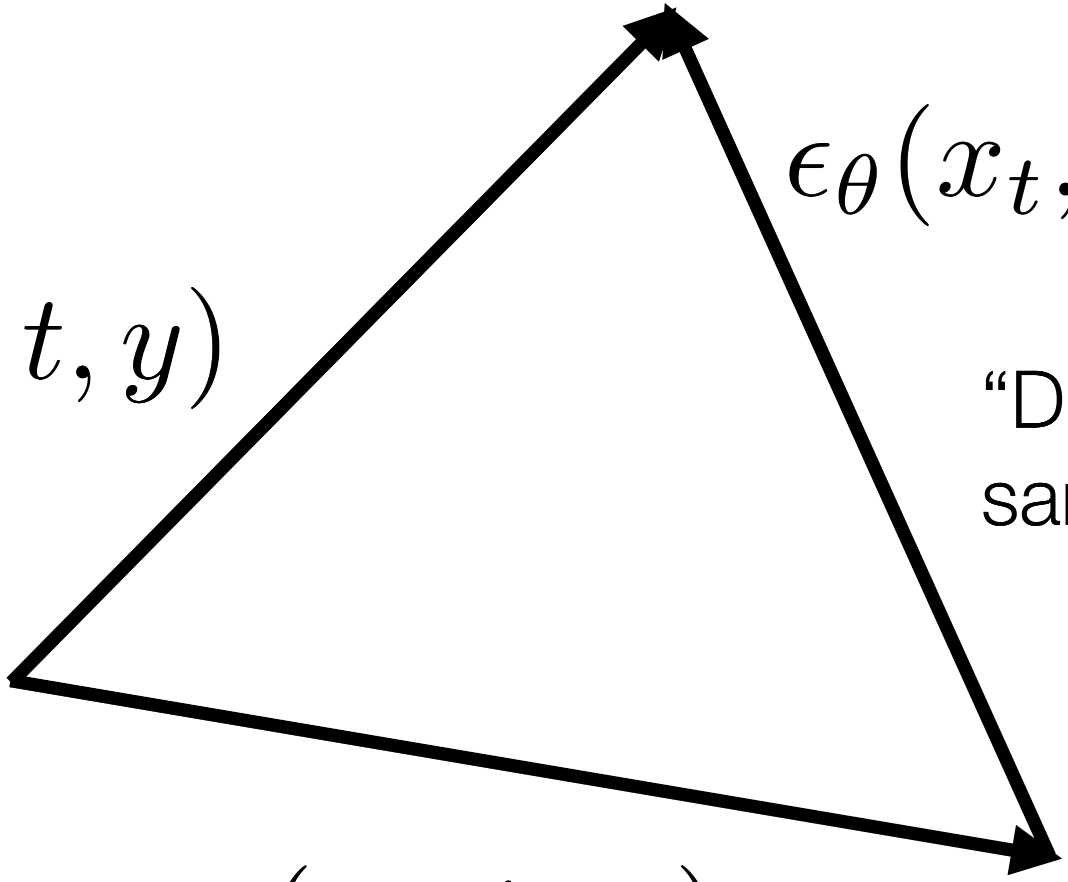


Classifier Free Guidance



"Direction" from unconditional sample to conditional sample

Classifier Free Guidance



"Direction" from unconditional sample to conditional sample

Use this as our guidance perturbation

Classifier Free Guidance

Our new noise estimate will then be:

$$\tilde{\epsilon}(x_t, t, y) = \epsilon_\theta(x_t, t, \varnothing) + \gamma\underbrace{(\epsilon_\theta(x_t, t, y) - \epsilon_\theta(x_t, t, \varnothing))}_{\text{“Direction” from unconditional to conditional}}$$

