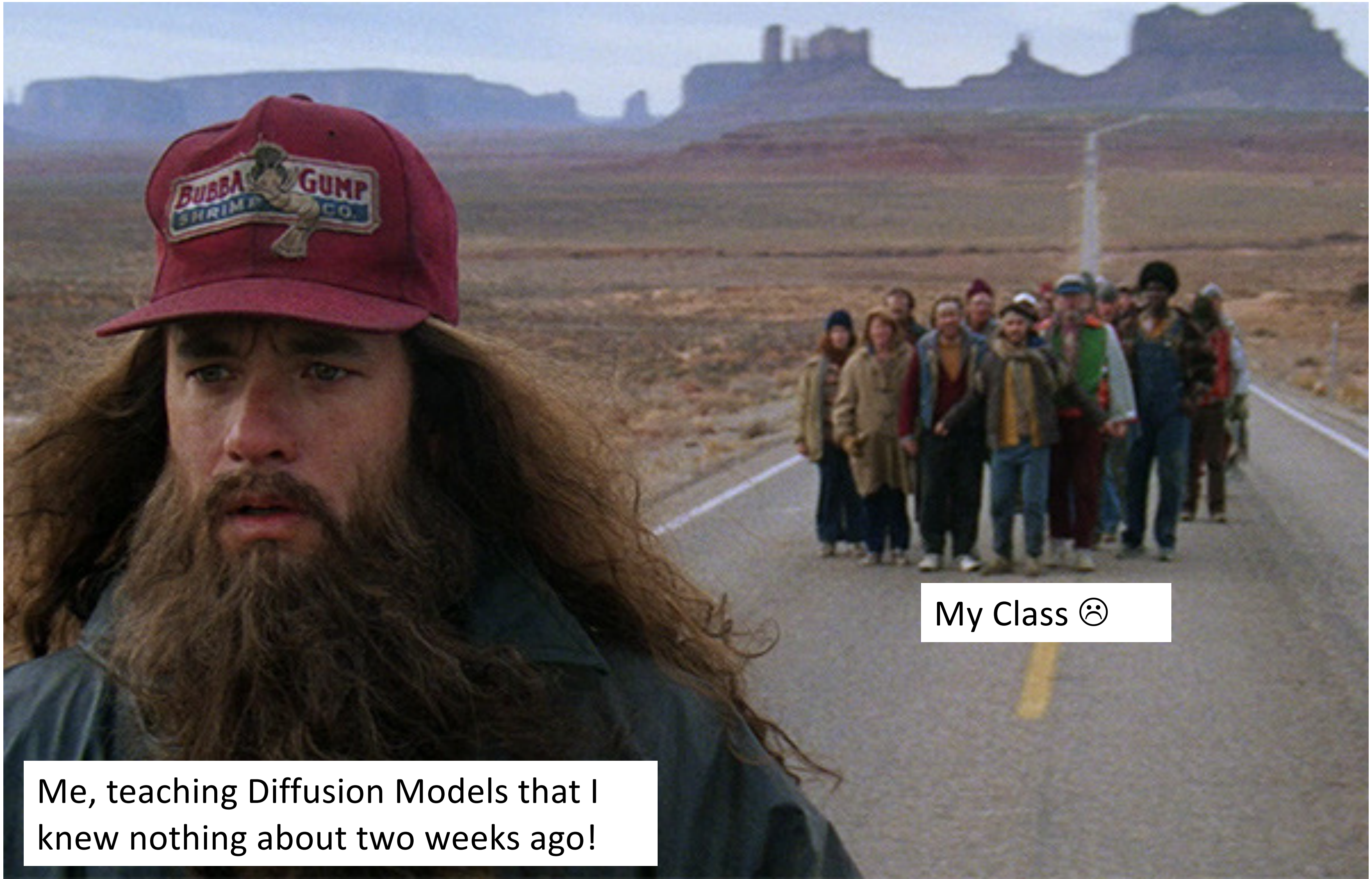
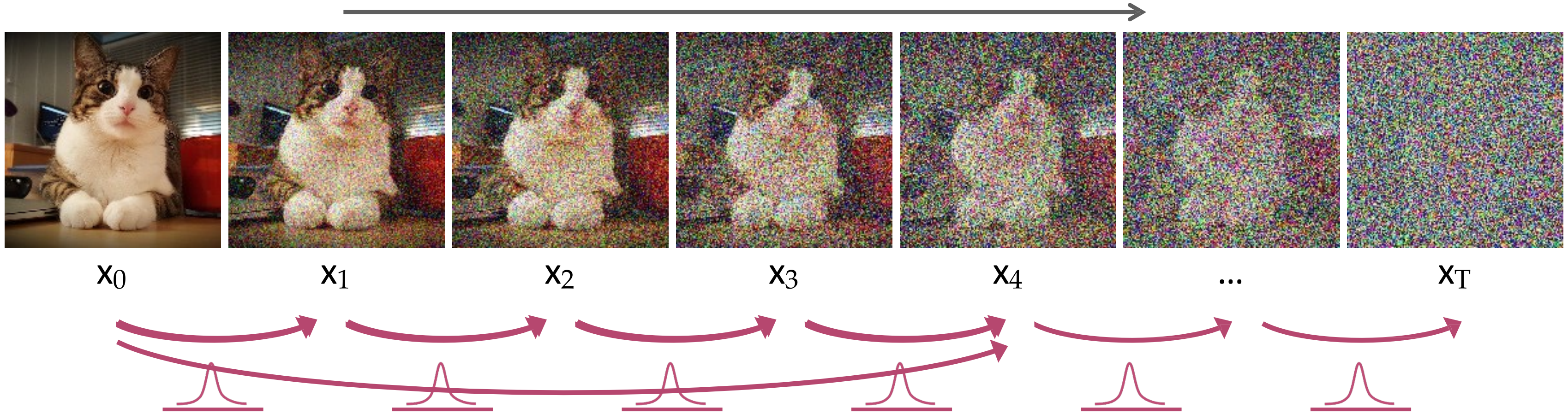
# Lecture X: Denoising Diffusion Models (cont.)



# Recap

# Forward Diffusion Process

Data




Noise

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \underbrace{\sqrt{1-\beta_t}\mathbf{x}_{t-1}}_{\text{mean}}, \underbrace{\beta_t\mathbf{I}}_{\text{variance}}) \quad \rightarrow$$

Sample: $x_t = \sqrt{1-\beta_t}\,x_{t-1} + \sqrt{\beta_t}\,\epsilon_{t-1}$

where, $\epsilon_{t-1} \sim \mathcal{N}(0, \mathbf{I})$

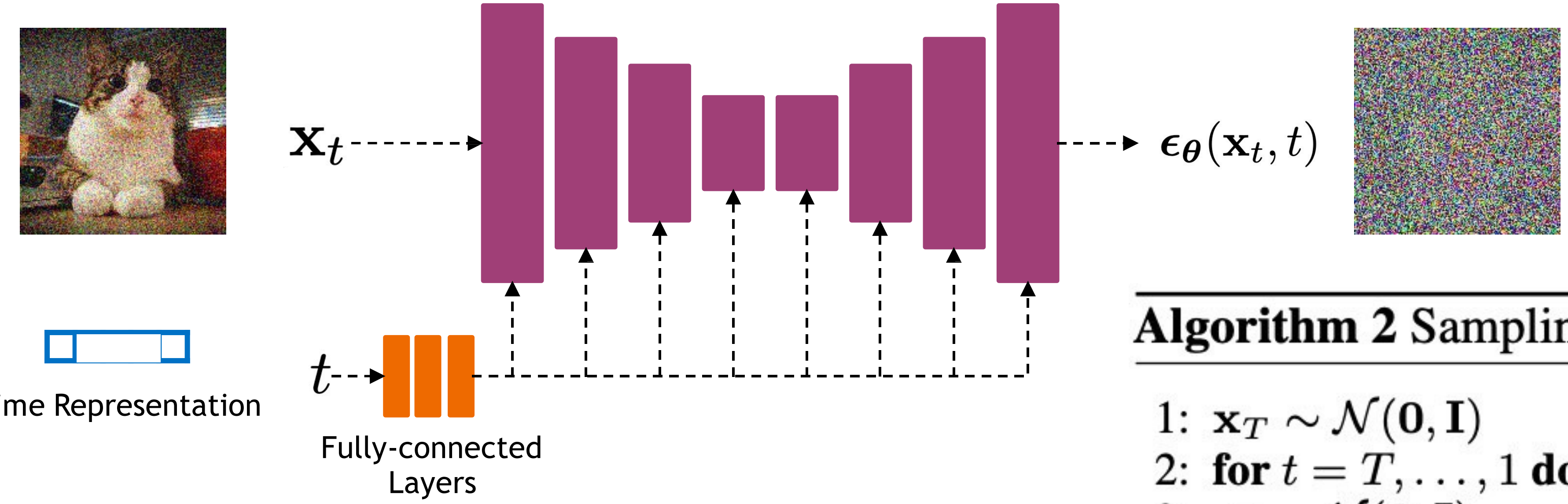
Define, $\bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$ $\rightarrow$ $q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}))$ (Diffusion Kernel)

For sampling: $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\ \mathbf{x}_0 + \sqrt{(1-\bar{\alpha}_t)}\ \epsilon$ where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

# Reverse Diffusion Process

$$L_{\text{simple}} = \mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0), \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), t \sim \mathcal{U}(1,T)} \left[ || \epsilon - \epsilon_\theta(\underbrace{\sqrt{\bar{\alpha}_t}\, \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\, \epsilon}_{\mathbf{x}_t}, t) ||^2 \right]$$

Diffusion models often use U-Net architectures with ResNet blocks and self-attention layers to represent $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$




---

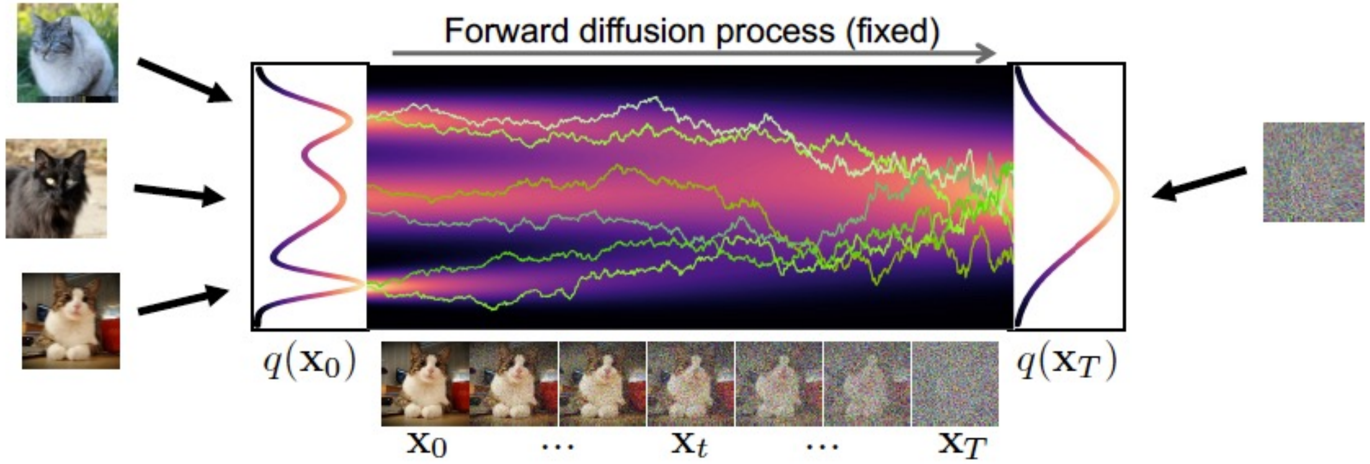
**Algorithm 2** Sampling

---

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \ldots, 1$ **do**
3: $\quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
4: $\quad \mathbf{x}_{t-1} = \boxed{\frac{1}{\sqrt{\alpha_t}} \left( \mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right)} + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

---

# The Generative Reverse Stochastic Differential Equation




**Forward Diffusion SDE:**

$$d\mathbf{x}_t = -\frac{1}{2}\beta(t)\mathbf{x}_t\, dt + \sqrt{\beta(t)}\, d\boldsymbol{\omega}_t$$

**Reverse Generative Diffusion SDE:**

$$d\mathbf{x}_t = \overbrace{\left[-\frac{1}{2}\beta(t)\mathbf{x}_t - \beta(t)\underbrace{\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t)}_{\text{“Score Function”}}\right] dt}^{\text{drift term}} + \overbrace{\sqrt{\beta(t)}\, d\bar{\boldsymbol{\omega}}_t}^{\text{diffusion term}}$$

➡ **Simulate reverse** How do we obtain the “Score Function”? **m noise!**

Song et al., *ICLR*, 2021
Anderson, in *Stochastic Processes and their Applications*, 1982

# Outline for today's class

- Theory:
  - How do we condition diffusion model?
  - How do people sample in practice? – DDIM sampling
  - How do we generate high-resolution images?
- Application:
  - Text to Image Generation
  - Image to Image Translation
  - Video Generation

# Outline for today's class

- Theory:
  - How do we condition diffusion model?
  - How do people sample in practice? – DDIM sampling
  - How do we generate high-resolution images?

- Application:
  - Text to Image Generation
  - Image to Image Translation
  - Video Generation

# Conditional diffusion models

Include condition as input to reverse process

Reverse process: $p_\theta(\mathbf{x}_{0:T}|\mathbf{c}) = p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c}), \quad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c}) = \mathcal{N}(\mathbf{x}_{t-1}; \boxed{\boldsymbol{\mu}_\theta(\mathbf{x}_t, t, \mathbf{c}), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t, \mathbf{c})})$

Variational upper bound: $L_\theta(\mathbf{x}_0|\mathbf{c}) = \mathbb{E}_q\left[L_T(\mathbf{x}_0) + \sum_{t>1} D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel \boxed{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c})) - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1, \mathbf{c})}\right].$

Incorporate conditions into U-Net

- Scalar conditioning: encode scalar as a vector embedding, simple spatial addition or adaptive group normalization layers.
- Image conditioning: channel-wise concatenation of the conditional image.
- Text conditioning: single vector embedding – spatial addition or adaptive group norm / a seq of vector embeddings - cross-attention.

# Classifier guidance

Using the gradient of a trained classifier as guidance

Recap: What is a score function?

**Forward Diffusion SDE:**

$$\mathrm{d}\mathbf{x}_t = -\frac{1}{2}\beta(t)\mathbf{x}_t\,\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\boldsymbol{\omega}_t$$

**Reverse Generative Diffusion SDE:**

$$\mathrm{d}\mathbf{x}_t = \overbrace{\left[-\frac{1}{2}\beta(t)\mathbf{x}_t - \beta(t)\underbrace{\nabla_{\mathbf{x}_t}\log q_t(\mathbf{x}_t)}_{\text{“Score Function”}}\right]}^{\text{drift term}}\mathrm{d}t + \overbrace{\sqrt{\beta(t)}\,\mathrm{d}\bar{\boldsymbol{\omega}}_t}^{\text{diffusion term}}$$

$$\min_{\boldsymbol{\theta}} \underbrace{\mathbb{E}_{t\sim\mathcal{U}(0,T)}}_{\substack{\text{diffusion}\\ \text{time } t}} \underbrace{\mathbb{E}_{\mathbf{x}_t\sim q_t(\mathbf{x}_t)}}_{\substack{\text{diffused}\\ \text{data } \mathbf{x}_t}} ||\underbrace{\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)}_{\substack{\text{neural}\\ \text{network}}} - \underbrace{\nabla_{\mathbf{x}_t}\log q_t(\mathbf{x}_t)}_{\substack{\text{score of}\\ \text{diffused data}\\ \text{(marginal)}}}||_2^2$$

# Classifier guidance

Using the gradient of a trained classifier as guidance

Applying Bayes rule to obtain conditional score function $\nabla_{x_t} log\ q_t(x_t/y)$

$$p(x \mid y) = \frac{p(y \mid x) \cdot p(x)}{p(y)}$$

$$\implies \log p(x \mid y) = \log p(y \mid x) + \log p(x) - \log p(y)$$

$$\implies \nabla_x \log p(x \mid y) = \nabla_x \log p(y \mid x) + \nabla_x \log p(x),$$

$$\nabla_x \log p_\gamma(x \mid y) = \nabla_x \log p(x) + \gamma \nabla_x \log p(y \mid x).$$ Classifier

Guidance scale: value >1 amplifies the influence of classifier signal.

$$p_\gamma(x \mid y) \propto p(x) \cdot p(y \mid x)^\gamma.$$

Slide Credits of guidance: https://benanne.github.io/2022/05/26/guidance.html

# Classifier guidance

Using the gradient of a trained classifier as guidance

$$\nabla_x \log p_\gamma(x \mid y) = \nabla_x \log p(x) + \gamma \nabla_x \log p(y \mid x).$$

Samples from an unconditional diffusion model with classifier guidance, for guidance scales 1.0 (left) and 10.0 (right), taken from Dhariwal & Nichol (2021).

