Diffusion models for text-to-image generation

Shichang Zhang

Oct. 2022

Some slides and figures are taken from
DALL-E by Adam Kutchak, George Lu, Fernando Treviño, and Sarah Wilson (https://www.crcv.ucf.edu/wp-content/uploads/2018/11/DALL-E.pdf)
CVPR 2022 tutorial by Karsten Kreis, Ruiqi Gao, and Arash Vahdat (https://cvpr2022-tutorial-diffusion-models.github.io/)

Outline

- Papers to cover (all from OpenAI)
  - CLIP: Learning Transferable Visual Models From Natural Language Supervision (ICML 2021)
  - DALL-E: Zero-Shot Text-to-Image Generation (ICML 2021)
  - Guided diffusion: Diffusion Models Beat GANs on Image Synthesis (NeurIPS 2021)
  - GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models (ICML 2022)
  - DALL-E 2 (unCLIP): Hierarchical Text-Conditional Image Generation with CLIP Latents
- Disco diffusion
  - Live demo

My personal opinions

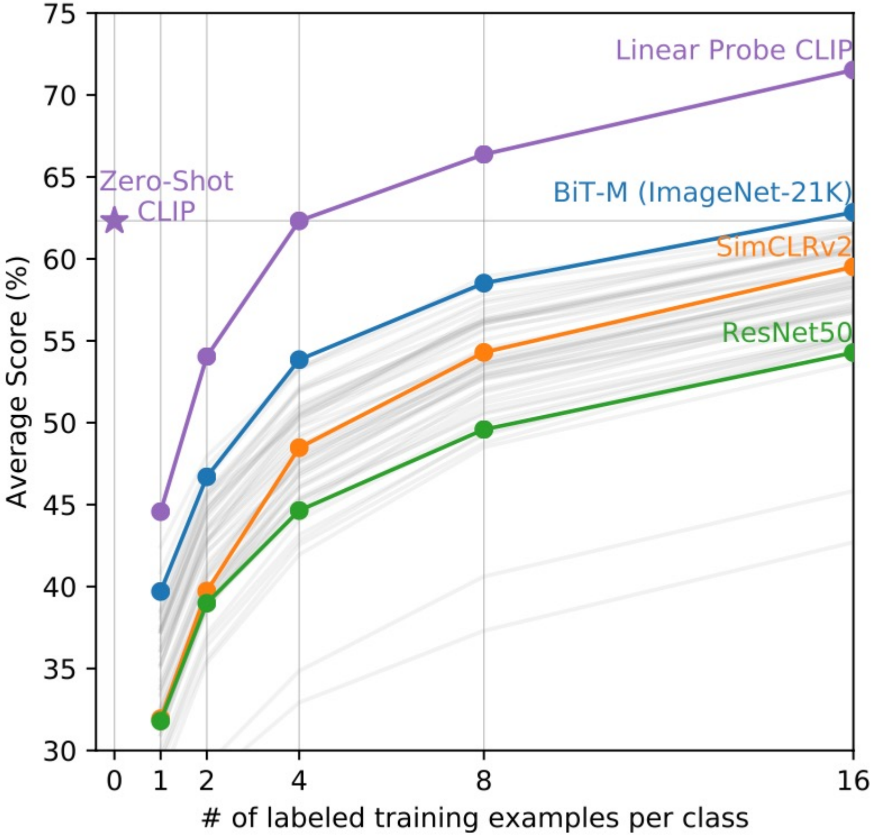
- Research-wise
  - Paper novelty is less significant
  - Paper insight is not deep
  - Paper writing can be improved
- Engineering-wise
  - Many implementation tricks
    - For large-scale training
    - For model performance
  - Huge amount of computational resources
- Business-wise
  - Very big social impact

Outline

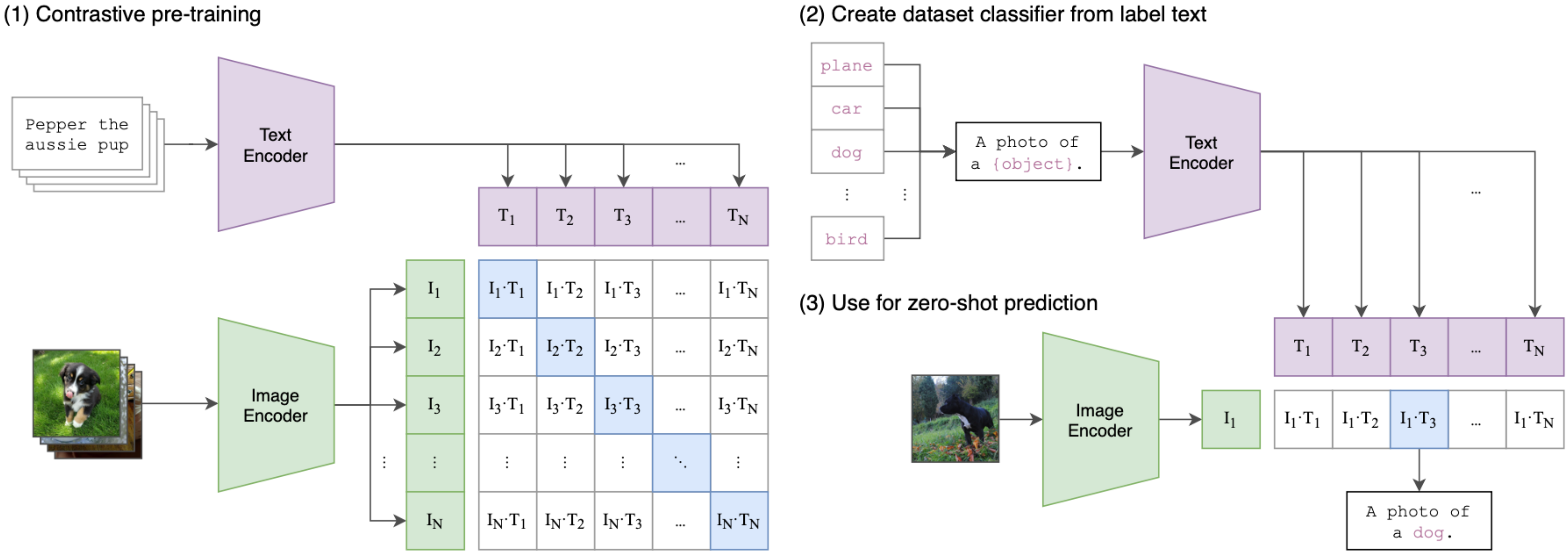
- CLIP: Learning Transferable Visual Models From Natural Language Supervision
- DALL-E: Zero-Shot Text-to-Image Generation
- Guided diffusion: Diffusion Models Beat GANs on Image Synthesis
- GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models
- DALL-E 2 (unCLIP): Hierarchical Text-Conditional Image Generation with CLIP Latents

CLIP: Learning Transferable Visual Models From Natural Language Supervision

- CLIP = contrastive learning between images and text
- Main contribution: 400 million (image, text) pairs



Contrastive learning and zero-shot prediction



	T_1	T_2	T_3	...	T_N
I_1	$I_1 \cdot T_1$	$I_1 \cdot T_2$	$I_1 \cdot T_3$	...	$I_1 \cdot T_N$
I_2	$I_2 \cdot T_1$	$I_2 \cdot T_2$	$I_2 \cdot T_3$	...	$I_2 \cdot T_N$
I_3	$I_3 \cdot T_1$	$I_3 \cdot T_2$	$I_3 \cdot T_3$	...	$I_3 \cdot T_N$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
I_N	$I_N \cdot T_1$	$I_N \cdot T_2$	$I_N \cdot T_3$	...	$I_N \cdot T_N$