Visual AutoRegressive modeling

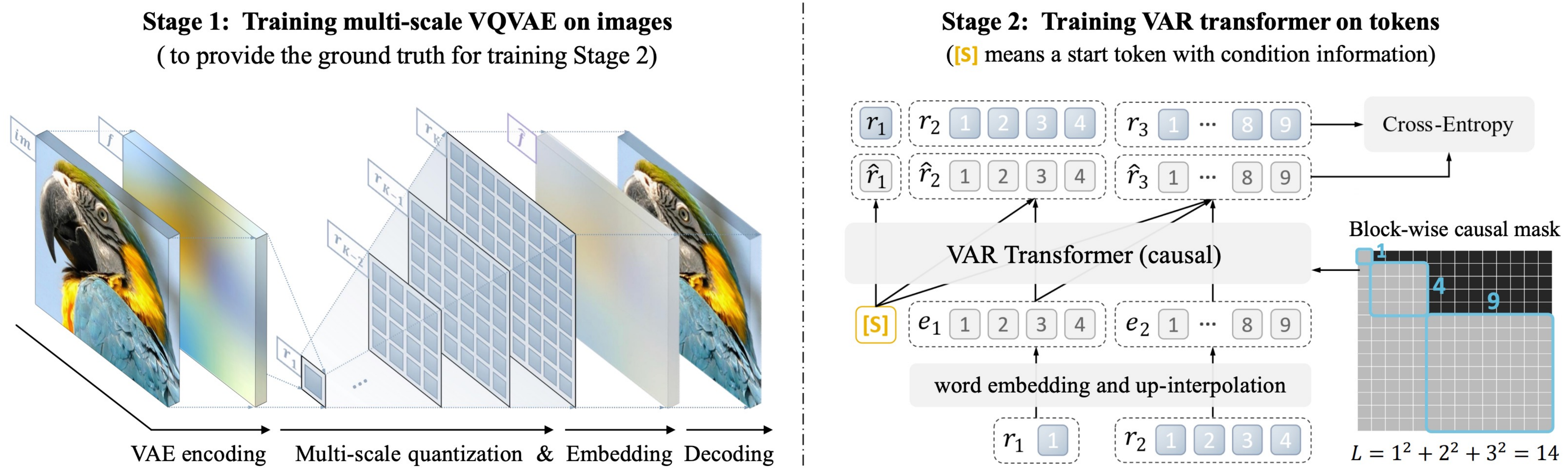


Figure 4: **VAR involves two separated training stages. Stage 1:** a multi-scale VQ autoencoder encodes an image into K token maps $R = (r_1, r_2, \ldots, r_K)$ and is trained by a compound loss (5). For details on "Multi-scale quantization" and "Embedding", check Algorithm 1 and 2. **Stage 2:** a VAR transformer is trained via next-scale prediction (6): it takes $([\mathtt{s}], r_1, r_2, \ldots, r_{K-1})$ as input to predict $(r_1, r_2, r_3, \ldots, r_K)$. The attention mask is used in training to ensure each r_k can only attend to $r_{\leq k}$. Standard cross-entropy loss is used.

"Visual Autoregressive Modeling: Scalable Image Generation via Next-Scale Prediction". In NeurIPS, 2024.

Image Editing

Input Image | **Edited Image** | **Input Image** | **Edited Image** | **Input Image** | **Edited Image**

Target Text: "A bird spreading wings" | "A person giving the thumbs up" | "A goat jumping over a cat"

"Imagic: Text-Based Real Image Editing with Diffusion Models". In CVPR, 2023.

$$\mathcal{L}(\mathbf{x}, \mathbf{e}, \theta) = \mathbb{E}_{t,\epsilon}\left[\|\epsilon - f_\theta(\mathbf{x}_t, t, \mathbf{e})\|_2^2\right]$$

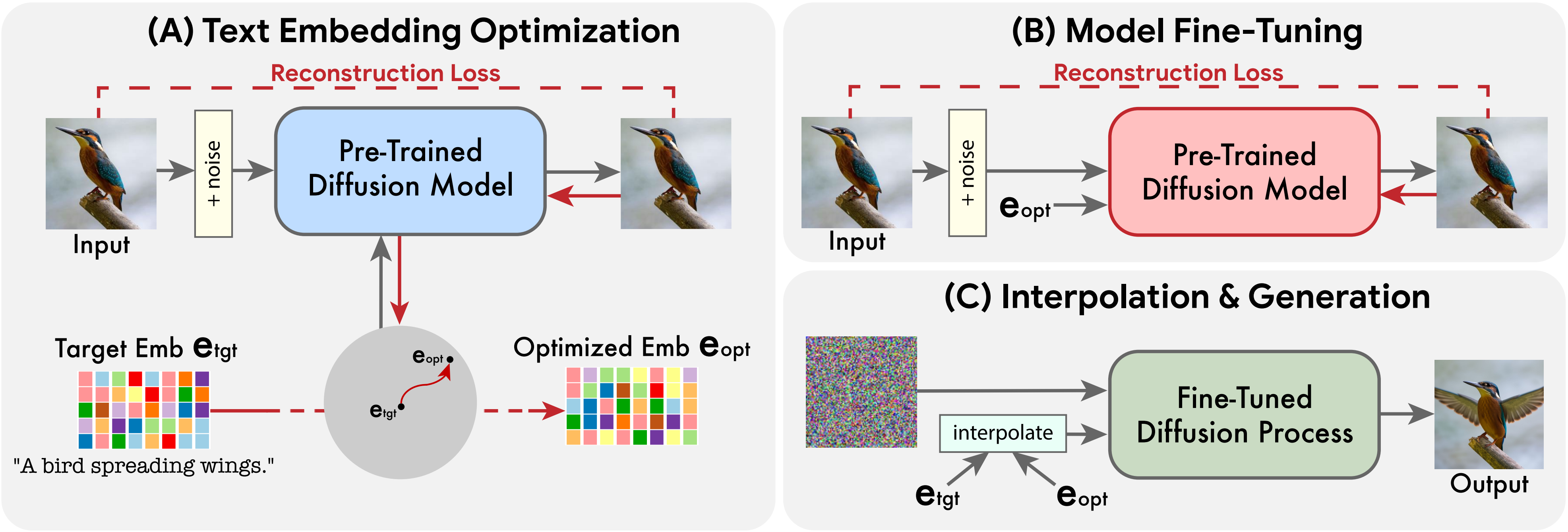


Figure 3. **Schematic description of *Imagic*.** *Given a real image and a target text prompt: (A) We encode the target text and get the initial text embedding $\mathbf{e}_{tgt}$, then optimize it to reconstruct the input image, obtaining $\mathbf{e}_{opt}$; (B) We then fine-tune the generative model to improve fidelity to the input image while fixing $\mathbf{e}_{opt}$; (C) Finally, we interpolate $\mathbf{e}_{opt}$ with $\mathbf{e}_{tgt}$ to generate the final editing result.*

$$\bar{\mathbf{e}} = \eta \cdot \mathbf{e}_{tgt} + (1 - \eta) \cdot \mathbf{e}_{opt}$$