# Classifier guidance

Using the gradient of a trained classifier as guidance

**Algorithm 1** Classifier guided diffusion sampling, given a diffusion model $(\mu_\theta(x_t), \Sigma_\theta(x_t))$, classifier $p_\phi(y|x_t)$, and gradient scale $s$.

Input: class label $y$, gradient scale $s$

$x_T \leftarrow$ sample from $\mathcal{N}(0, \mathbf{I})$

**for all** $t$ from $T$ to 1 **do**

$\quad \mu, \Sigma \leftarrow \mu_\theta(x_t), \Sigma_\theta(x_t)$

$\quad x_{t-1} \leftarrow$ sample from $\mathcal{N}(\underbrace{\mu}_{\text{Score model}} + s\Sigma \underbrace{\nabla_{x_t} \log p_\phi(y|x_t)}_{\text{Classifier gradient}}, \Sigma)$

**end for**

**return** $x_0$

- Train unconditional Diffusion model

- Take your favorite classifier, depending on the conditioning type

- During inference / sampling mix the gradients of the classifier with the predicted score function of the unconditional diffusion model.

# Classifier guidance

## Problems of classifier guidance

$$\nabla_x \log p_\gamma(x \mid y) = \nabla_x \log p(x) + \gamma \nabla_x \log p(y \mid x).$$

Classifier

Guidance scale: value >1 amplifies the influence of classifier signal.

- At each step of denoising the input to the classifier is a noisy image $x_t$ . Classifier is never trained on noisy image. So one needs to re-train classifier on noisy images! Can't use existing pre-trained classifiers.
- Most of the information in the input x is not relevant to predicting y, and as a result, taking the gradient of the classifier w.r.t. its input can yield arbitrary (and even adversarial) directions in input space.

# Classifier-free guidance

Get guidance by Bayes' rule on conditional diffusion models

$$p(y \mid x) = \frac{p(x \mid y) \cdot p(y)}{p(x)}$$

$$\implies \log p(y \mid x) = \log p(x \mid y) + \log p(y) - \log p(x)$$

$$\implies \boxed{\nabla_x \log p(y \mid x) = \nabla_x \log p(x \mid y) - \nabla_x \log p(x).}$$

We proved this in classifier guidance.

$$\nabla_x \log p_\gamma(x \mid y) = \nabla_x \log p(x) + \gamma \nabla_x \log p(y \mid x).$$

$$\nabla_x \log p_\gamma(x \mid y) = \nabla_x \log p(x) + \gamma \left(\nabla_x \log p(x \mid y) - \nabla_x \log p(x)\right),$$

$$\nabla_x \log p_\gamma(x \mid y) = (1-\gamma)\nabla_x \log p(x) + \gamma \nabla_x \log p(x \mid y).$$

Score function for unconditional diffusion model

Score function for conditional diffusion model

# Classifier-free guidance

Get guidance by Bayes' rule on conditional diffusion models

$$\nabla_x \log p_\gamma(x \mid y) = (1-\gamma)\nabla_x \log p(x) + \gamma \nabla_x \log p(x \mid y).$$

Score function for unconditional diffusion model

Score function for conditional diffusion model

This is a barycentric combination of the conditional and the unconditional score function. For $\gamma = 0$, we recover the unconditional model, and for $\gamma = 1$ we get the standard conditional model. But $\gamma > 1$ is where the magic happens. Below are some examples from OpenAI's GLIDE model[8], obtained using classifier-free guidance.

Two sets of samples from OpenAI's GLIDE model, for the prompt *'A stained glass window of a panda eating bamboo.'*, taken from their paper. Guidance scale 1 (no guidance) on the left, guidance scale 3 on the right.

In practice

$$\hat{\epsilon} = (1+\omega)\epsilon_\theta(x_t, y) - \omega\epsilon_\theta(x_t)$$

# Classifier-free guidance

Get guidance by Bayes' rule on conditional diffusion models

$$\hat{\epsilon} = (1+\omega)\epsilon_\theta(x_t, y) - \omega\epsilon_\theta(x_t)$$

$$\nabla_x \log p_\gamma(x \mid y) = (1-\gamma)\nabla_x \log p(x) + \gamma \nabla_x \log p(x \mid y).$$

Score function for unconditional diffusion model

Score function for conditional diffusion model

In practice:

- Train a conditional diffusion model p(x|y), with *conditioning dropout*: some percentage of the time, the conditioning information y is removed (10-20% tends to work well).
- The conditioning is often replaced with a special input value representing the absence of conditioning information.
- The resulting model is now able to function both as a conditional model p(x|y), and as an unconditional model p(x), depending on whether the conditioning signal is provided.
- During inference / sampling simply mix the score function of conditional and unconditional diffusion model based on guidance scale.

# Classifier-free guidance

Trade-off for sample quality and sample diversity

Non-guidance

Guidance scale = 1

Guidance scale = 3

Large guidance weight ($\omega$) usually leads to better individual sample quality but less sample diversity.

Ho & Salimans, “Classifier-Free Diffusion Guidance”, 2021.

# Classifier guidance

$$\nabla_x \log p_\gamma(x \mid y) = \nabla_x \log p(x) + \gamma \nabla_x \log p(y \mid x).$$

Guidance scale

Classifier

X Need to train a separate ”noise-robust” classifier + unconditional diffusion model.

X Gradient of the classifier w.r.t. input yields arbitrary values.

# Classifier-free guidance

$$\nabla_x \log p_\gamma(x \mid y) = (1-\gamma)\nabla_x \log p(x) + \gamma \nabla_x \log p(x \mid y).$$

Score function for unconditional diffusion model

Score function for conditional diffusion model

+ Train conditional & unconditional diffusion model jointly via drop-out.

+ All pixels in input receive equally ‘good’ gradients.

Rather than constructing a generative model from classifier, we construct a classifier from a generative model!

Most recent papers use classifier-free guidance! Very simple yet very powerful idea!

# Outline for today's class

- Theory:
  - How do we condition diffusion model?
  - How do people sample in practice? – DDIM sampling
  - How do we generate high-resolution images?

- Application:
  - Text to Image Generation
  - Image to Image Translation
  - Video Generation

# How to accelerate sampling process?

- During test time, one needs to run the diffusion model > 1000 steps, which is slow!
- If we naively sample less steps, the quality is worse.
- Can we somehow sample less steps and have the quality?
- DDIM sampling shows how this can be done! (Most recent papers use this trick)
- Note that: During training, we only train for t-1 to t and randomly sample t between [0,T]. We do no need to run the whole generation process

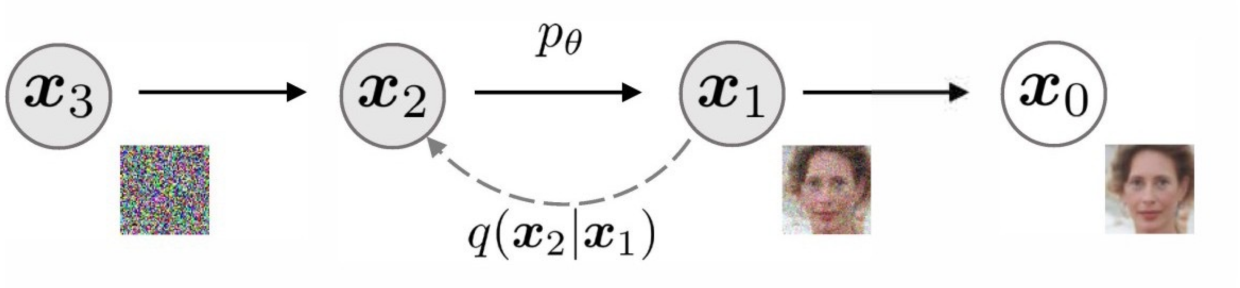
# DDIM sampling overview

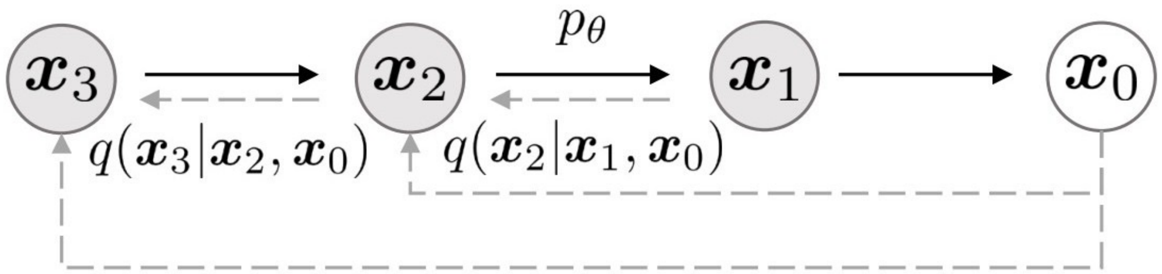
Key Idea:

- A regular diffusion model is Markovian process -> Generation at time t-1 only depends on time t, and independent of all other time stamps.
- Diffusion Markov process has optimal solution when we minimize this loss function:

$$L_{\text{simple}}(\theta) \coloneqq \mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\right\|^2\right]$$

- We want to use the trained Markovian Diffusion process at test time, but for faster sampling we define a non-Markovian process whose optimal solution is obtained by minimizing the same loss function as Markovian one!



# DDIM sampling overview

Markovian Diffusion (often called DDPM sampling – Denoising Diffusion Probabilistic Model):

- Given noisy image $X_t$ -> predict noise map $\epsilon_\theta(\mathbf{x}_t, t)$ -> subtract noise map from $X_t$ (with some weighting) to obtain $X_{t-1}$

$$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

$$\mathbf{x}_{t-1} = \boxed{\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t)\right)} + \sigma_t \mathbf{z}$$

Non-Markovian Diffusion (DDIM sampling):

- Given noisy image $X_t$ -> predict noise map $\epsilon_\theta(\mathbf{x}_t, t)$ -> generate clean image $X'_0$ -> add the predicted noise map to $X'_0$ to obtain $X_{t-1}$.

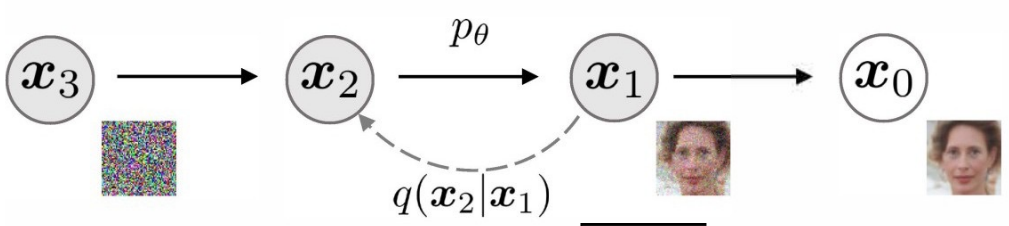
$$\boldsymbol{x}_{t-1} = \sqrt{\alpha_{t-1}} \underbrace{\left(\frac{\boldsymbol{x}_t - \sqrt{1-\alpha_t}\boxed{\epsilon_\theta^{(t)}(\boldsymbol{x}_t)}}{\sqrt{\alpha_t}}\right)}_{\text{" predicted } \boldsymbol{x}_0\text{"}} + \underbrace{\sqrt{1-\alpha_{t-1}-\sigma_t^2} \cdot \boxed{\epsilon_\theta^{(t)}(\boldsymbol{x}_t)}}_{\text{"direction pointing to } \boldsymbol{x}_t\text{"}} + \underbrace{\sigma_t \epsilon_t}_{\text{random noise}}$$

- Different choice of $\sigma$ results in different generative process without re-training the model

- When $\sigma = 0$ for all t, we have a deterministic generative process, with randomness from only t=T (the last step).

# Denoising Diffusion Implicit Models (DDIM)

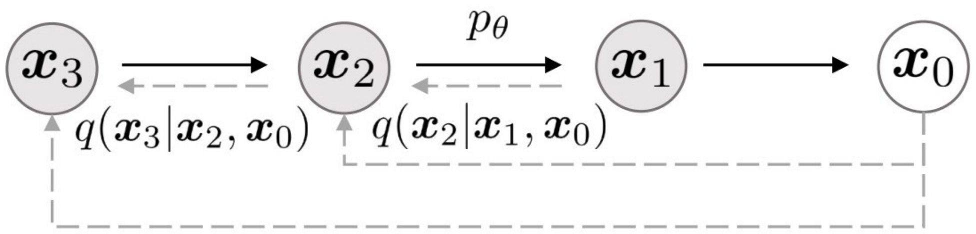
## Why can we do this?




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

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \sigma_t^2\mathbf{I})$$

Regular diffusion model




Define a family of forward processes that meets the above requirement:

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}\left(\sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}-\tilde{\sigma}_t^2} \cdot \frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}}, \tilde{\sigma}_t^2\mathbf{I}\right)$$

The corresponding reverse process is

$$p(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}\left(\sqrt{\bar{\alpha}_{t-1}}\hat{\mathbf{x}}_0 + \sqrt{1-\bar{\alpha}_{t-1}-\tilde{\sigma}_t^2} \cdot \frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\hat{\mathbf{x}}_0}{\sqrt{1-\bar{\alpha}_t}}, \tilde{\sigma}_t^2\mathbf{I}\right)$$

$$:= (\boldsymbol{x}_t - \sqrt{1-\alpha_t} \cdot \epsilon_\theta^{(t)}(\boldsymbol{x}_t))/\sqrt{\alpha_t}.$$

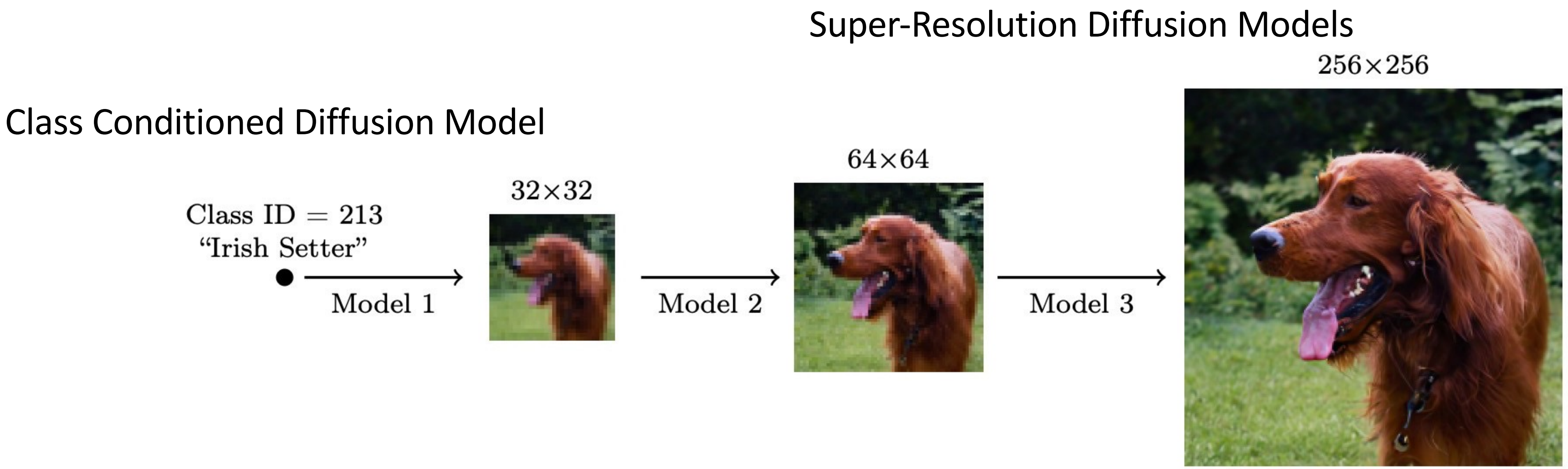
This has same loss function as the regular/Markovian diffusion model. Thus we can use this strategy without re-training.

# Outline for today’s class

- Theory:
  - How do we condition diffusion model?
  - How do people sample in practice? – DDIM sampling
  - How do we generate high-resolution images?
- Application:
  - Text to Image Generation
  - Image to Image Translation
  - Video Generation

# Cascaded generation

## Pipeline




Similar cascaded / multi-resolution image generation also exist in GAN (Big-GAN & StyleGAN)

Cascaded Diffusion Models outperform Big-GAN in FID and IS and VQ-VAE2 in Classification Accuracy Score.


[Ho et al., "Cascaded Diffusion Models for High Fidelity Image Generation", 2021.]

# Noise conditioning augmentation

Reduce compounding error

Problem:

- During training super-resolution models are trained on original low-res images from the dataset.
- During inference, these low-res images are generated by class conditioned diffusion model, which has artifacts and poor quality than original low-res images used for training.

Mismatch issue

Solution: Noise conditioning augmentation.