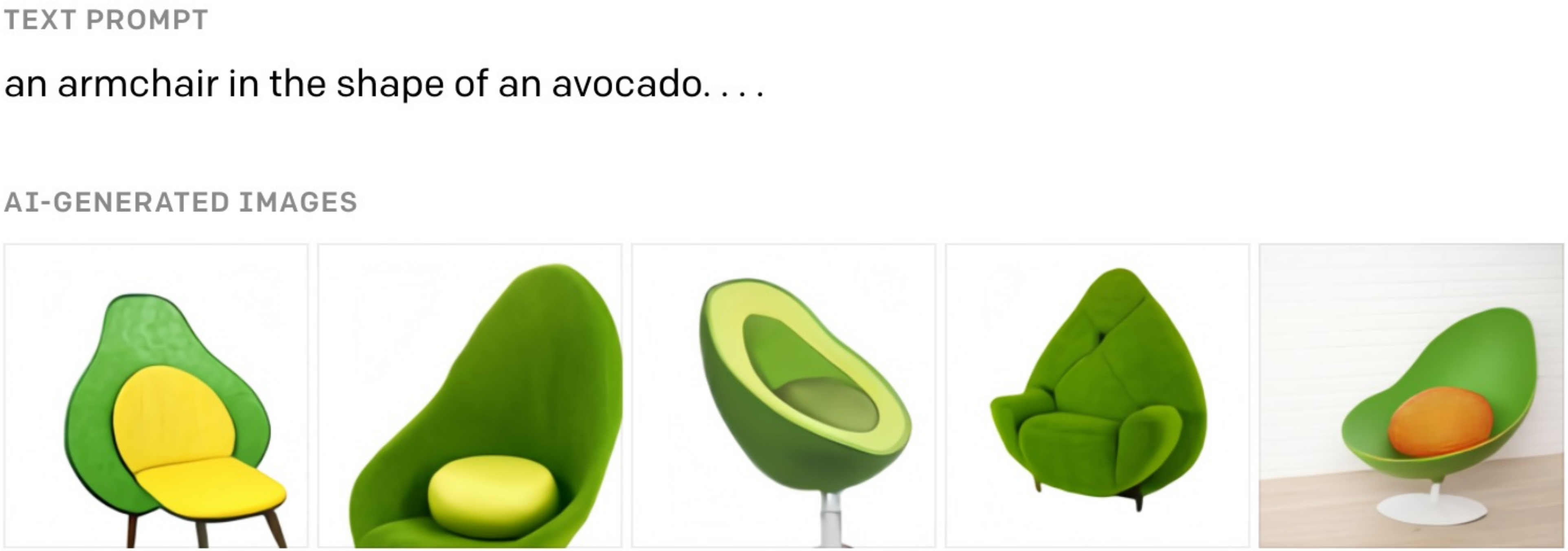
T_1	T_2	T_3	...	T_N
$I_1 \cdot T_1$	$I_1 \cdot T_2$	$I_1 \cdot T_3$	...	$I_1 \cdot T_N$

Outline

- CLIP: Learning Transferable Visual Models From Natural Language Supervision
- DALL-E: Zero-Shot Text-to-Image Generation
- Guided diffusion: Diffusion Models Beat GANs on Image Synthesis
- GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models
- DALL-E 2 (unCLIP): Hierarchical Text-Conditional Image Generation with CLIP Latents

DALL-E: Zero-Shot Text-to-Image Generation

- How to combine text information and image information?
- DALL-E = discrete VAE (VQ-VAE) + Transformer

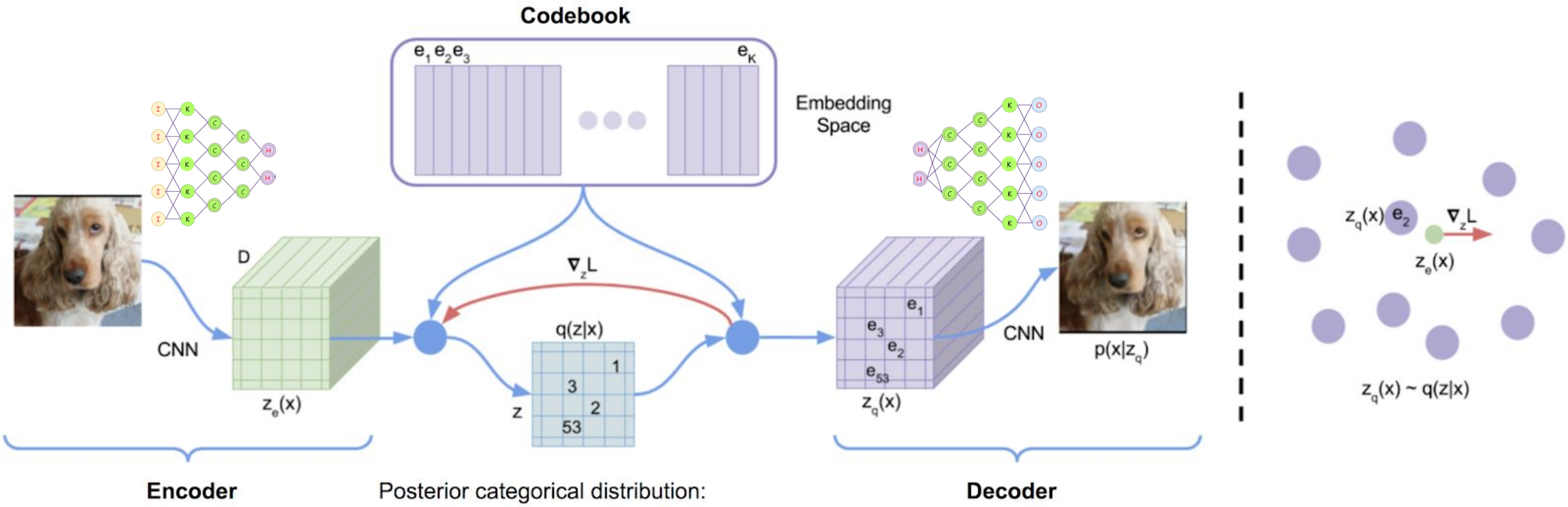


TEXT PROMPT

an armchair in the shape of an avocado. . . .

AI-GENERATED IMAGES

VQ-VAE: Vector-Quantized VAE

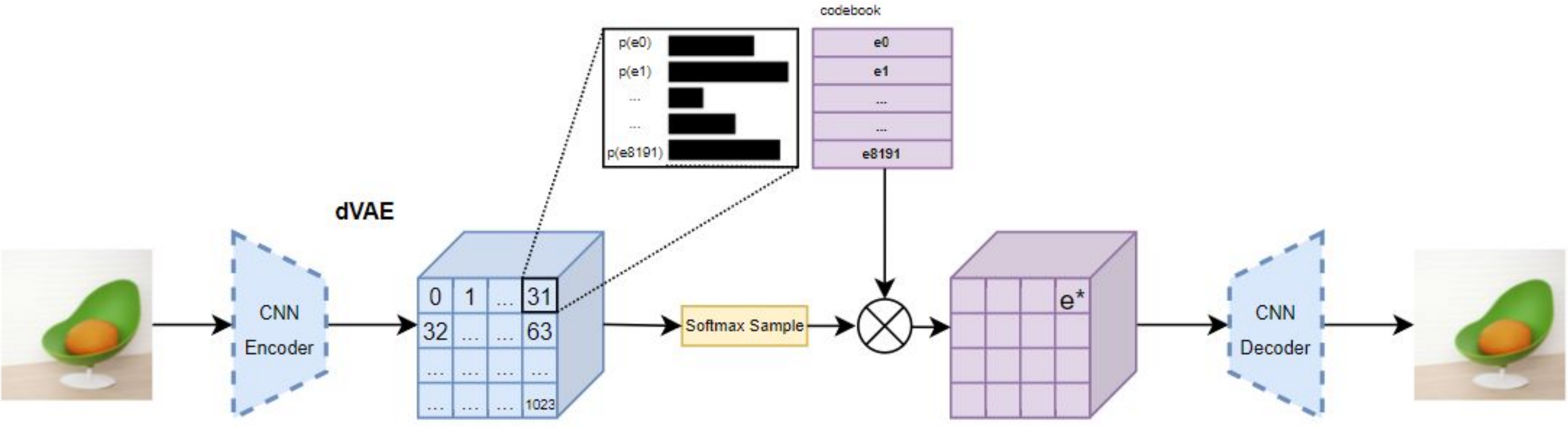


Also see Shichang Zhang's slides for a paper reading group in Spring 2020
https://drive.google.com/file/d/1jA6um5ILlr13IdrW5MP36MyTpsIZ-GUP/view

Learning visual codebook with dVAE

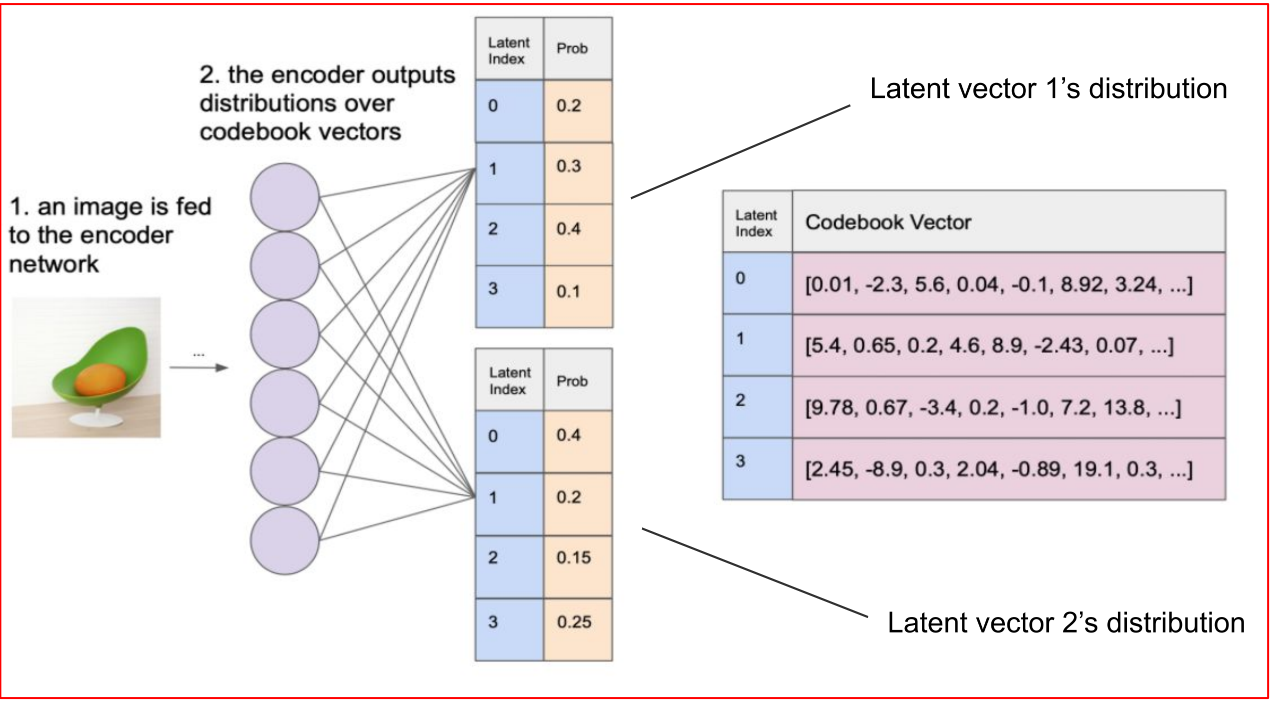
NOT the ICLR 2017 paper “DISCRETE VARIATIONAL AUTOENCODERS” by Jason Tyler Rolfe

- Discrete Variational Autoencoder (dVAE)
  - Similar to VQ-VAE but uses distributions instead of nearest neighbors



dVAE encoding and decoding

Encoding



1. an image is fed to the encoder network

2. the encoder outputs distributions over codebook vectors

Latent vector 1's distribution

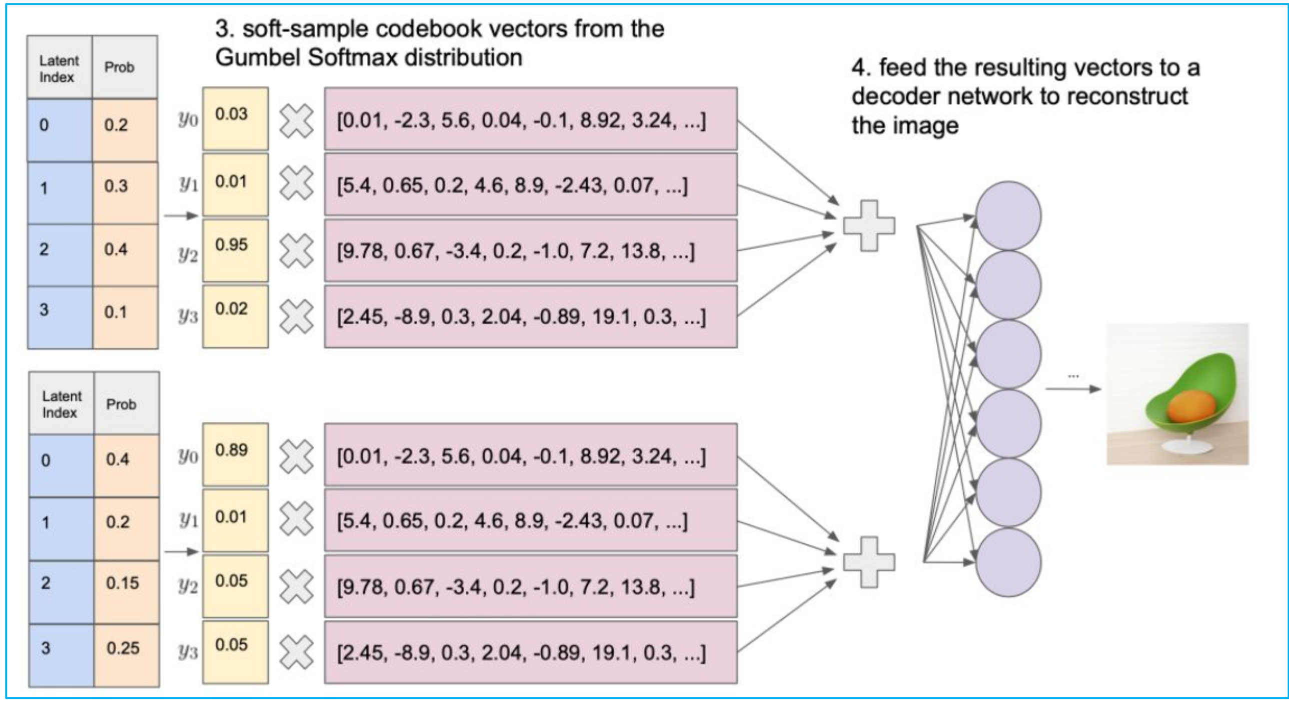
Latent Index	Prob
0	0.2
1	0.3
2	0.4
3	0.1

Latent vector 2's distribution

Latent Index	Prob
0	0.4
1	0.2
2	0.15
3	0.25

Latent Index	Codebook Vector
0	[0.01, -2.3, 5.6, 0.04, -0.1, 8.92, 3.24, ...]
1	[5.4, 0.65, 0.2, 4.6, 8.9, -2.43, 0.07, ...]
2	[9.78, 0.67, -3.4, 0.2, -1.0, 7.2, 13.8, ...]
3	[2.45, -8.9, 0.3, 2.04, -0.89, 19.1, 0.3, ...]