- During training, add varying amounts of Gaussian noise (or blurring by Gaussian kernel) to the low-res images.
- During inference, sweep over the optimal amount of noise added to the low-res images.
- BSR-degradation process: applies JPEG compressions noise, camera sensor noise, different image interpolations for downsampling, Gaussian blur kernels and Gaussian noise in a random order to an image.

Ho et al., "Cascaded Diffusion Models for High Fidelity Image Generation", 2021.
Nichol et al., "GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models", 2021.

# Outline for today's class

- Theory:
  - How do we condition diffusion model?
  - How do people sample in practice? – DDIM sampling
  - How do we generate high-resolution images?

- Application:
  - Text to Image Generation
  - Image to Image Translation
  - Video Generation

# GLIDE

OpenAI

- A 64x64 base model + a 64x64 → 256x256 super-resolution model.
- Tried classifier-free and CLIP guidance. Classifier-free guidance works better than CLIP guidance.

"a hedgehog using a calculator" | "a corgi wearing a red bowtie and a purple party hat" | "robots meditating in a vipassana retreat" | "a fall landscape with a small cottage next to a lake"

Samples generated with classifier-free guidance (256x256)

Nichol et al., "GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models", 2021.

# CLIP guidance

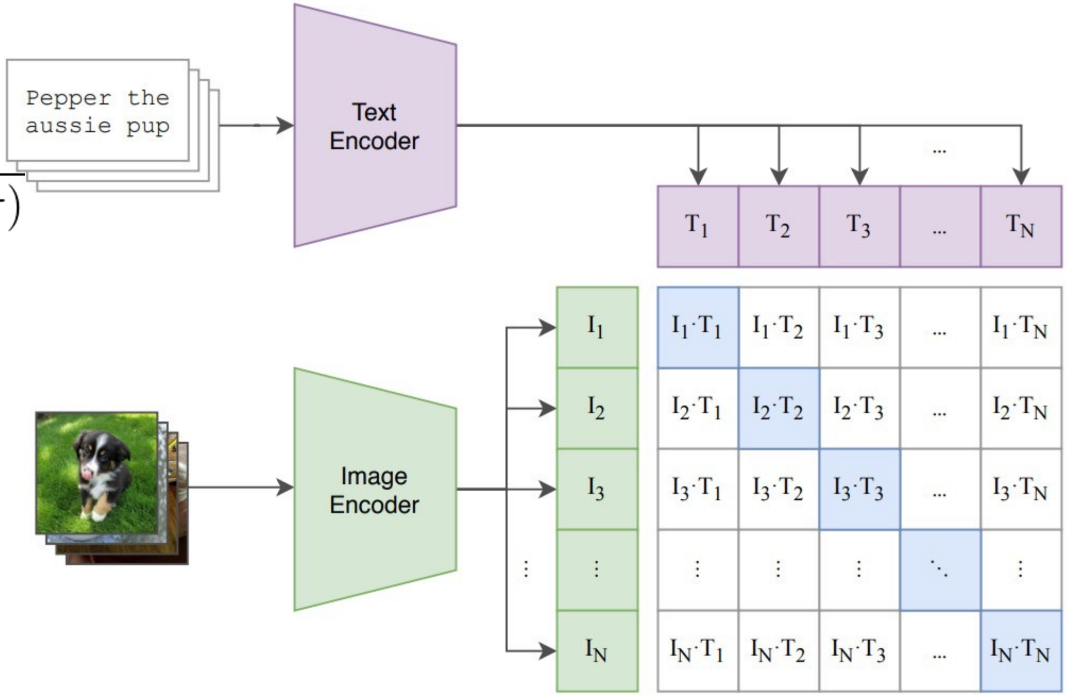
## What is a CLIP model?

- Trained by contrastive cross-entropy loss:

$$-\log \frac{\exp(f(\mathbf{x}_i)\cdot g(\mathbf{c}_j)/\tau)}{\sum_k \exp(f(\mathbf{x}_i)\cdot g(\mathbf{c}_k)/\tau)} - \log \frac{\exp(f(\mathbf{x}_i)\cdot g(\mathbf{c}_j)/\tau)}{\sum_k \exp(f(\mathbf{x}_k)\cdot g(\mathbf{c}_j)/\tau)}$$

- The optimal value of $f(\mathbf{x})\cdot g(\mathbf{c})$ is

$$\log \frac{p(\mathbf{x},\mathbf{c})}{p(\mathbf{x})p(\mathbf{c})} = \log p(\mathbf{c}|\mathbf{x}) - \log p(\mathbf{c})$$




Radford et al., "Learning Transferable Visual Models From Natural Language Supervision", 2021.
Nichol et al., "GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models", 2021.

# CLIP guidance

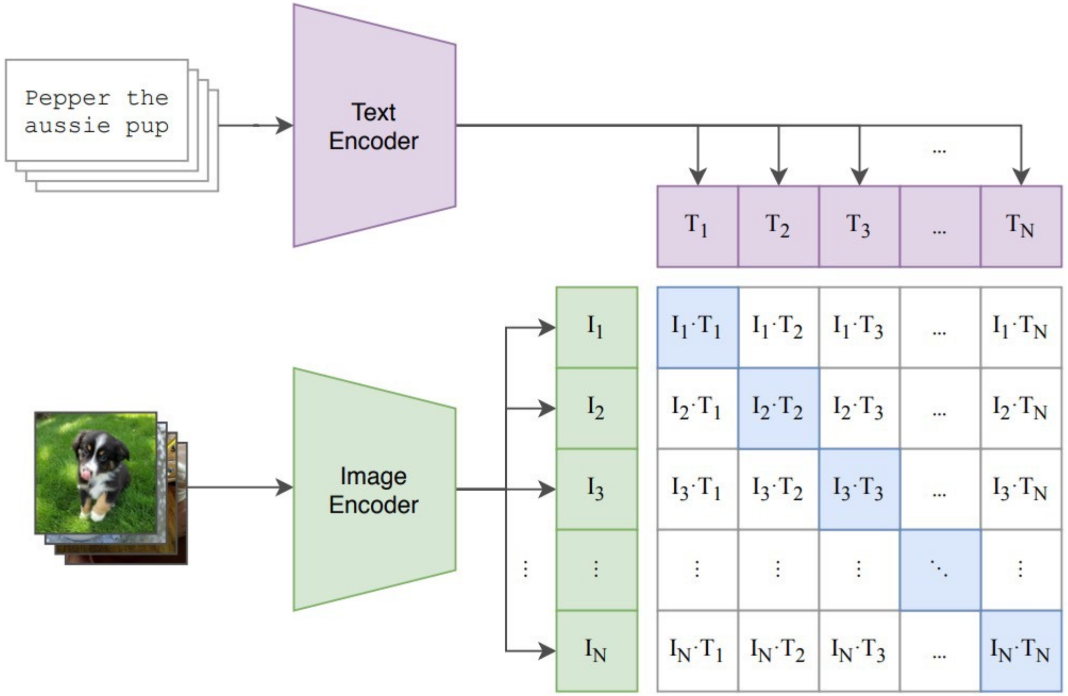
Replace the classifier in classifier guidance with a CLIP model

- Sample with a modified score:

$$\nabla_{\mathbf{x}_t}[\log p(\mathbf{x}_t|\mathbf{c}) + \omega \log p(\mathbf{c}|\mathbf{x}_t)]$$

$$= \nabla_{\mathbf{x}_t}[\log p(\mathbf{x}_t|\mathbf{c}) + \omega\underbrace{(\log p(\mathbf{c}|\mathbf{x}_t) - \log p(\mathbf{c}))}_{\text{CLIP model}}]$$

$$= \nabla_{\mathbf{x}_t}[\log p(\mathbf{x}_t|\mathbf{c}) + \omega(f(\mathbf{x}_t) \cdot g(\mathbf{c}))]$$




Radford et al., "Learning Transferable Visual Models From Natural Language Supervision", 2021.
Nichol et al., "GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models", 2021.

# GLIDE

## OpenAI

- Fine-tune the model especially for inpainting: feed randomly occluded images with an additional mask channel as the input.

"an old car in a snowy forest"

"a man wearing a white hat"

Text-conditional image inpainting examples

[Nichol et al., "GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models", 2021.]

# DALL·E 2

OpenAI

a shiba inu wearing a beret and black turtleneck

a close up of a handpalm with leaves growing from it

1kx1k Text-to-image generation.

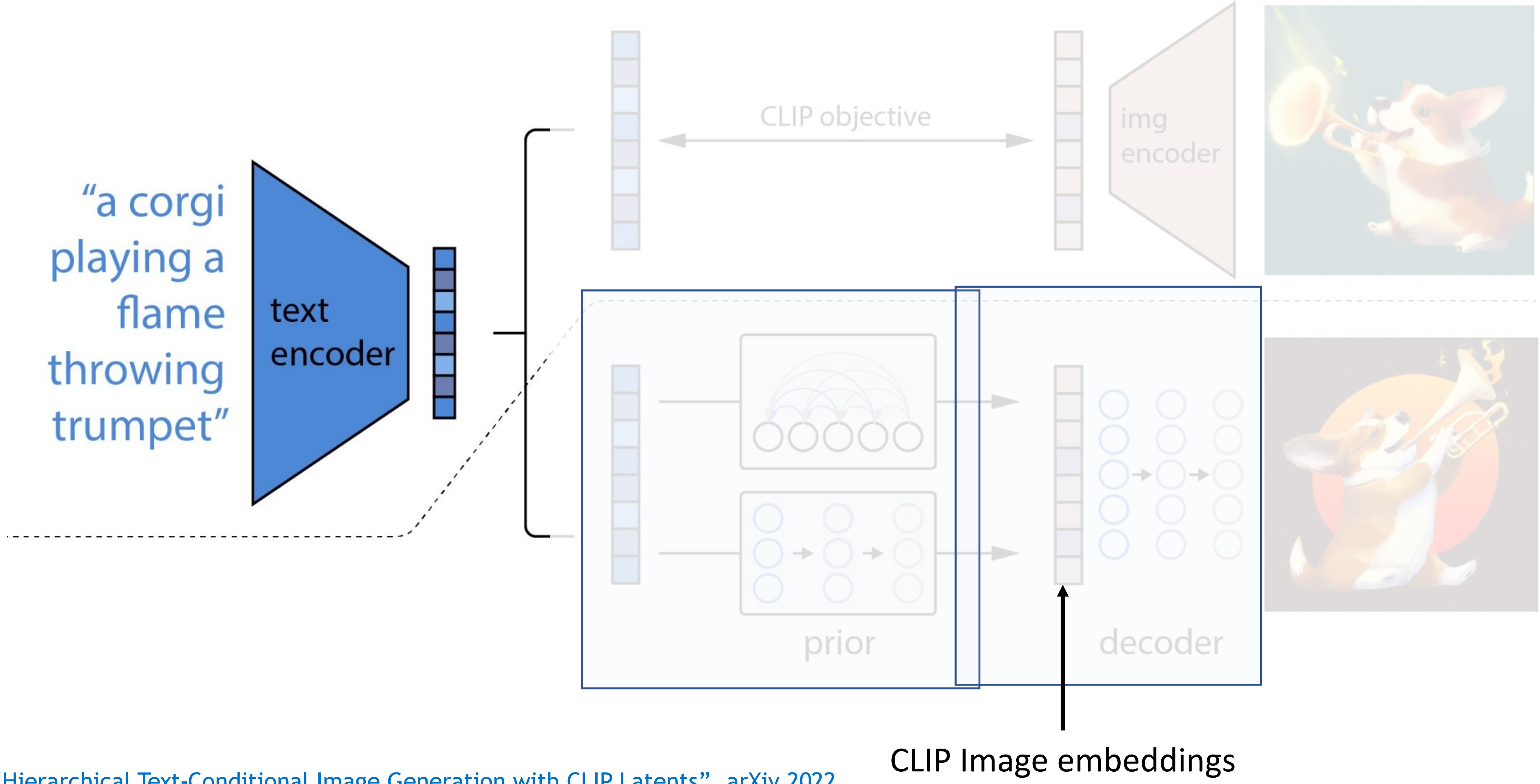
Outperform DALL-E (autoregressive transformer).

[Ramesh et al., "Hierarchical Text-Conditional Image Generation with CLIP Latents", arXiv 2022.]

# DALL·E 2

## Model components

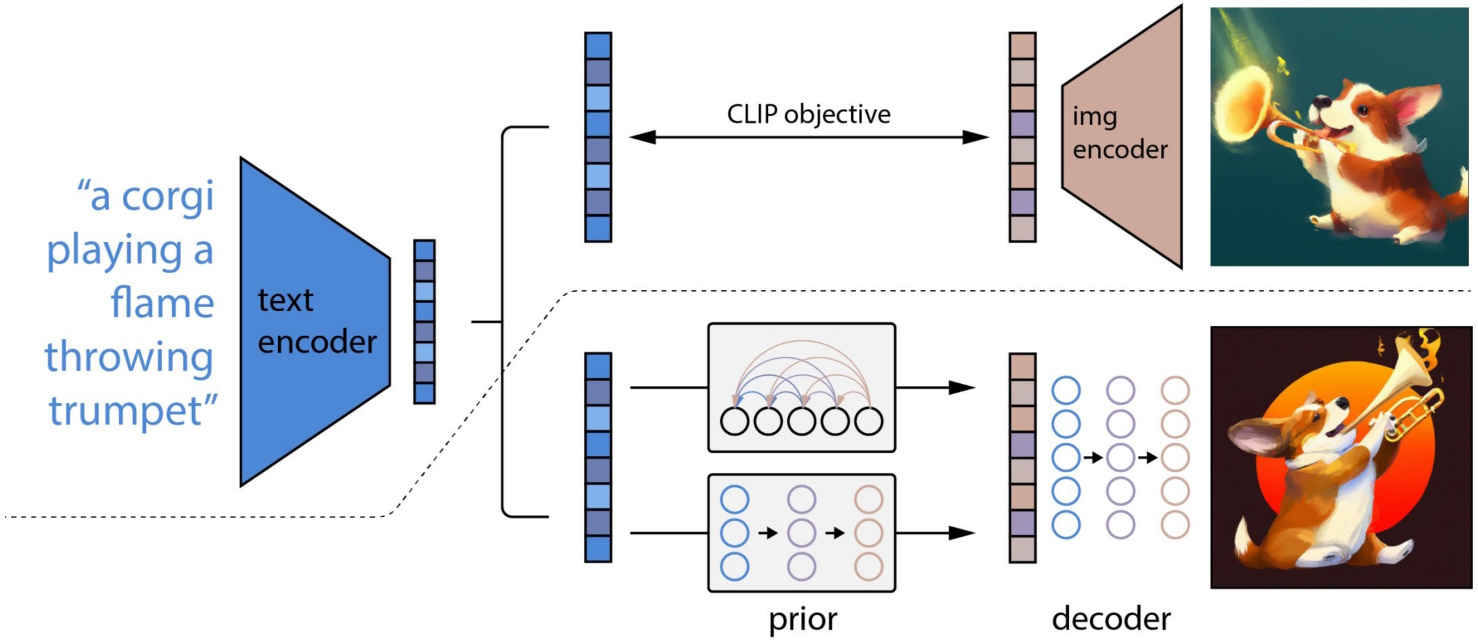
Prior: produces CLIP image embeddings conditioned on the caption.
Decoder: produces images conditioned on CLIP image embeddings and text.




Ramesh et al., "Hierarchical Text-Conditional Image Generation with CLIP Latents", arXiv 2022.

# DALL·E 2

## Model components




Why conditional on CLIP image embeddings?

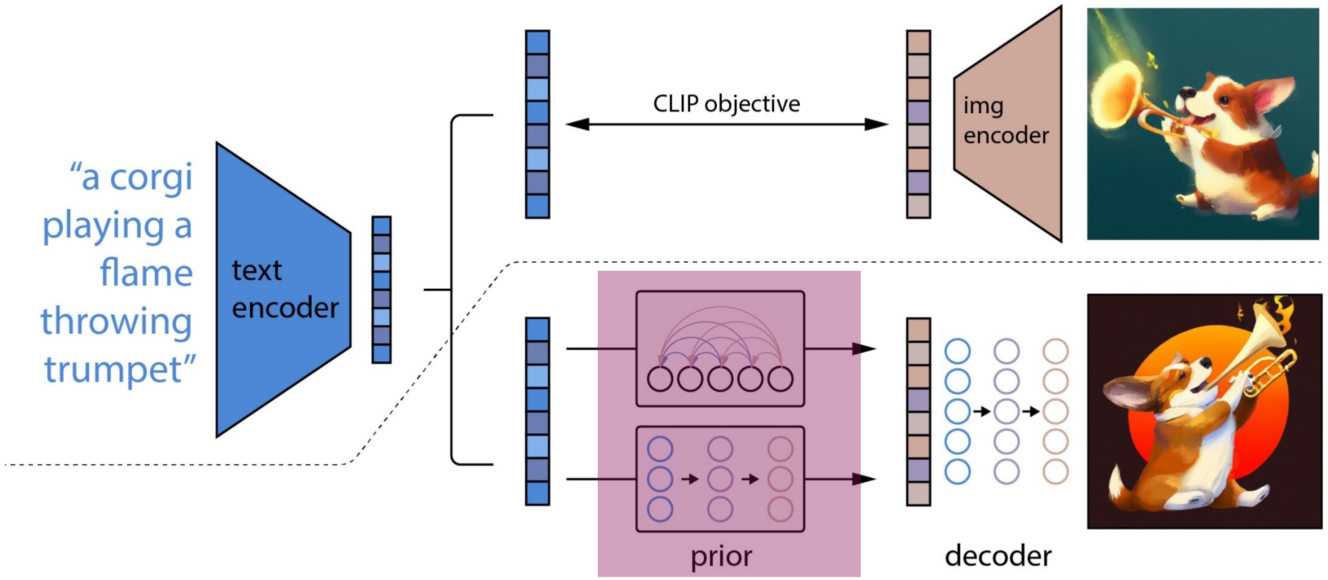
CLIP image embeddings capture high-level semantic meaning.