Decoding



3. soft-sample codebook vectors from the Gumbel Softmax distribution

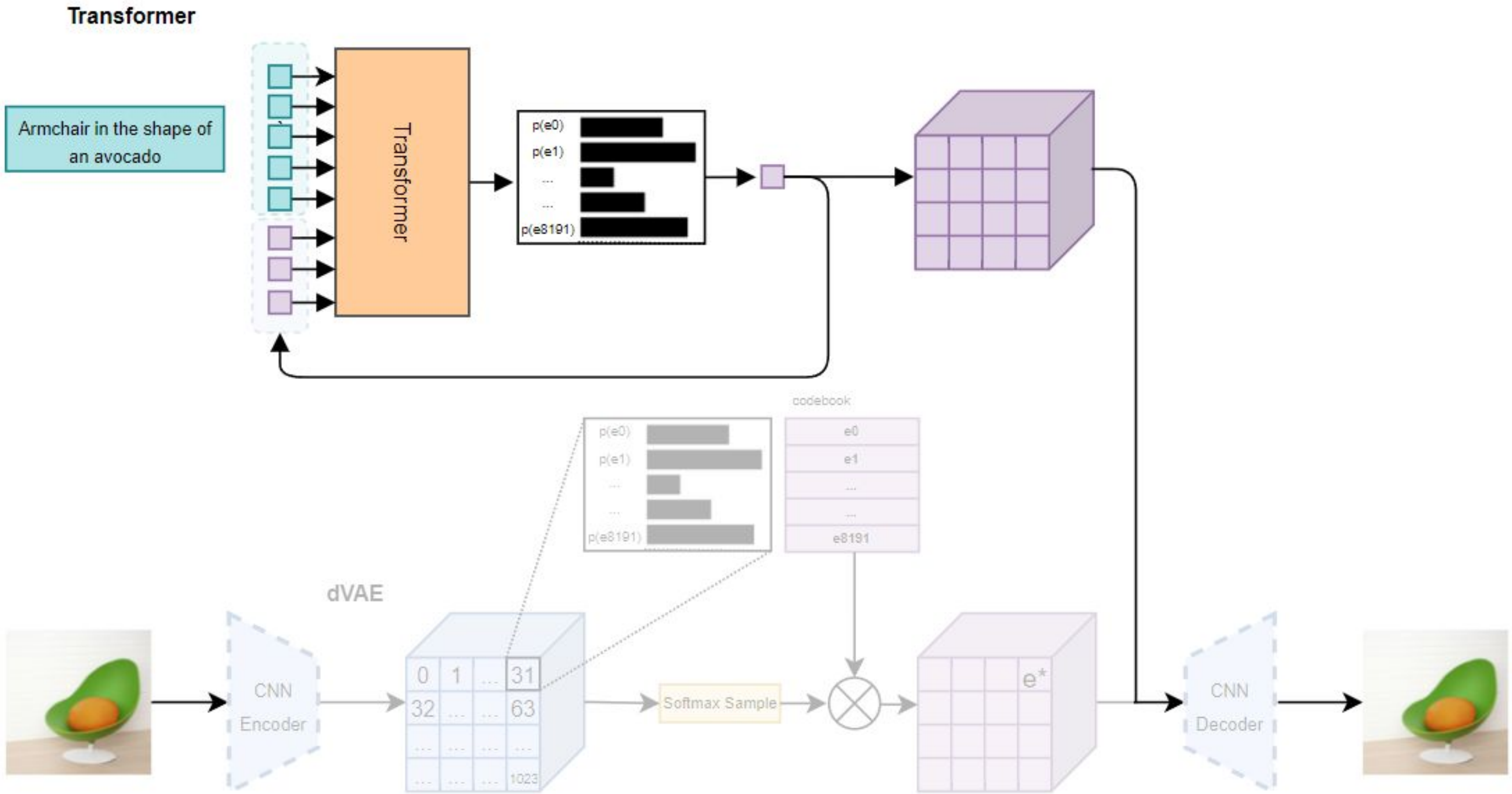
Latent Index	Prob			
0	0.2	y_0	0.03	[0.01, -2.3, 5.6, 0.04, -0.1, 8.92, 3.24, ...]
1	0.3	y_1	0.01	[5.4, 0.65, 0.2, 4.6, 8.9, -2.43, 0.07, ...]
2	0.4	y_2	0.95	[9.78, 0.67, -3.4, 0.2, -1.0, 7.2, 13.8, ...]
3	0.1	y_3	0.02	[2.45, -8.9, 0.3, 2.04, -0.89, 19.1, 0.3, ...]

Latent Index	Prob			
0	0.4	y_0	0.89	[0.01, -2.3, 5.6, 0.04, -0.1, 8.92, 3.24, ...]
1	0.2	y_1	0.01	[5.4, 0.65, 0.2, 4.6, 8.9, -2.43, 0.07, ...]
2	0.15	y_2	0.05	[9.78, 0.67, -3.4, 0.2, -1.0, 7.2, 13.8, ...]
3	0.25	y_3	0.05	[2.45, -8.9, 0.3, 2.04, -0.89, 19.1, 0.3, ...]

4. feed the resulting vectors to a decoder network to reconstruct the image

Learning prior distribution

- Transformer for generating image tokens autoregressively



Outline

- CLIP: Learning Transferable Visual Models From Natural Language Supervision
- DALL-E: Zero-Shot Text-to-Image Generation
- Guided diffusion: Diffusion Models Beat GANs on Image Synthesis
- GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models
- DALL-E 2 (unCLIP): Hierarchical Text-Conditional Image Generation with CLIP Latents

Guided diffusion: Diffusion Models Beat GANs on Image Synthesis

- Guided diffusion = Diffusion model + Learnable variance + Classifier guidance

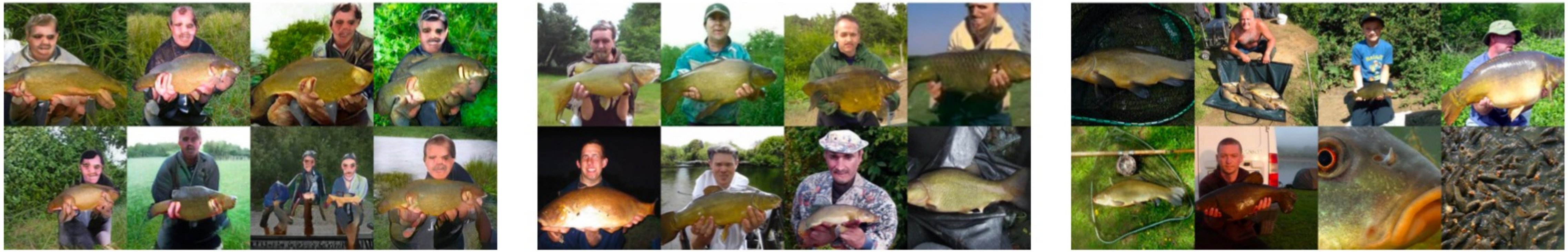
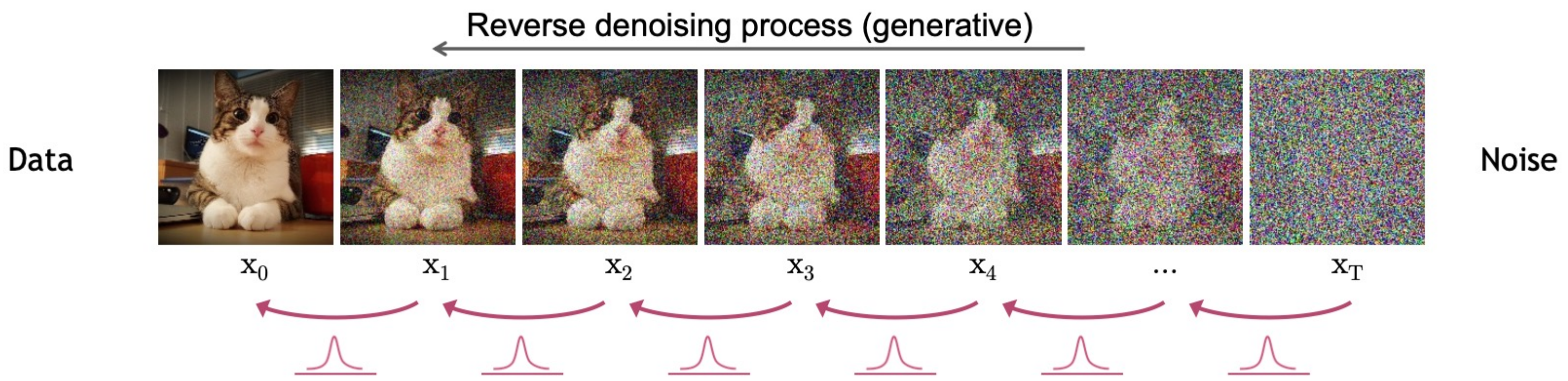


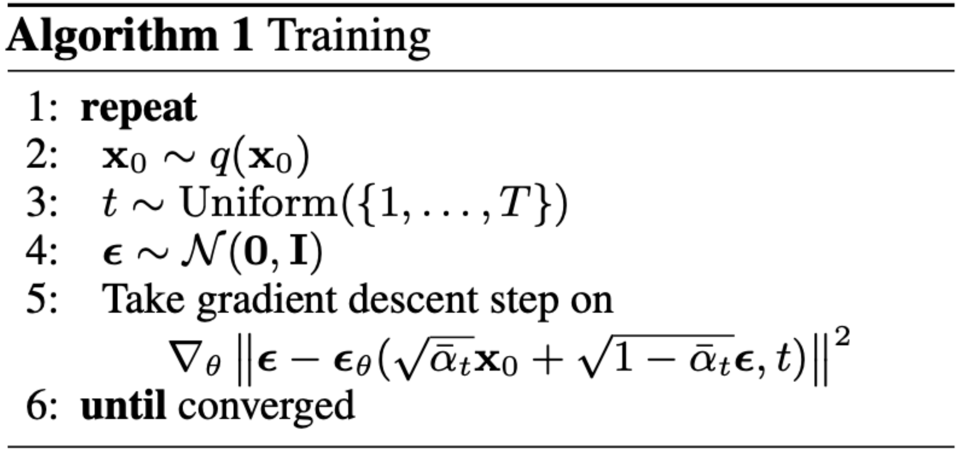
Figure 6: Samples from BigGAN-deep with truncation 1.0 (FID 6.95, left) vs samples from our diffusion model with guidance (FID 4.59, middle) and samples from the training set (right).

Fréchet inception distance (FID). Fréchet distance is equivalent to Wasserstein-2 distance. The smaller the better.

$$d_F(\mu, \nu) := \left(\inf_{\gamma \in \Gamma(\mu,\nu)} \int_{\mathbb{R}^n \times \mathbb{R}^n} \|x - y\|^2 \, \mathrm{d}\gamma(x, y) \right)^{1/2}$$

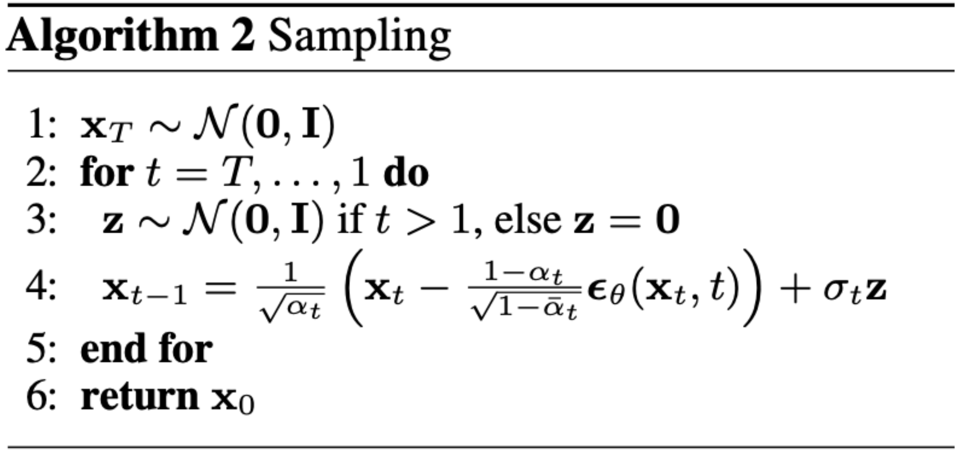
DDPM: Denoising Diffusion Probabilistic Models





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

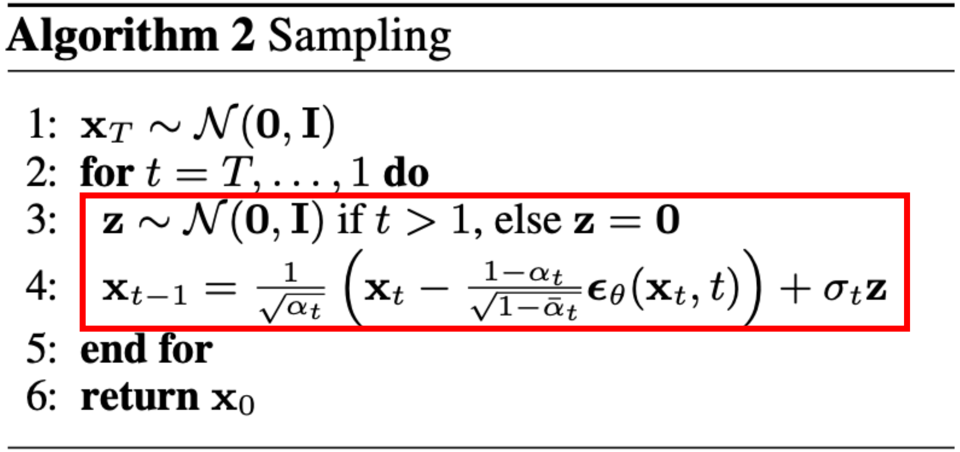
```
$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, \ldots, 1$ do
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
  $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t,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

“Train the reverse process mean function approximator to predict μ , or by modifying its parameterization, predict ϵ”

Sampling

Guided diffusion

DDPM

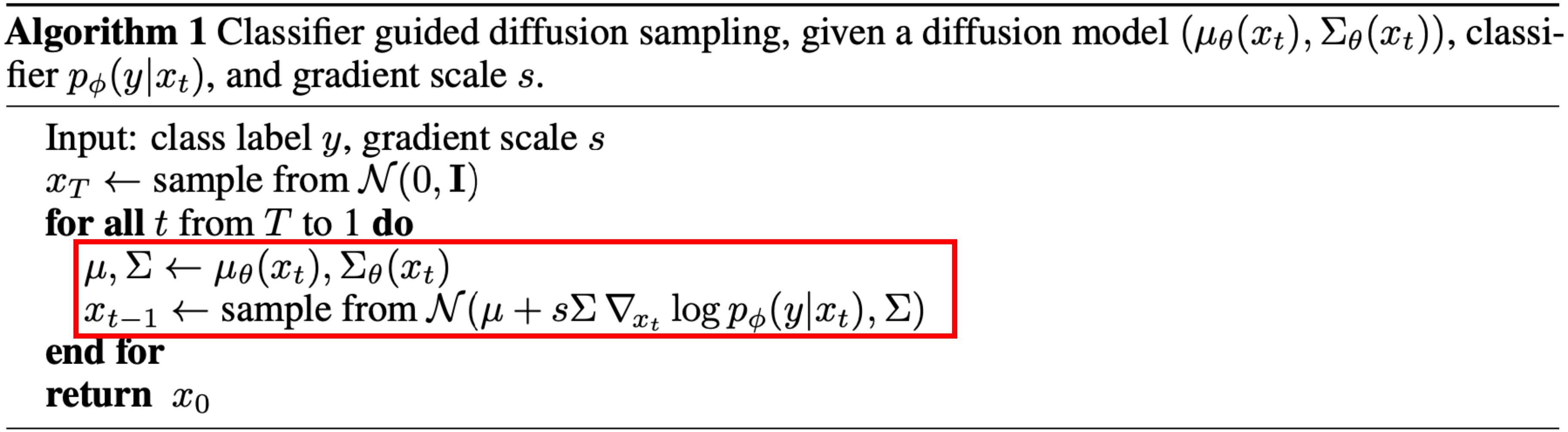


Algorithm 2 Sampling

```
$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, \dots, 1$ 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 $\mathbf{x}_0$
```

$$\mu \leftarrow \mu_\theta(\mathbf{x_t})$$
$$\boldsymbol{\Sigma} \leftarrow \sigma_\mathbf{t}$$
$$\mathbf{x_{t-1}} \leftarrow \text{sample from } \mathcal{N}(\mu, \boldsymbol{\Sigma})$$

+ Learnable variance
+ Classifier 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
    $\mu, \Sigma \leftarrow \mu_\theta(x_t), \Sigma_\theta(x_t)$
    $x_{t-1} \leftarrow$ sample from $\mathcal{N}(\mu + s\Sigma\, \nabla_{x_t} \log p_\phi(y|x_t), \Sigma)$
end for
return $x_0$
```

$$\begin{aligned}&\mu, \Sigma \leftarrow \mu_\theta(x_t), \Sigma_\theta(x_t)\\&x_{t-1} \leftarrow \text{sample from } \mathcal{N}(\mu + s\Sigma\, \nabla_{x_t} \log p_\phi(y|x_t), \Sigma)\end{aligned}$$