Latents in the decoder model take care of the rest.

The bipartite latent representation enables several text-guided image manipulation tasks.

# DALL·E 2

## Model components (1/2): prior model



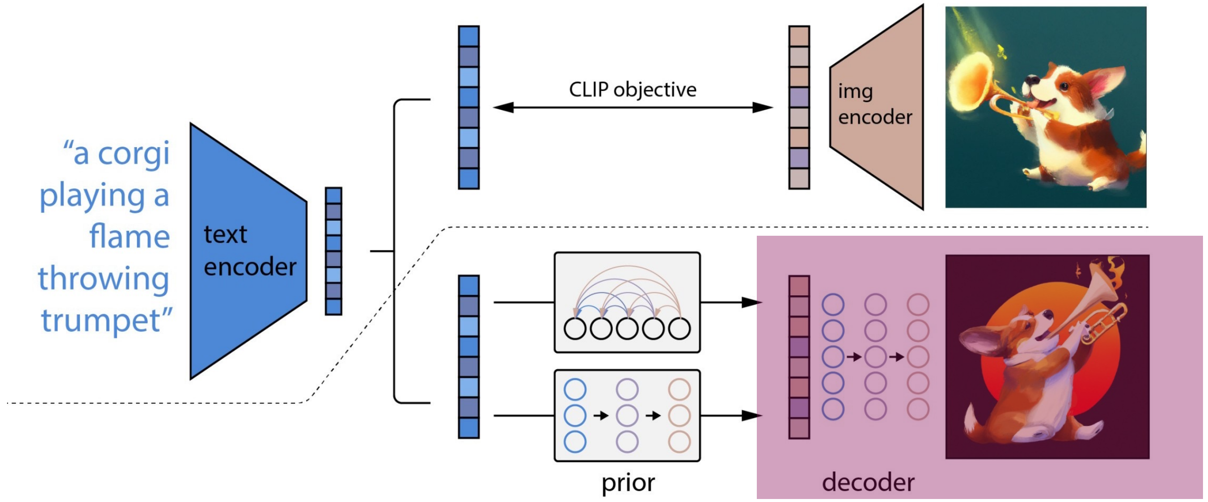

Prior: produces CLIP image embeddings conditioned on the caption.

- Option 1. autoregressive prior: quantize image embedding to a seq. of discrete codes and predict them autoregressively.
- Option 2. diffusion prior: model the continuous image embedding by diffusion models conditioned on caption.


Ramesh et al., "Hierarchical Text-Conditional Image Generation with CLIP Latents", arXiv 2022.

# DALL·E 2

## Model components (2/2): decoder model




Decoder: produces images conditioned on CLIP image embeddings (and text).

- Cascaded diffusion models: 1 base model (64x64), 2 super-resolution models (64x64 → 256x256, 256x256 → 1024x1024).
- Largest super-resolution model is trained on patches and takes full-res inputs at inference time.
- Classifier-free guidance & noise conditioning augmentation are important.


Ramesh et al., "Hierarchical Text-Conditional Image Generation with CLIP Latents", arXiv 2022.

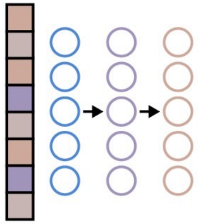
# DALL·E 2

## Bipartite latent representations

Bipartite latent representations $(\mathbf{z}, \mathbf{x}_T)$

$\mathbf{z}$: CLIP image embeddings

$\mathbf{x}_T$: inversion of DDIM sampler
(latents in the decoder model)



decoder

Near exact
reconstruction

Ramesh et al., "Hierarchical Text-Conditional Image Generation with CLIP Latents", arXiv 2022.

# DALL·E 2

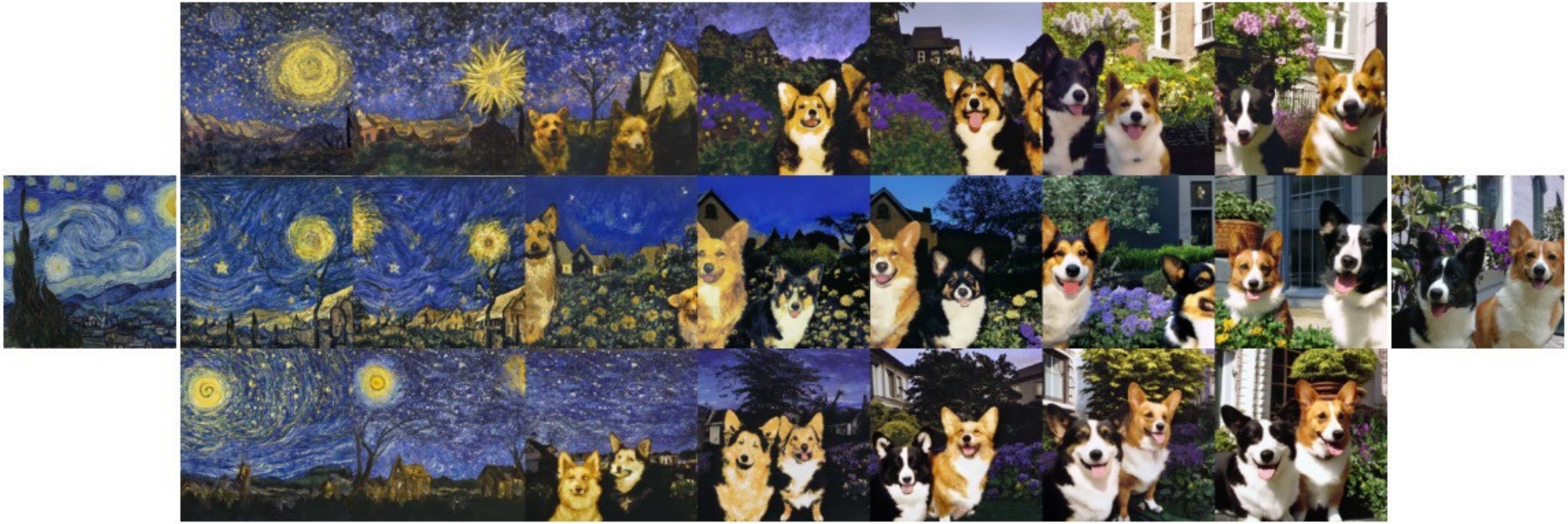
## Image variations

Fix the CLIP embedding $\mathbf{z}$.

Decode using different decoder latents $\mathbf{x}_T$

# DALL·E 2

## Image interpolation



Interpolate image CLIP embeddings $\mathbf{z}$.

Use different $\mathbf{x}_T$ to get different interpolation trajectories.

Ramesh et al., "Hierarchical Text-Conditional Image Generation with CLIP Latents", arXiv 2022.

# DALL·E 2

## Text Diffs

a photo of a cat → an anime drawing of a super saiyan cat, artstation

a photo of a victorian house → a photo of a modern house

a photo of an adult lion → a photo of lion cub

Change the image CLIP embedding towards the difference of the text CLIP embeddings of two prompts.

Decoder latent is kept as a constant.

# Imagen

Google Research, Brain team

Input: text;    Output: 1kx1k images

- An unprecedented degree of photorealism
    - SOTA automatic scores & human ratings
- A deep level of language understanding
- Extremely simple
    - no latent space, no quantization


A brain riding a rocketship heading towards the moon.

[Saharia et al., "Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding", arXiv 2022.]

# Imagen

Google Research, Brain team


A photo of a Shiba Inu dog with a backpack riding a bike. It is wearing sunglasses and a beach hat.

[Saharia et al., "Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding", arXiv 2022.]

# Imagen

Google Research, Brain team


A dragon fruit wearing karate belt in the snow.

[Saharia et al., "Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding", arXiv 2022.]

# Imagen

Google Research, Brain team


A relaxed garlic with a blindfold reading a newspaper while floating in a pool of tomato soup.

Saharia et al., "Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding", arXiv 2022.

# Imagen

Google Research, Brain team

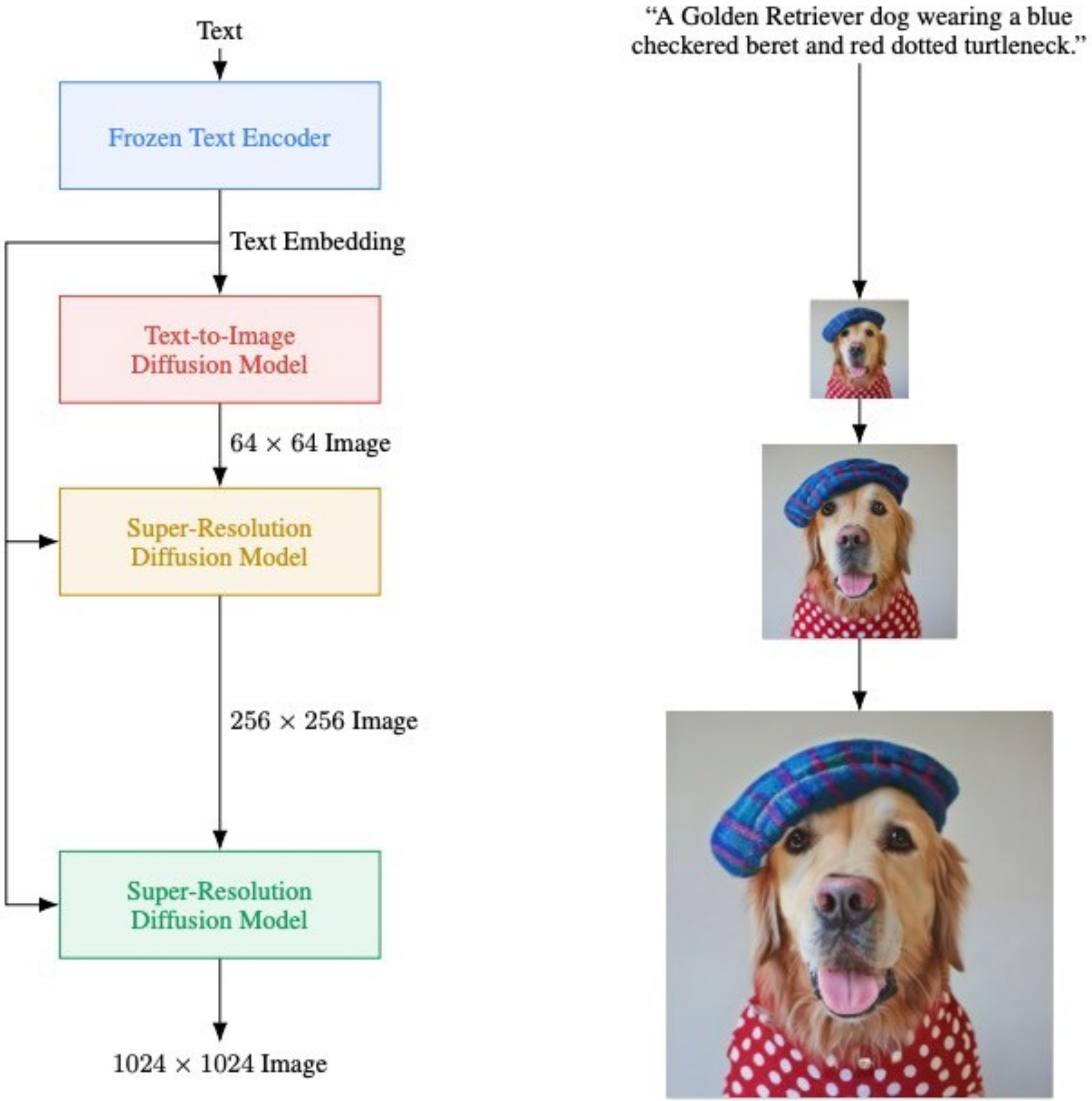

A cute hand-knitted koala wearing a sweater with 'CVPR' written on it.

Saharia et al., “Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding”, arXiv 2022.

# Imagen

Key modeling components:

- Cascaded diffusion models
- Classifier-free guidance and dynamic thresholding.
- Frozen large pretrained language models as text encoders. (T5-XXL)

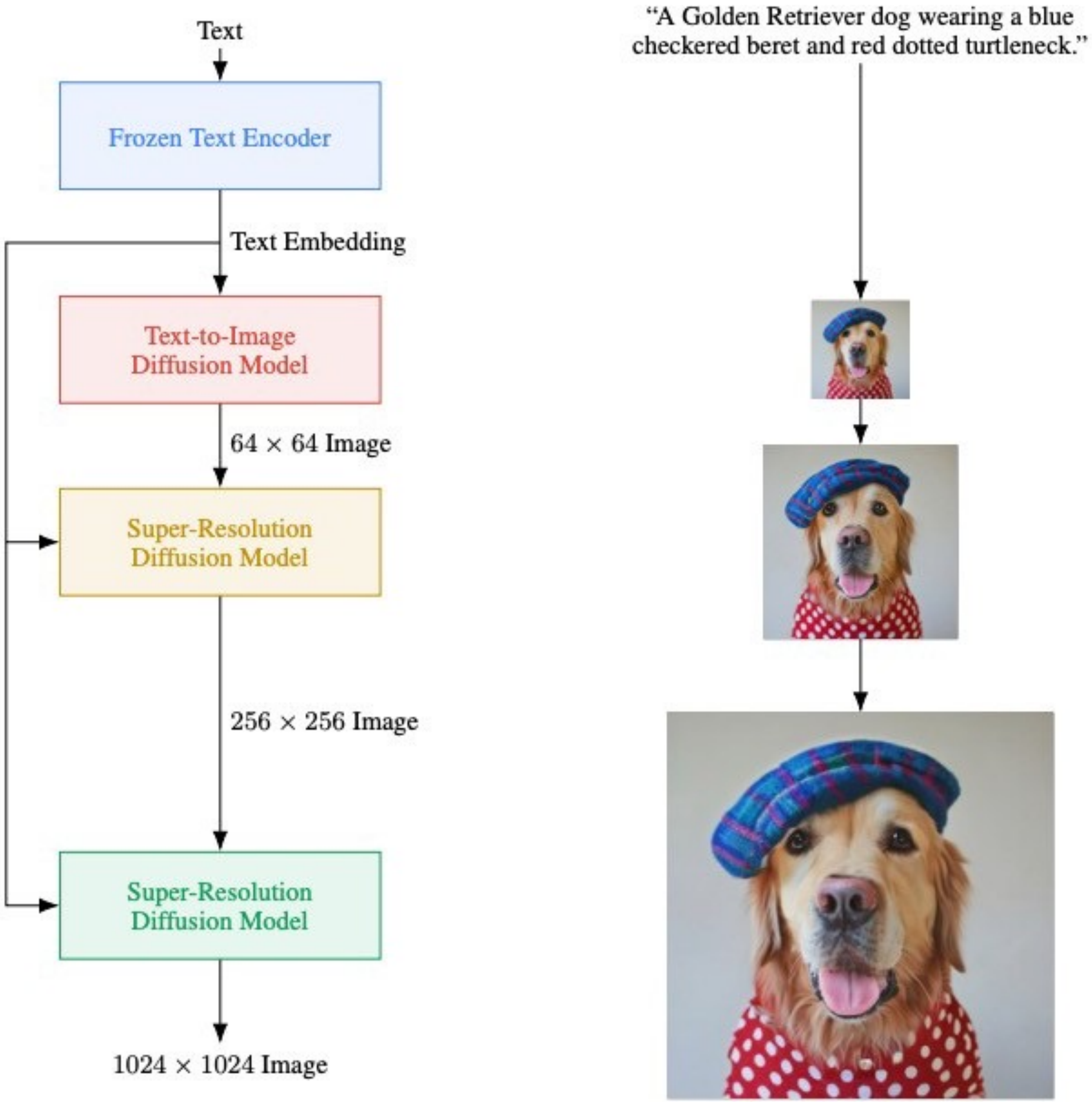



[Saharia et al., "Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding", arXiv 2022.]