Classifier guidance

- Move image towards the direction that maximizes the label probability
- The scaling factor

$s \cdot \nabla_x \log p(y|x) = \nabla_x \log \frac{1}{Z} p(y|x)^s$ Sharpen $\log p(y|x)$ to $\frac{1}{Z} p(y|x)^s$

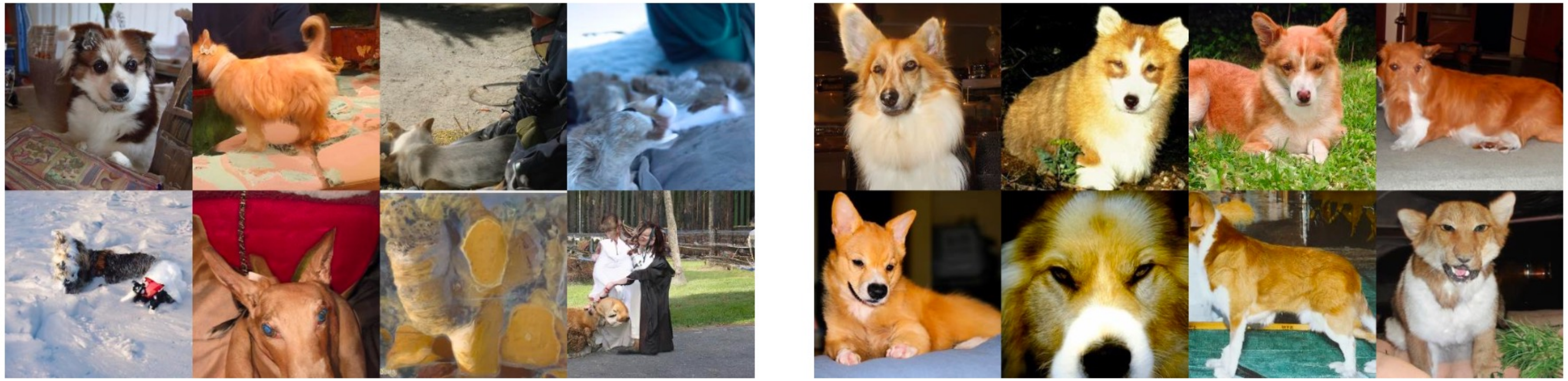


Figure 3: Samples from an unconditional diffusion model with classifier guidance to condition on the class "Pembroke Welsh corgi". Using classifier scale 1.0 (left; FID: 33.0) does not produce convincing samples in this class, whereas classifier scale 10.0 (right; FID: 12.0) produces much more class-consistent images.

Outline

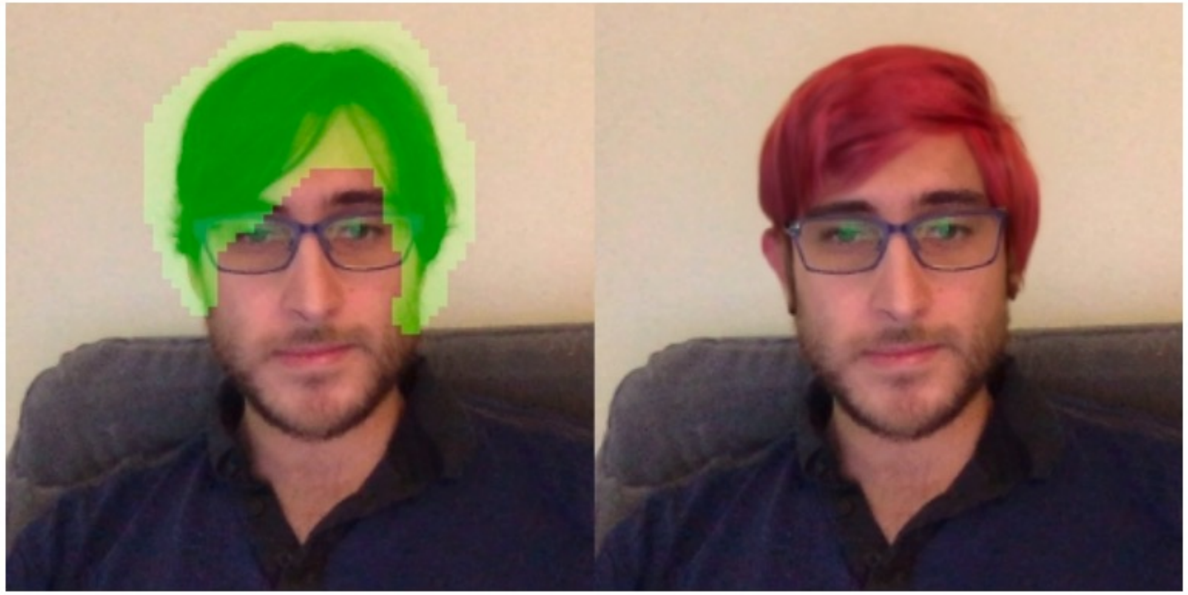
- CLIP: Learning Transferable Visual Models From Natural Language Supervision
- DALL-E: Zero-Shot Text-to-Image Generation
- Guided diffusion: Diffusion Models Beat GANs on Image Synthesis
- GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models
- DALL-E 2 (unCLIP): Hierarchical Text-Conditional Image Generation with CLIP Latents

GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models

- How to combine text information and image information?
- GLIDE = Diffusion model + Classifier-free guidance

"a hedgehog using a calculator"

"a corgi wearing a red bowtie and a purple party hat"



"a man with red hair"

Conditional diffusion model

- Encode condition when doing mean(error) sampling

Regular diffusion

(DDPM)

$$p_\theta(x_{t-1}|x_t) \coloneqq \mathcal{N}(\mu_\theta(x_t), \Sigma_\theta(x_t))$$

Condition on label y

(with classifier guidance)

$$\hat{\mu}_\theta(x_t|y) = \mu_\theta(x_t|y) + s \cdot \Sigma_\theta(x_t|y)\nabla_{x_t} \log p_\phi(y|x_t)$$

Condition on caption c

(with CLIP guidance)

$$\hat{\mu}_\theta(x_t|c) = \mu_\theta(x_t|c) + s \cdot \Sigma_\theta(x_t|c)\nabla_{x_t} \left(f(x_t) \cdot g(c)\right)$$

Condition on label y or caption c

(with classifier-free guidance)

$$\hat{\epsilon}_\theta(x_t|y) = \epsilon_\theta(x_t|\emptyset) + s \cdot (\epsilon_\theta(x_t|y) - \epsilon_\theta(x_t|\emptyset))$$

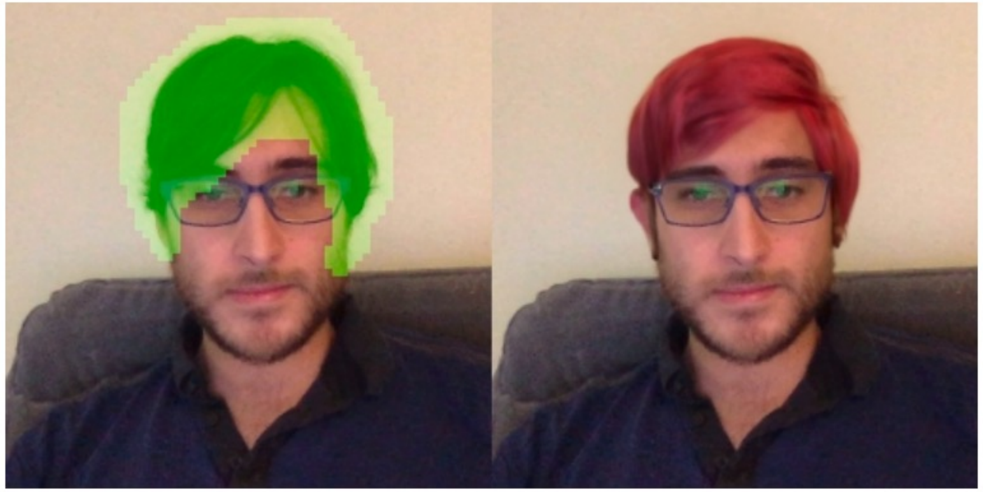
$$\hat{\epsilon}_\theta(x_t|c) = \epsilon_\theta(x_t|\emptyset) + s \cdot (\epsilon_\theta(x_t|c) - \epsilon_\theta(x_t|\emptyset))$$