# Imagen

Key observations:

- Beneficial to use text conditioning for all super-res models.
    - Noise conditioning augmentation weakens information from low-res models, thus needs text conditioning as extra information input.
- Scaling text encoder is extremely efficient.
    - More important than scaling diffusion model size.
- Human raters prefer T5-XXL as the text encoder over CLIP encoder on DrawBench.

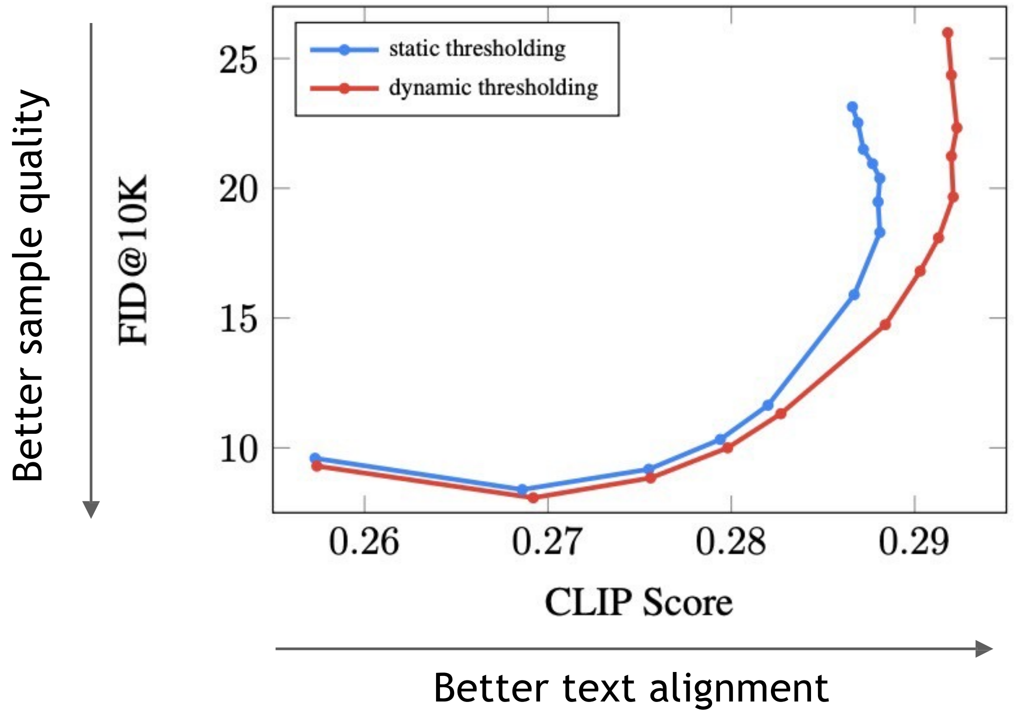



Saharia et al., "Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding", arXiv 2022.

# Imagen

## Dynamic thresholding

- Large classifier-free guidance weights → better text alignment, worse image quality




Saharia et al., "Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding", arXiv 2022.

# Imagen

## Dynamic thresholding

- Large classifier-free guidance weights → better text alignment, worse image quality
- Hypothesis : at large guidance weight, the generated images are saturated due to the very large gradient updates during sampling
- Solution – dynamic thresholding: adjusts the pixel values of samples at each sampling step to be within a dynamic range computed over the statistics of the current samples.

Saharia et al., "Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding", arXiv 2022.

# Imagen

## Dynamic thresholding

- Large clas
- Hypothesi sampling dates during
- Solution – range com ynamic

Static thresholding

Dynamic thresholding

[Saharia et al., "Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding", arXiv 2022.]

# Imagen

## DrawBench: new benchmark for text-to-image evaluations

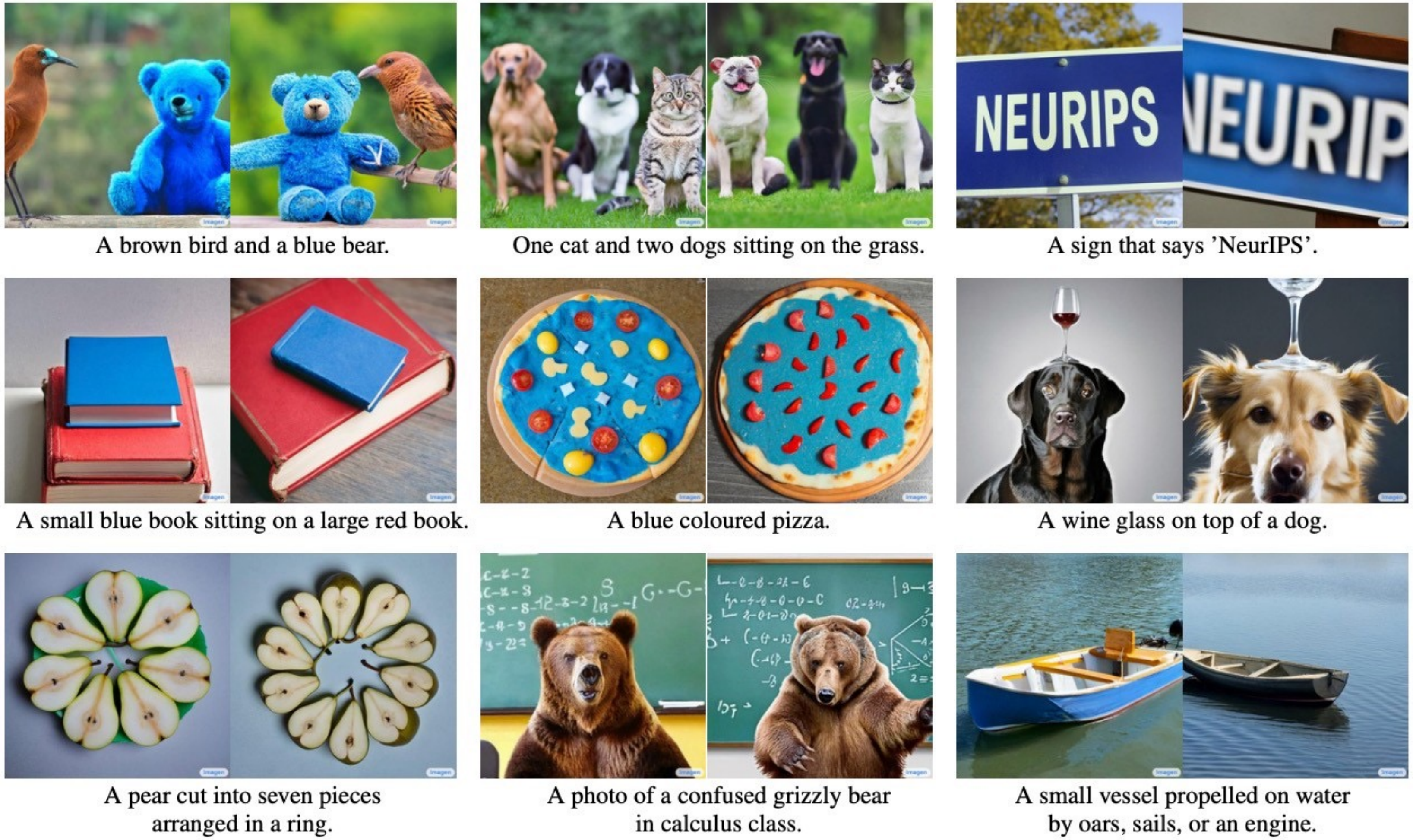
- A set of 200 prompts to evaluate text-to-image models across multiple dimensions.
    - E.g., the ability to faithfully render different colors, numbers of objects, spatial relations, text in the scene, unusual interactions between objects.
    - Contains complex prompts, e.g, long and intricate descriptions, rare words, misspelled prompts.

Saharia et al., "Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding", arXiv 2022.

# Imagen

## DrawBench: new benchmark for text-to-image evaluations

- A set of 200 pro
  - E.g., the cene, unusual interactic
  - Contains




A brown bird and a blue bear.
One cat and two dogs sitting on the grass.
A sign that says 'NeurIPS'.
A small blue book sitting on a large red book.
A blue coloured pizza.
A wine glass on top of a dog.
A pear cut into seven pieces arranged in a ring.
A photo of a confused grizzly bear in calculus class.
A small vessel propelled on water by oars, sails, or an engine.

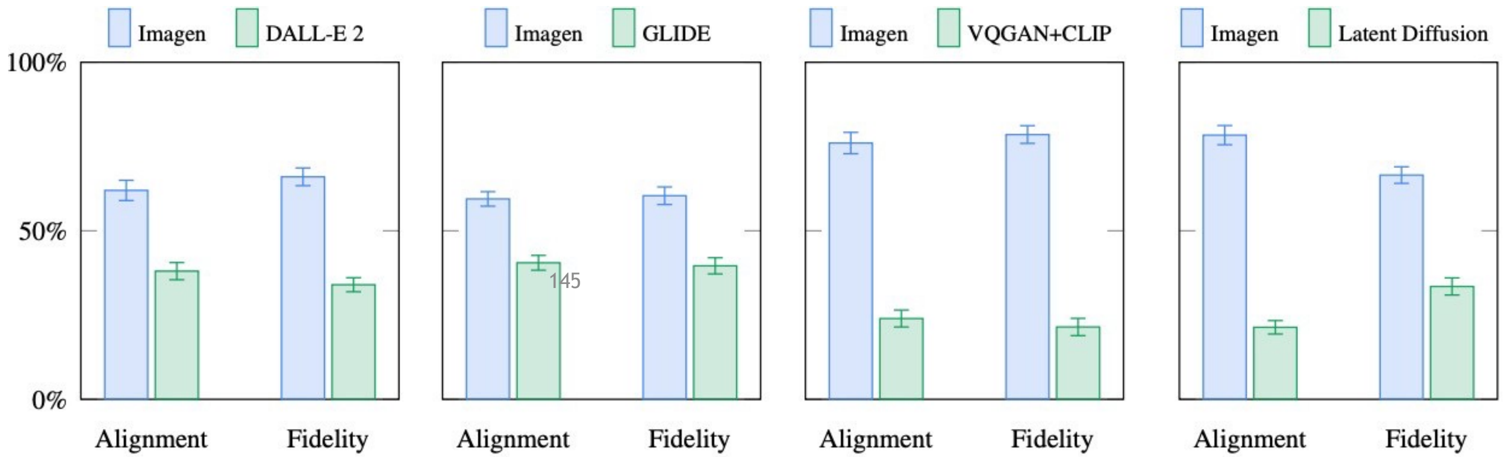
Saharia et al., "Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding", arXiv 2022.

# Imagen
## Evaluations

Imagen got SOTA automatic evaluation scores on COCO dataset

| Model | FID-30K | Zero-shot FID-30K |
|---|---|---|
| AttnGAN [76] | 35.49 | |
| DM-GAN [83] | 32.64 | |
| DF-GAN [69] | 21.42 | |
| DM-GAN + CL [78] | 20.79 | |
| XMC-GAN [81] | 9.33 | |
| LAFITE [82] | 8.12 | |
| Make-A-Scene [22] | 7.55 | |
| DALL-E [53] | | 17.89 |
| LAFITE [82] | | 26.94 |
| GLIDE [41] | | 12.24 |
| DALL-E 2 [54] | | 10.39 |
| **Imagen (Our Work)** | | **7.27** |

Imagen is preferred over recent work by human raters in sample quality & image-text alignment on DrawBench.






Saharia et al., "Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding", arXiv 2022.

# Stable Diffusion

## Latest & Publicly available text-to-image generation

To be discussed in detail in paper presentation

**High-Resolution Image Synthesis with Latent Diffusion Models**

Robin Rombach*, Andreas Blattmann*, Dominik Lorenz, Patrick Esser, Björn Ommer

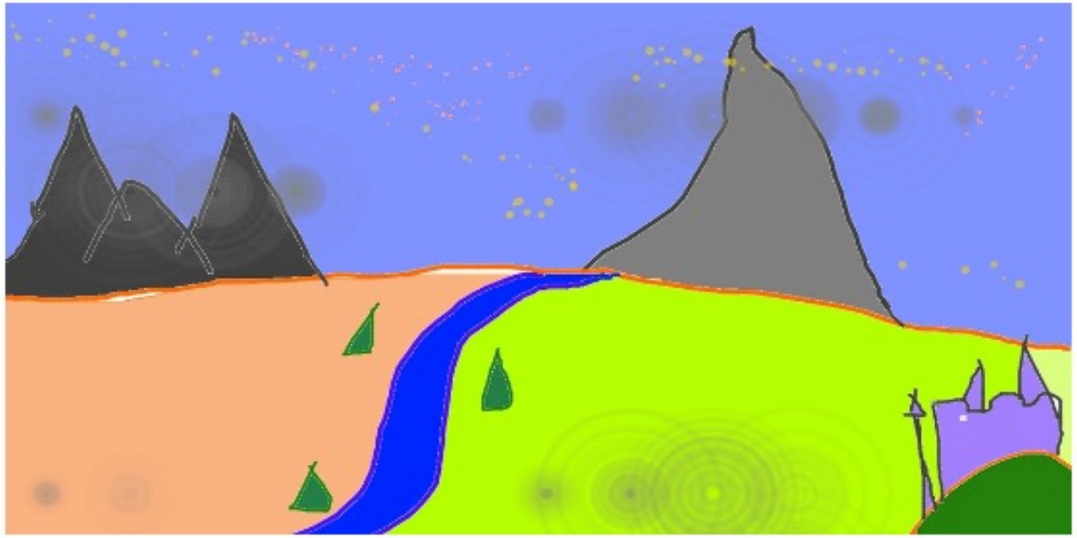
*CVPR '22 Oral | GitHub | arXiv | Project page*

Stable Diffusion is a latent text-to-image diffusion model. Thanks to a generous compute donation from Stability AI and support from LAION, we were able to train a Latent Diffusion Model on 512x512 images from a subset of the LAION-5B database. Similar to Google's Imagen, this model uses a frozen CLIP ViT-L/14 text encoder to condition the model on text prompts. With its 860M UNet and 123M text encoder, the model is relatively lightweight and runs on a GPU with at least 10GB VRAM. See this section below and the model card.

# Stable Diffusion

## Latest & Publicly available text-to-image generation

Input



Outputs

HW assignment: Use stable diffusion API to generate ‘interesting’ image from text prompt. All submissions will be rated for top 3!

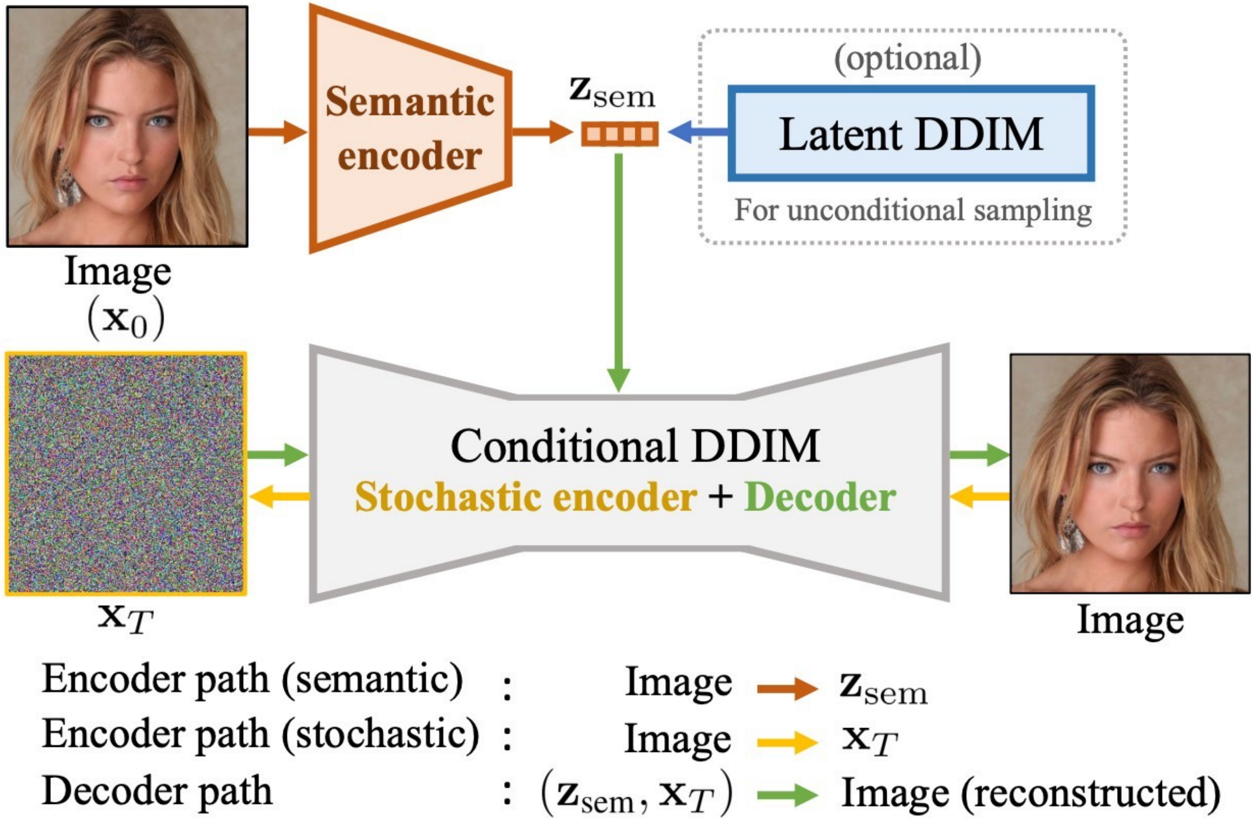
# Outline for today's class

- Theory:
  - How do we condition diffusion model?
  - How do people sample in practice? – DDIM sampling
  - How do we generate high-resolution images?