Guidance for Text-to-image generation

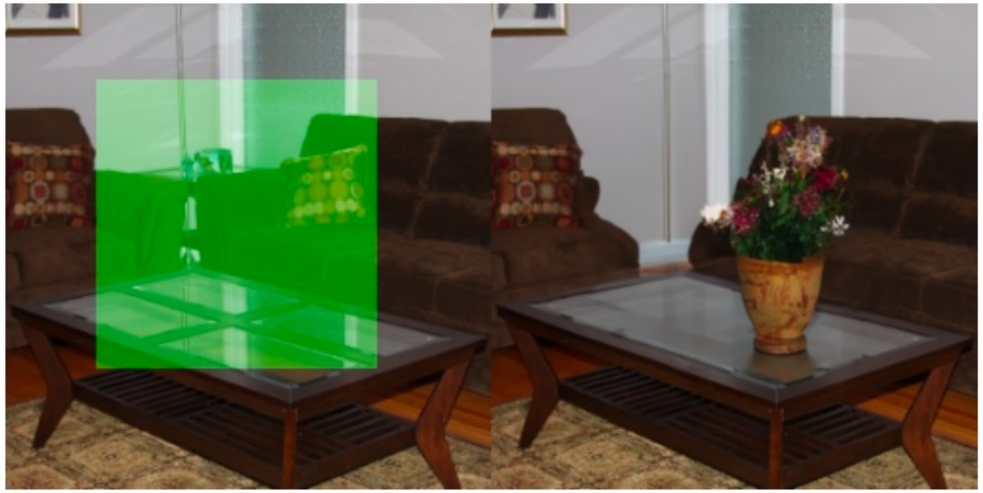
- Classifier-free guidance outperforms CLIP guidance
- Advantage of classifier-free guidance
  - Doesn’t rely on the knowledge of a separate classification model
  - Works for both class labels and more complicated condition (caption)

GLIDE for image inpainting

- Diffusion model inpainting can be performed by sampling from the diffusion model as usual, but replacing the known region of the image with a sample from $q(xt|x0)$ after each sampling step



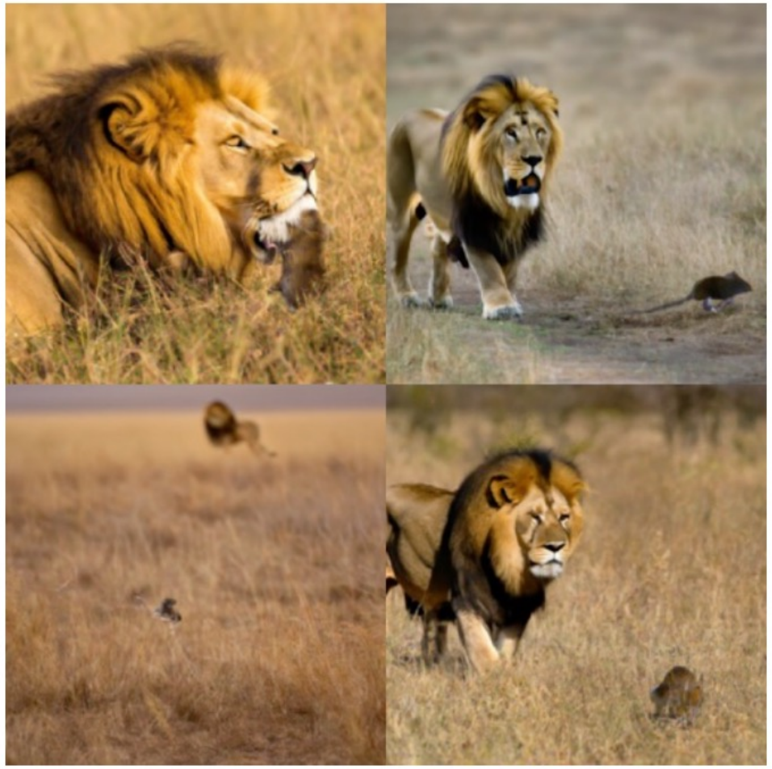
“a man with red hair”



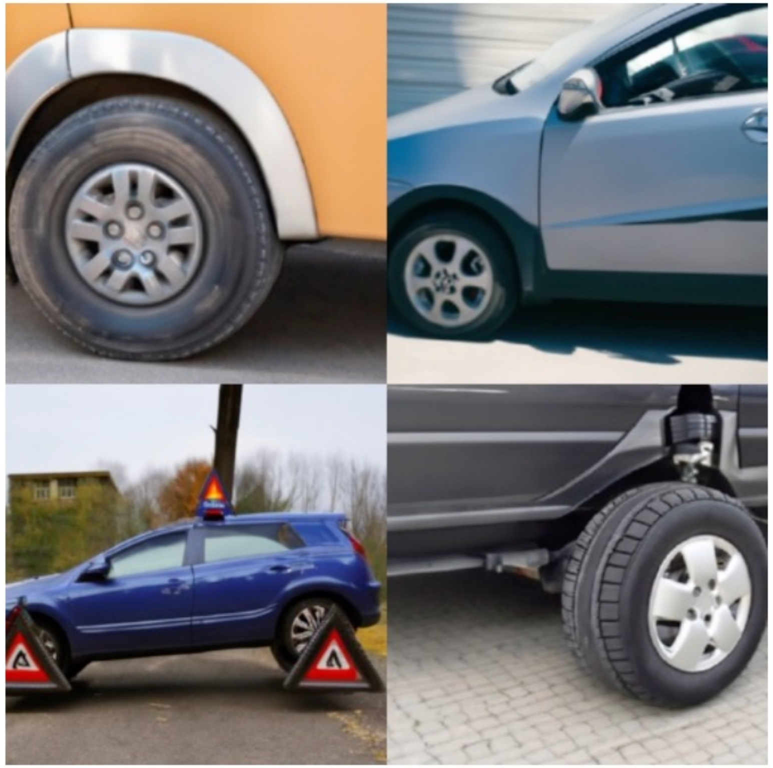
“a vase of flowers”

Limitations

- Generate unusual scenarios is hard



“a mouse hunting a lion”



“a car with triangular wheels”

Outline

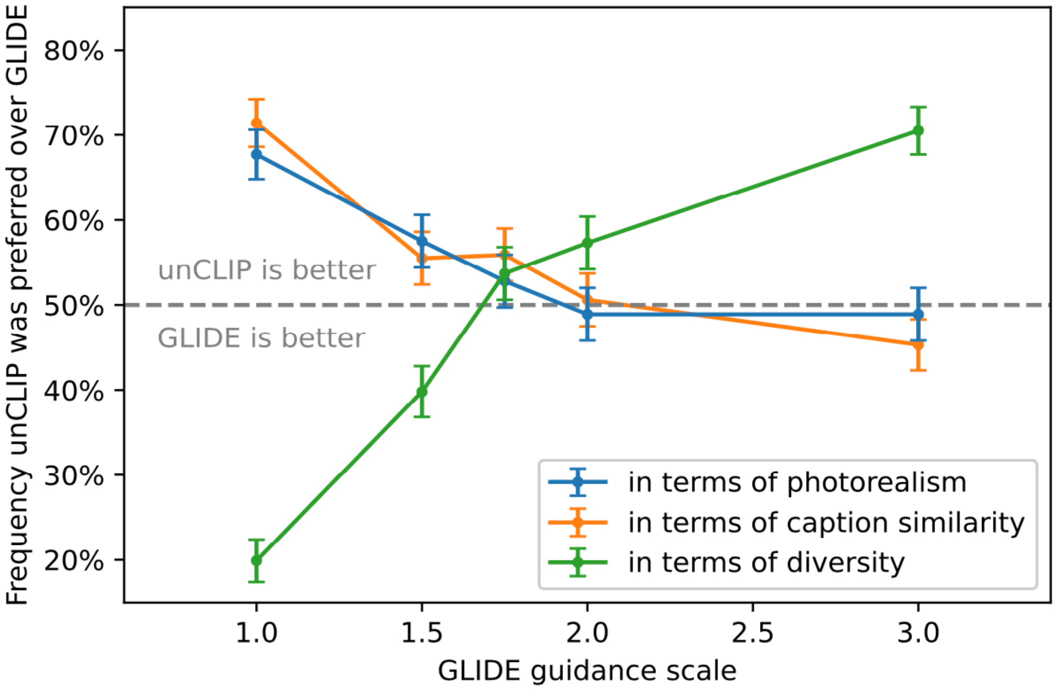
- CLIP: Learning Transferable Visual Models From Natural Language Supervision
- DALL-E: Zero-Shot Text-to-Image Generation
- Guided diffusion: Diffusion Models Beat GANs on Image Synthesis
- GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models
- DALL-E 2 (unCLIP): Hierarchical Text-Conditional Image Generation with CLIP Latents