- Application:
  - Text to Image Generation
  - Image to Image Translation
  - Video Generation

# Diffusion Autoencoders

Learning semantic meaningful latent representations in diffusion models

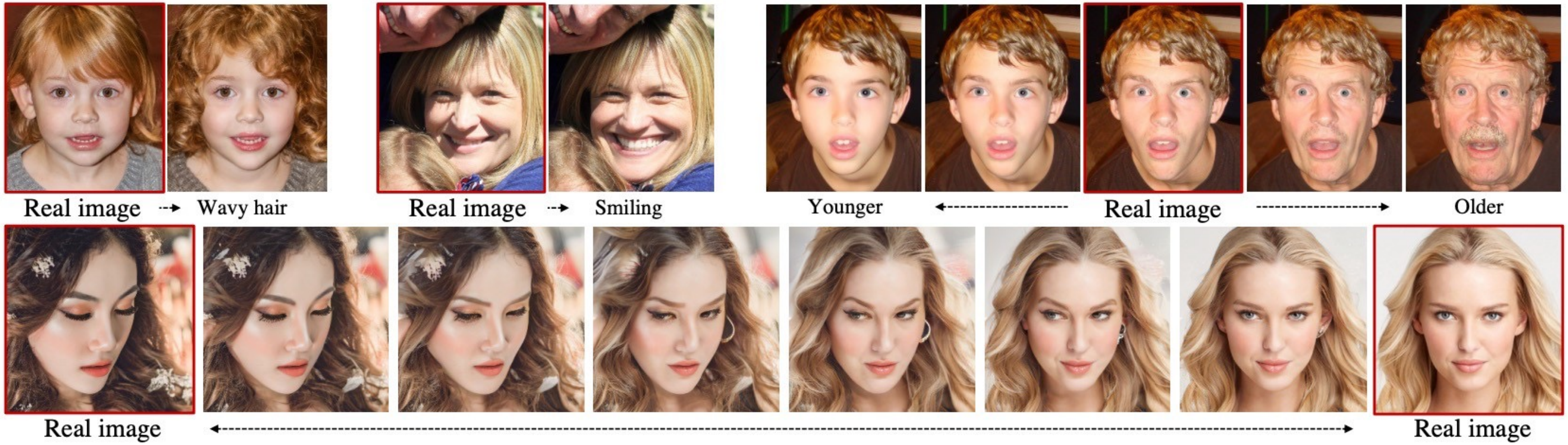



To be discussed in detail in paper presentation

Preechakul et al., "Diffusion Autoencoders: Toward a Meaningful and Decodable Representation", CVPR 2022.

# Diffusion Autoencoders

Learning semantic meaningful latent representations in diffusion models



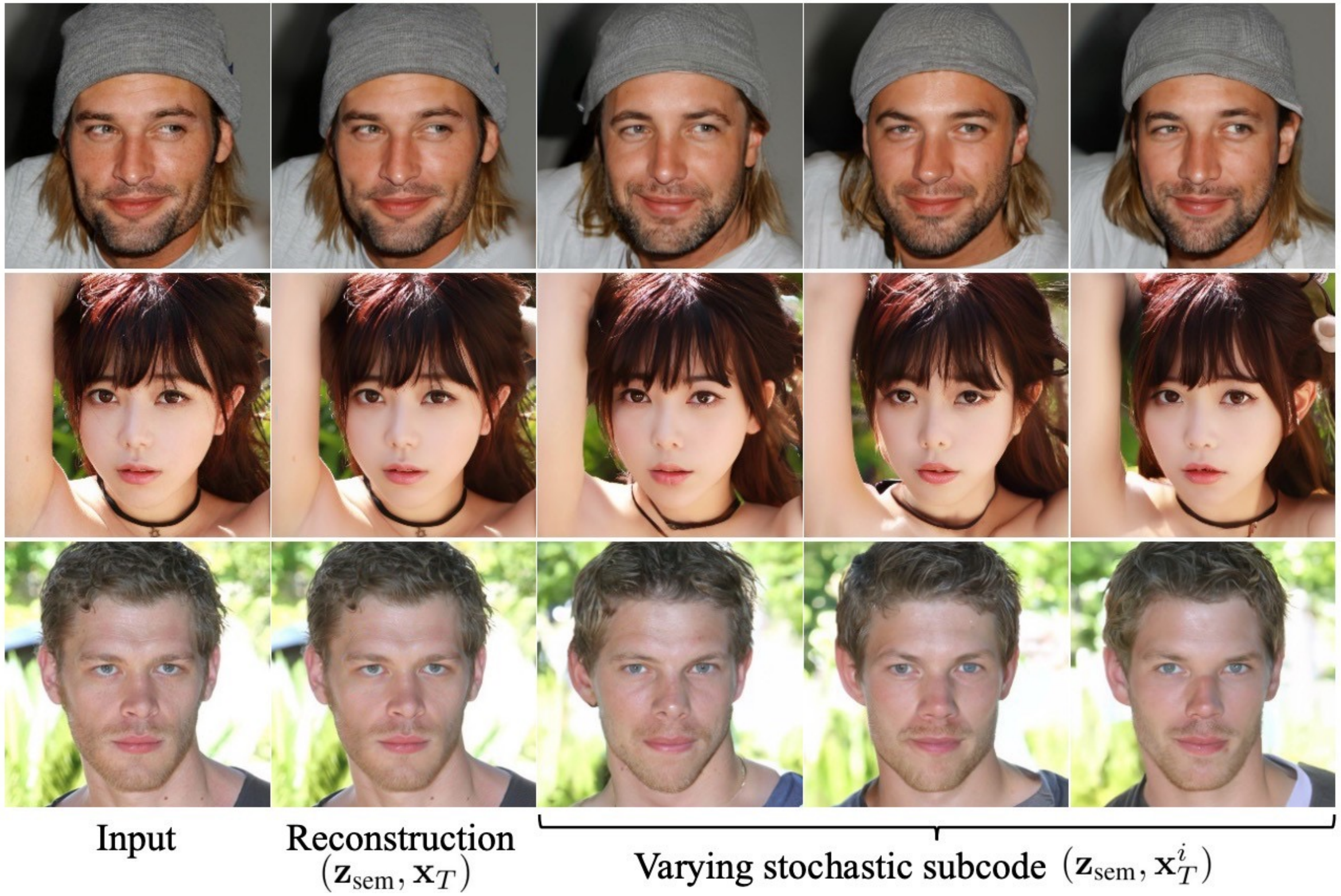

Changing the semantic latent $\mathbf{z}_{\text{sem}}$

Very similar to StyleGAN based editing. $Z_{sem}$ is the latent representation similar to the W/W+ space of StyleGAN

Preechakul et al., "Diffusion Autoencoders: Toward a Meaningful and Decodable Representation", CVPR 2022.

# Diffusion Autoencoders

Learning semantic meaningful latent representations in diffusion models




Preechakul et al., "Diffusion Autoencoders: Toward a Meaningful and Decodable Representation", CVPR 2022.

# Super-Resolution

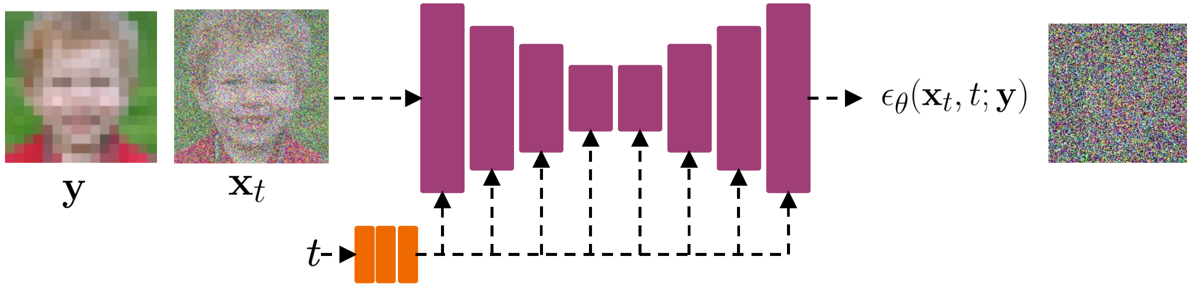
## Super-Resolution via Repeated Refinement (SR3)

Image super-resolution can be considered as training $p(\mathbf{x}|\mathbf{y})$ where $\mathbf{y}$ is a low-resolution image and $\mathbf{x}$ is the corresponding high-resolution image

Train a score model for $\mathbf{x}$ conditioned on $\mathbf{y}$ using:

$$\mathbb{E}_{\mathbf{x},\mathbf{y}}\, \mathbb{E}_{\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\, \mathbb{E}_t\, ||\epsilon_\theta(\mathbf{x}_t, t; \mathbf{y}) - \epsilon||_p^p$$

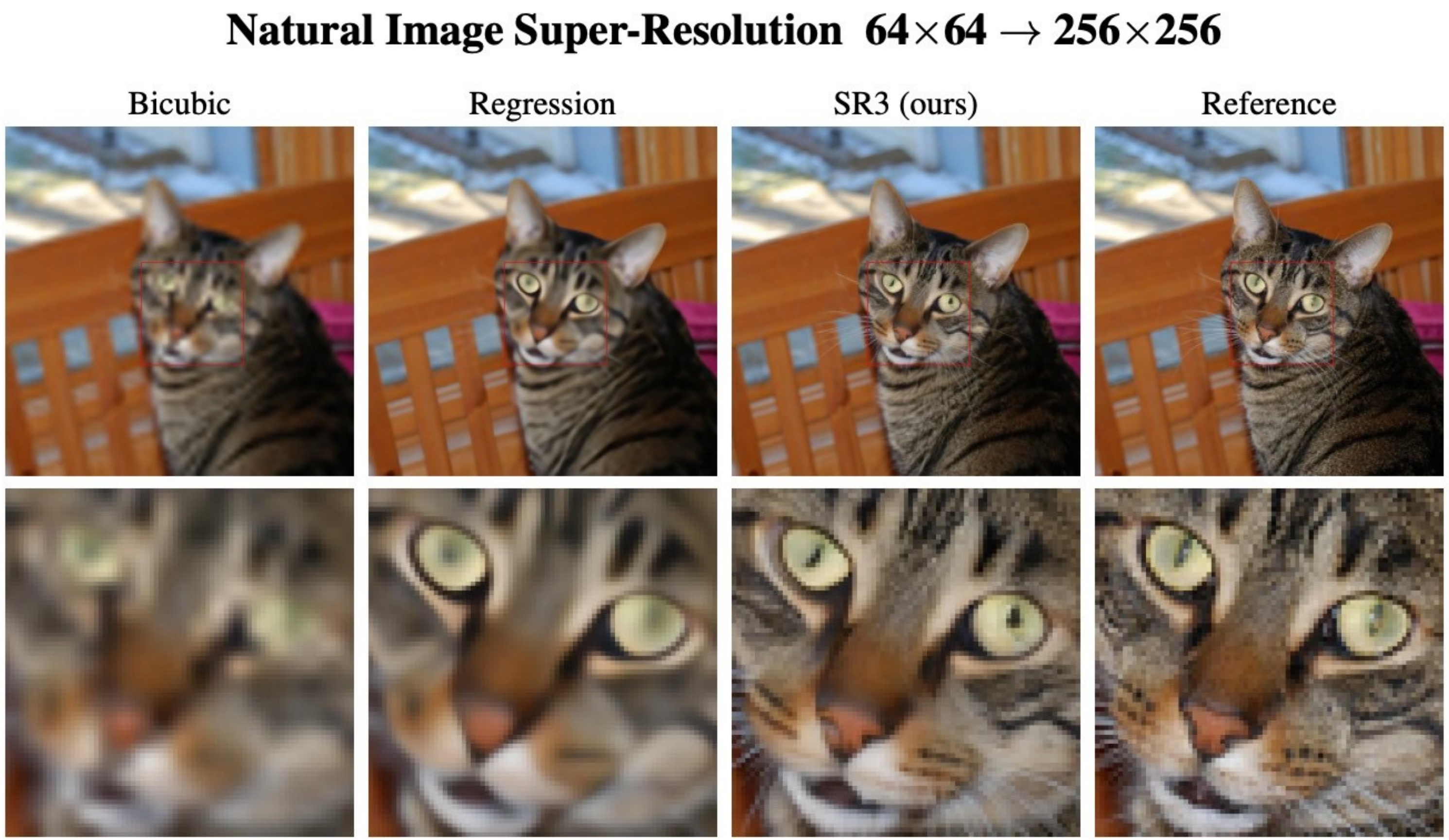
The conditional score is simply a U-Net with $\mathbf{x_t}$ and $\mathbf{y}$ (resolution image) concatenated.




Saharia et al., Image Super-Resolution via Iterative Refinement, 2021

# Super-Resolution

Super-Resolution via Repeated Refinement (SR3)




[Saharia et al., Image Super-Resolution via Iterative Refinement, 2021](#)

# Image-to-Image Translation

## Palette: Image-to-Image Diffusion Models

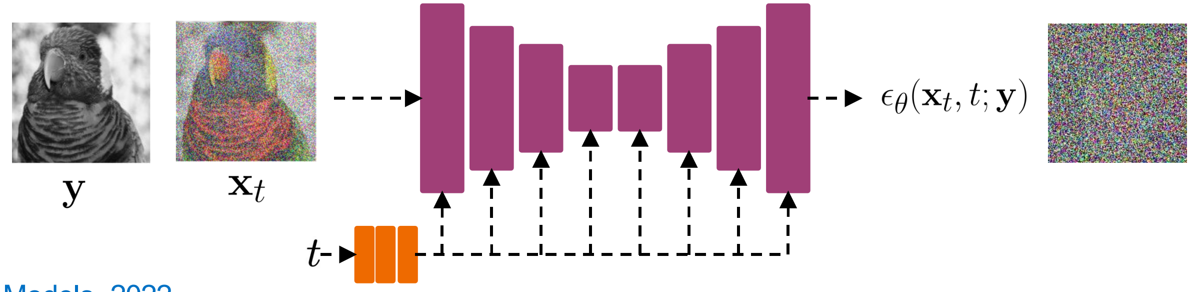
Many image-to-image translation applications can be considered as training $p(\mathbf{x}|\mathbf{y})$ where $\mathbf{y}$ is the input image.

For example, for colorization, $\mathbf{x}$ is a colored image and $\mathbf{y}$ is a gray-level image.

Train a score model for $\mathbf{x}$ conditioned on $\mathbf{y}$ using:

$$\mathbb{E}_{\mathbf{x},\mathbf{y}}\, \mathbb{E}_{\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\, \mathbb{E}_t\, ||\epsilon_\theta(\mathbf{x}_t, t; \mathbf{y}) - \epsilon||_p^p$$

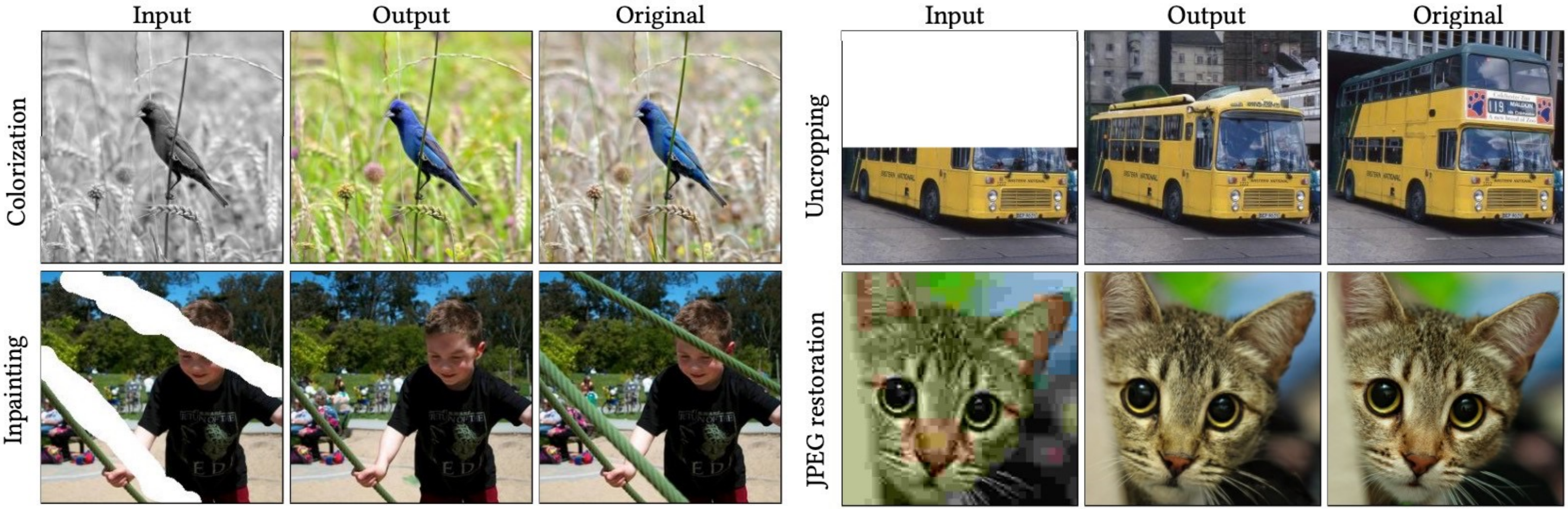
The conditional score is simply a U-Net with $\mathbf{x}_t$ and $\mathbf{y}$ concatenated.




Saharia et al., Palette: Image-to-Image Diffusion Models, 2022

# Image-to-Image Translation

Palette: Image-to-Image Diffusion Models




Saharia et al., Palette: Image-to-Image Diffusion Models, 2022

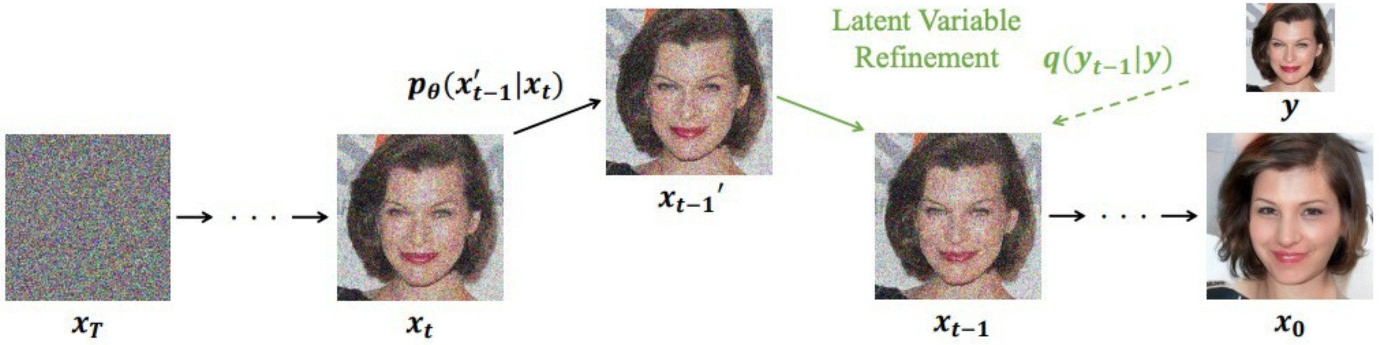
# Conditional Generation

## Iterative Latent Variable Refinement (ILVR)

To be discussed in detail in paper presentation

A simple technique to guide the generation process of an unconditional diffusion model using a reference image.

Given the conditioning (reference) image $\mathbf{y}$ the generation process is modified to pull the samples towards the reference image.




**for** $t = T, ..., 1$ **do**

$\quad \mathbf{z} \sim N(\mathbf{0}, \mathbf{I})$

$\quad x'_{t-1} \sim p_\theta(x'_{t-1}|x_t)$ ▷ unconditional proposal

$\quad y_{t-1} \sim q(y_{t-1}|y)$ ▷ condition encoding

$\quad x_{t-1} \leftarrow \phi_N(y_{t-1}) + x'_{t-1} - \phi_N(x'_{t-1})$

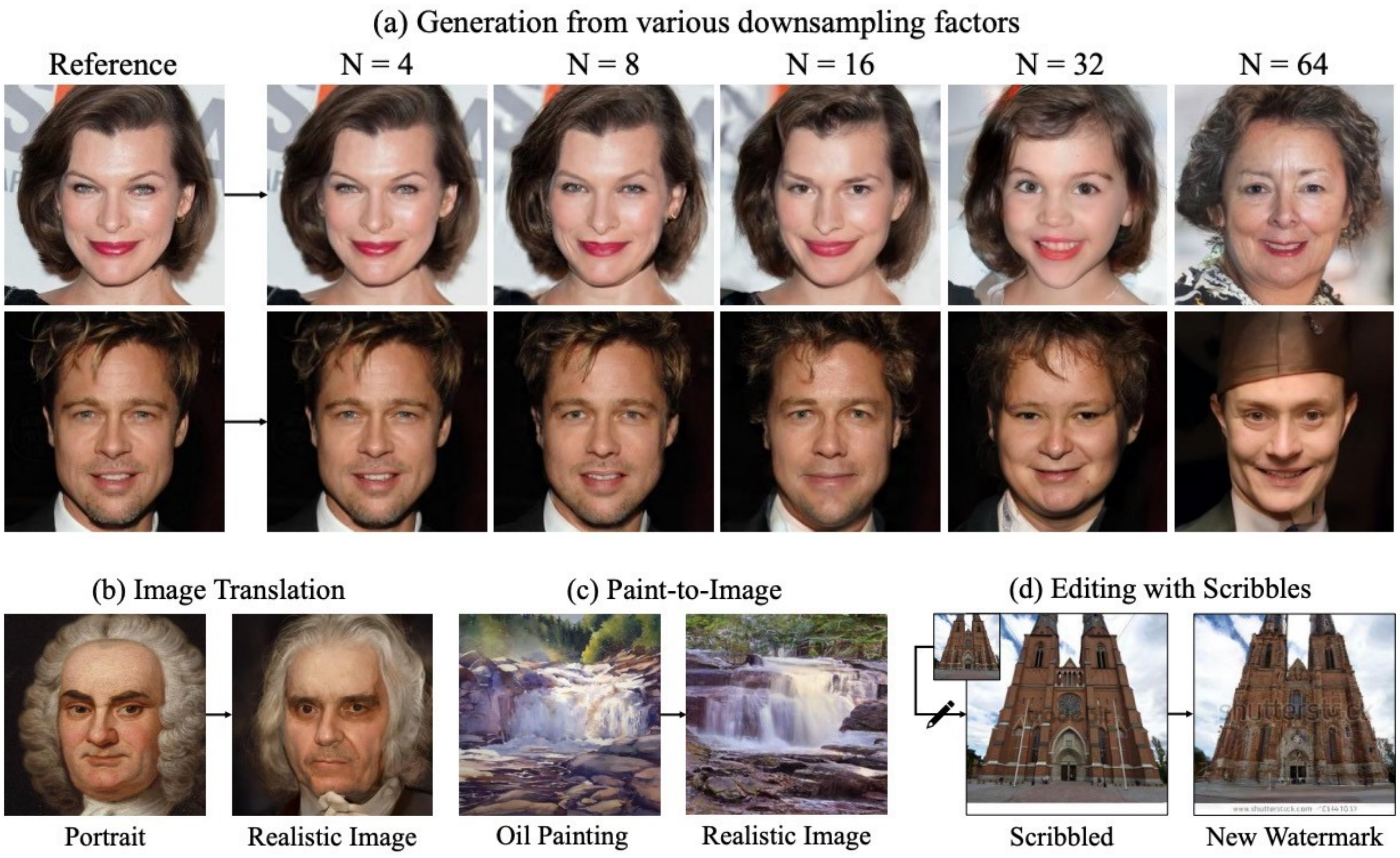
**end for**

Low-pass filter
Downsampling / Upsampling by a factor of N

Choi et al., ILVR: Conditioning Method for Denoising Diffusion Probabilistic Models, ICCV 2021

# Conditional Generation

## Iterative Latent Variable Refinement (ILVR)

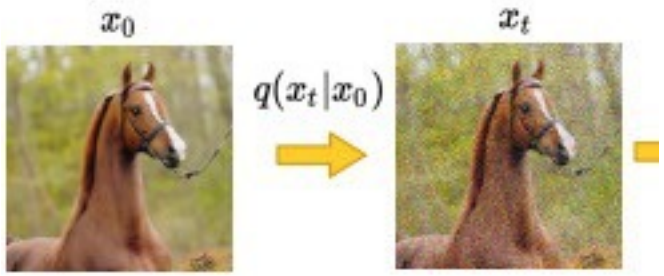



[Choi et al., ILVR: Conditioning Method for Denoising Diffusion Probabilistic Models, ICCV 2021]

# Semantic Segmentation

## Label-efficient semantic segmentation with diffusion models

Can we use representation learned from diffusion models for downstream applications such as semantic segmentation?

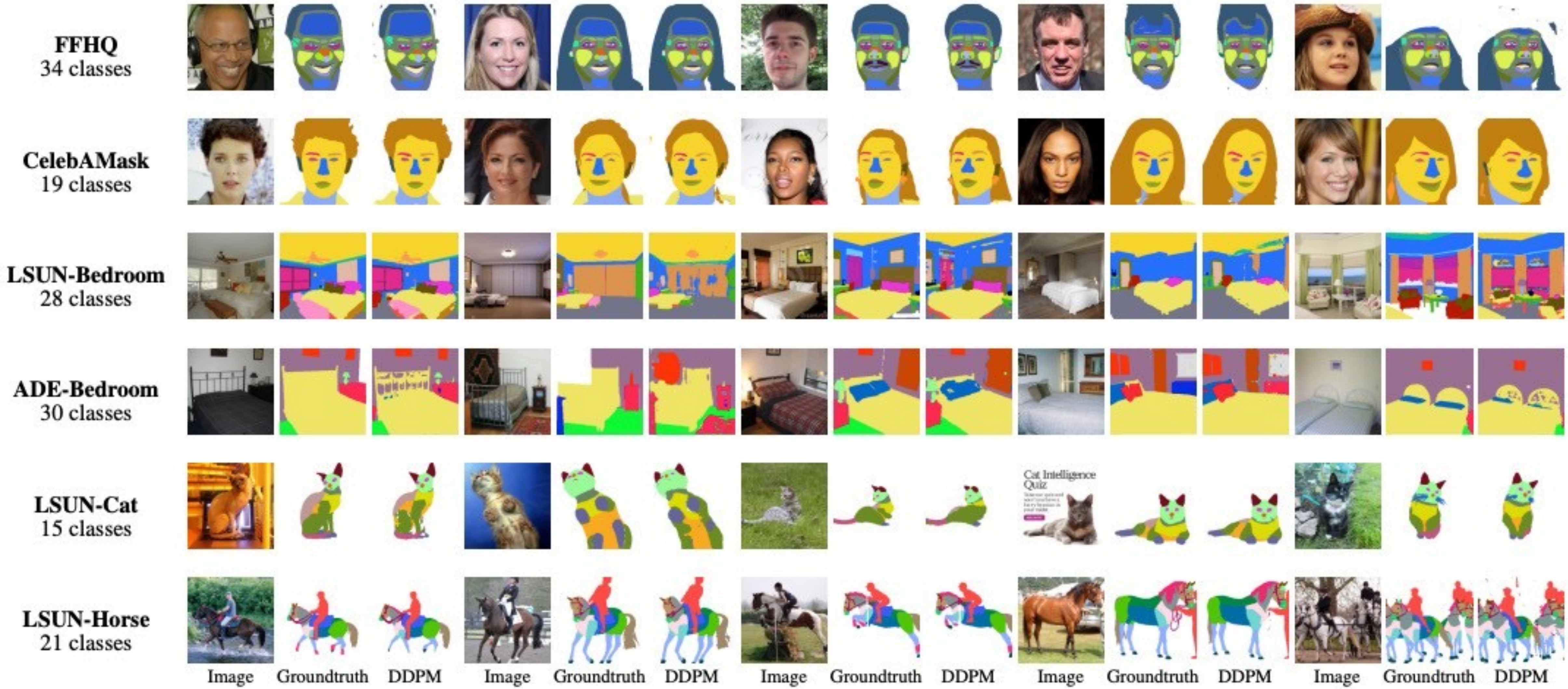



[Baranchuk et al., Label-Efficient Semantic Segmentation with Diffusion Models, ICLR 2022](#)

# Semantic Segmentation

## Label-efficient semantic segmentation with diffusion models

The experimental results show that the proposed method outperforms Masked Autoencoders, GAN and VAE-based models.




[Baranchuk et al., Label-Efficient Semantic Segmentation with Diffusion Models, ICLR 2022]()

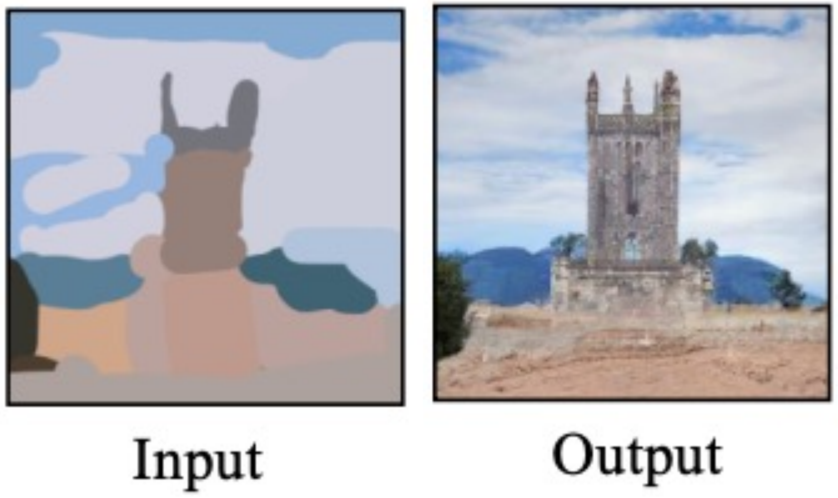
# Image Editing

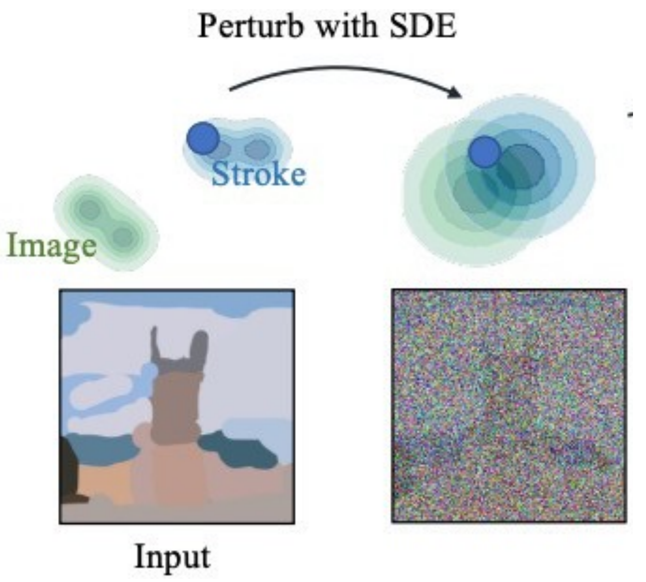
## SDEdit

Goal: Given a stroke painting with color, generate a photorealistic image

Key Idea:

- Latent Distribution of stroke and real images do not overlap.
- But once we apply forward diffusion on them, their distribution start overlapping as finally it becomes gaussian noise.

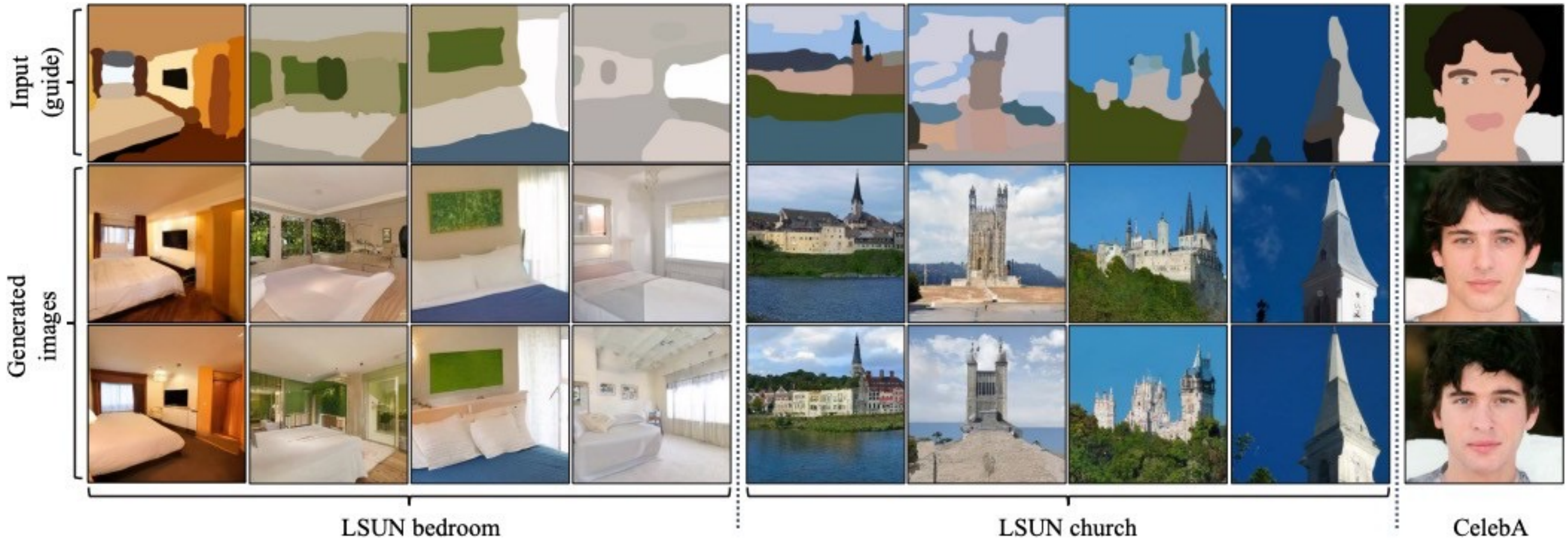



Meng et al., SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations, ICLR 2022

# Image Editing

SDEdit




[Meng et al., SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations, ICLR 2022](#)

# Outline for today's class

- Theory:
  - How do we condition diffusion model?
  - How do people sample in practice? – DDIM sampling
  - How do we generate high-resolution images?
- Application:
  - Text to Image Generation
  - Image to Image Translation
  - Video Generation

# Video Generation

Samples from a text-conditioned video diffusion model, conditioned on the string ***fireworks***.

[Ho et al., “Video Diffusion Models”, *arXiv*, 2022](#)
[Harvey et al., “Flexible Diffusion Modeling of Long Videos”, arXiv, 2022](#)

[Yang et al., “Diffusion Probabilistic Modeling for Video Generation”, *arXiv*, 2022](#)
[Höppe et al., “Diffusion Models for Video Prediction and Infilling”, arXiv, 2022](#)
[Voleti et al., “MCVD: Masked Conditional Video Diffusion for Prediction, Generation, and Interpolation”, *arXiv*, 2022](#)

# Video Generation

**Video Generation Tasks:**

- Unconditional Generation (Generate all frames)
- Future Prediction (Generate future from past fames)
- Past Prediction (Generate past from future fames)
- Interpolation (Generate intermediate frames)

➡ Learn a model of the form:

$$p_{\boldsymbol{\theta}}(\mathbf{x}^{t_1}, \cdots, \mathbf{x}^{t_K} | \mathbf{x}^{\tau_1}, \cdots, \mathbf{x}^{\tau_M})$$

Given frames: $\mathbf{x}^{\tau_1}, \cdots, \mathbf{x}^{\tau_M}$

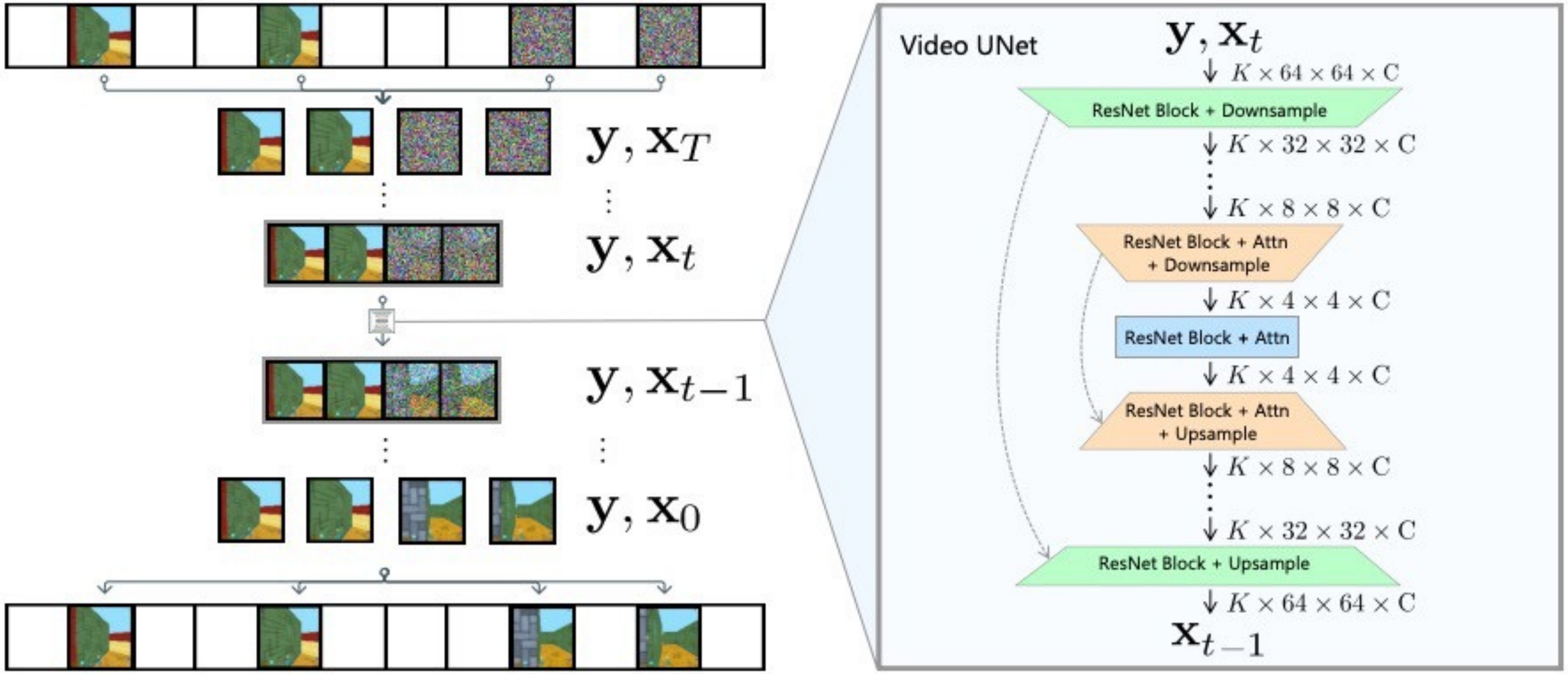
Frames to be predicted: $\mathbf{x}^{t_1}, \cdots, \mathbf{x}^{t_K}$


Ho et al., "Video Diffusion Models", *arXiv*, 2022
Harvey et al., "Flexible Diffusion Modeling of Long Videos", arXiv, 2022
Yang et al., "Diffusion Probabilistic Modeling for Video Generation", *arXiv*, 2022
Höppe et al., "Diffusion Models for Video Prediction and Infilling", arXiv, 2022
Voleti et al., "MCVD: Masked Conditional Video Diffusion for Prediction, Generation, and Interpolation", *arXiv*, 2022

# Video Generation

Learn one model for everything:

- Architecture as **one diffusion model** over **all frames concatenated**.
- Mask frames to be predicted; provide conditioning frames; vary applied masking/conditioning for different tasks during training.
- Use **time position encodings** to encode times.




(image from: Harvey et al., "Flexible Diffusion Modeling of Long Videos", *arXiv*, 2022)

# Video Generation

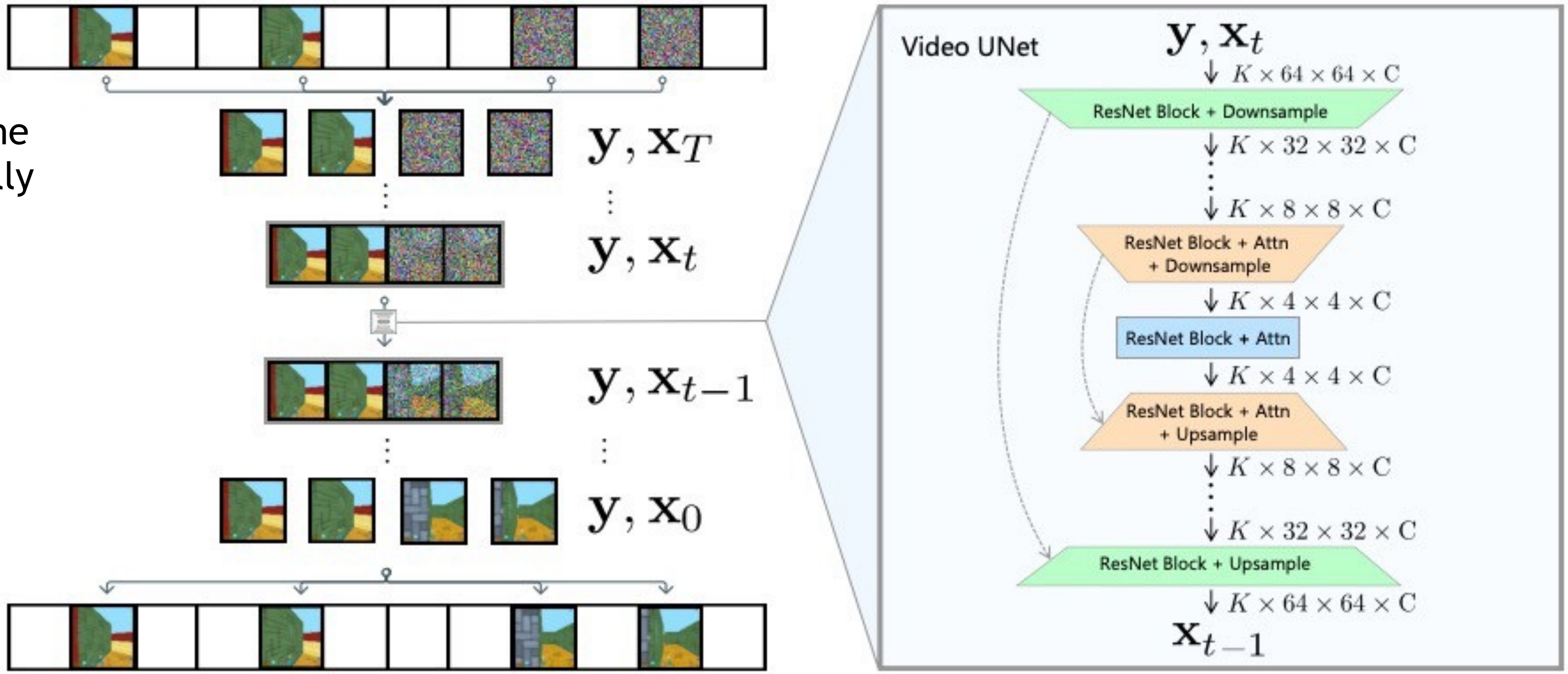
**Architecture Details:**

Data is 4D (image height, image width, #frames, channels)

- Option (1): 3D Convolutions. Can be computationally expensive.
- Option (2): Spatial 2D Convolutions + Attention Layers along frame axis.

➡ Additional Advantage:

Ignoring the attention layers, the model can be trained additionally on pure image data!




(image from: Harvey et al., “Flexible Diffusion Modeling of Long Videos”, *arXiv*, 2022)

# Video Generation

## Results

**Long term video generation in hierarchical manner:**

- 1. Generate future frames in sparse manner, conditioning on frames far back
- 2. Interpolate in-between frames

➡ 1+ hour coherent video generation possible!

Test Data:

Generated:

(video from: Harvey et al., "Flexible Diffusion Modeling of Long Videos", *arXiv*, 2022, https://plai.cs.ubc.ca/2022/05/20/flexible-diffusion-modeling-of-long-videos/)

# Make-A-Video

Make-A-Video research builds on the recent progress made in text-to-image generation technology built to enable text-to-video generation. The system uses images with descriptions to learn what the world looks like and how it is often described. It also uses unlabeled videos to learn how the world moves. With this data, Make-A-Video lets you bring your imagination to life by generating whimsical, one-of-a-kind videos with just a few words or lines of text.


A dog wearing a Superhero outfit with red cape flying through the sky

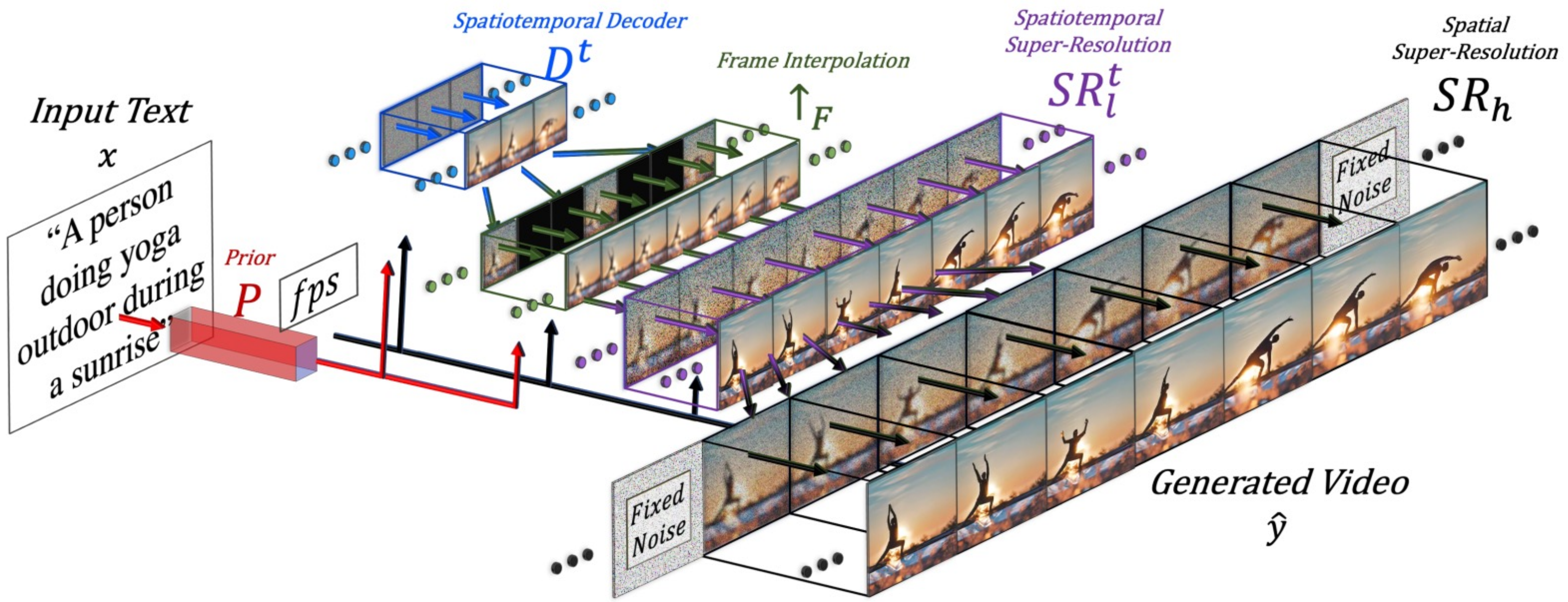



Figure 2: **Make-A-Video high-level architecture.** Given input text $x$ translated by the prior P into an image embedding, and a desired frame rate $fps$, the decoder $\mathrm{D^t}$ generates 16 $64 \times 64$ frames, which are then interpolated to a higher frame rate by $\uparrow_F$, and increased in resolution to $256 \times 256$ by $\mathrm{SR}_l^t$ and $768 \times 768$ by $\mathrm{SR}_h$, resulting in a high-spatiotemporal-resolution generated video $\hat{y}$.

# Open Problems

- Sampling from diffusion model is still slow (even with DDIM you need 250 steps)
    - How can one sample with even fewer steps?

- How good is the latent space of diffusion model for downstream tasks?
    - ResNet on ImageNet gives us great image features
    - LLM gives great text features
    - Can diffusion model beat imagenet feature?
    - Can diffusion model help us in discriminative tasks?

# Open Problems

- How do we better control diffusion model for conditional generation?
  - With GANs we can do very fine-grained condition and can even use 3D intrinsics.
  - While Diffusion model produces impressive image quality, it often ignores detailed conditioning like segmentation mask, and works best for vague conditioning like text or class.
  - How do we enable that?

- Can we generate more involved videos with diffusion models?
  - This is something that is limited with GAN
  - Current diffusion models are limited in the kinds of video in can generate, but already a huge progress from GAN

- Many many cool and creative applications that we haven’t imagined before due to lack of such a powerful tool!

# GAN vs Diffusion Model

Do you think GAN would disappear in few years and Diffusion model would be the de-facto for generative models?