DALL-E 2: Hierarchical Text-Conditional Image Generation with CLIP Latents (unCLIP)

- DALL-E 2 = Diffusion model + CLIP embeddings

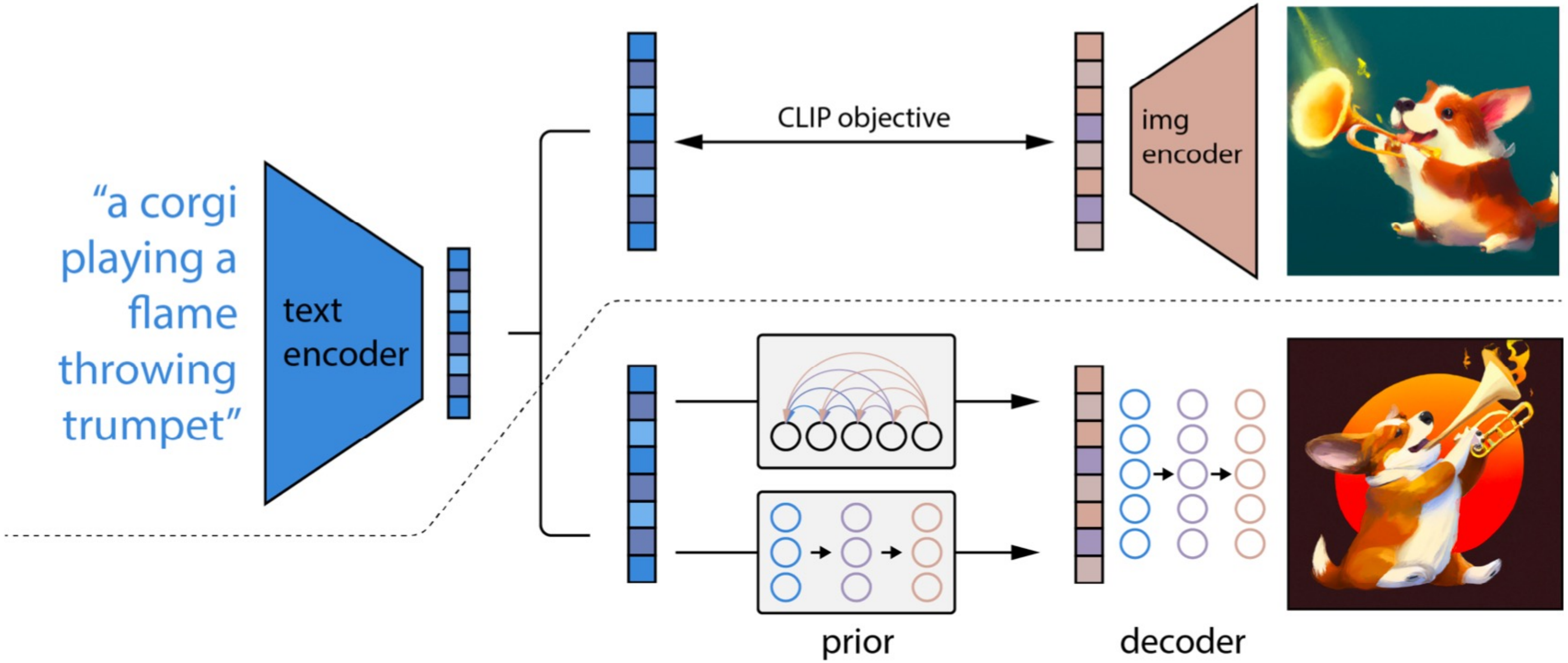
panda mad scientist mixing sparkling chemicals, artstation

a corgi’s head depicted as an explosion of a nebula



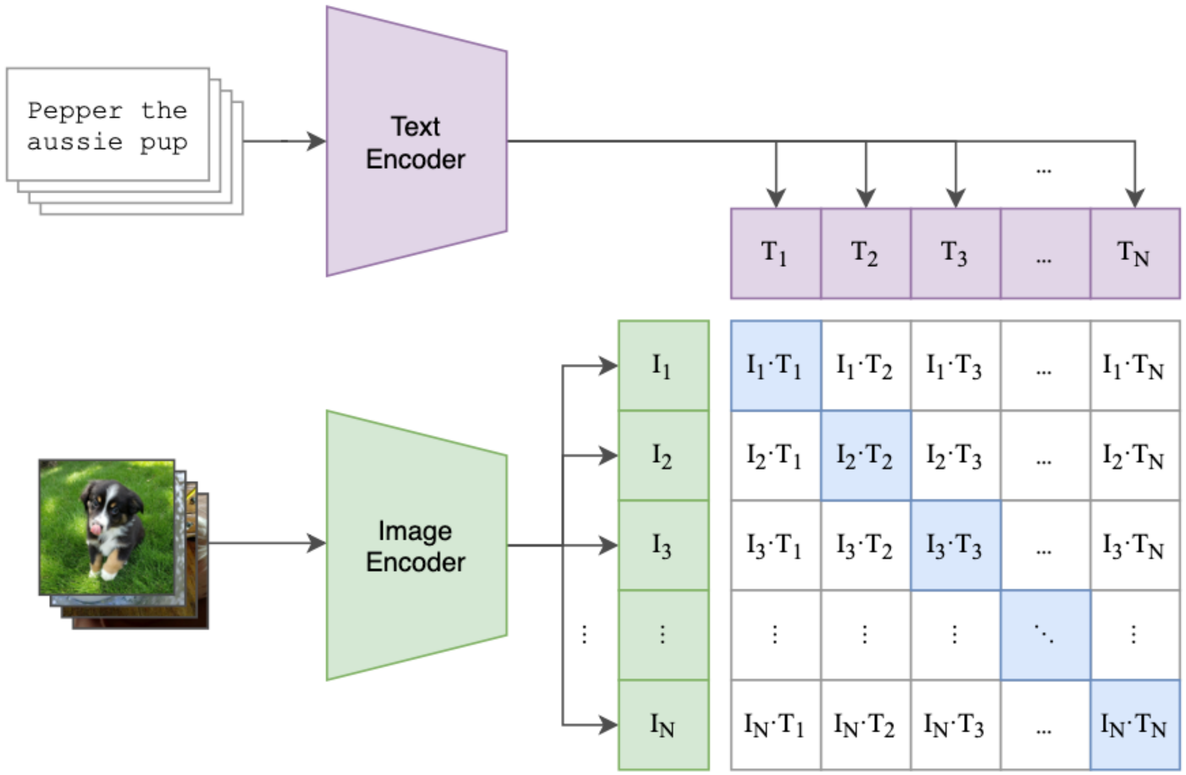
unCLIP

- A decoder that produces images conditioned on CLIP image embeddings (and optionally text captions)
- A prior that produces CLIP image embeddings conditioned on captions



Reminder: CLIP

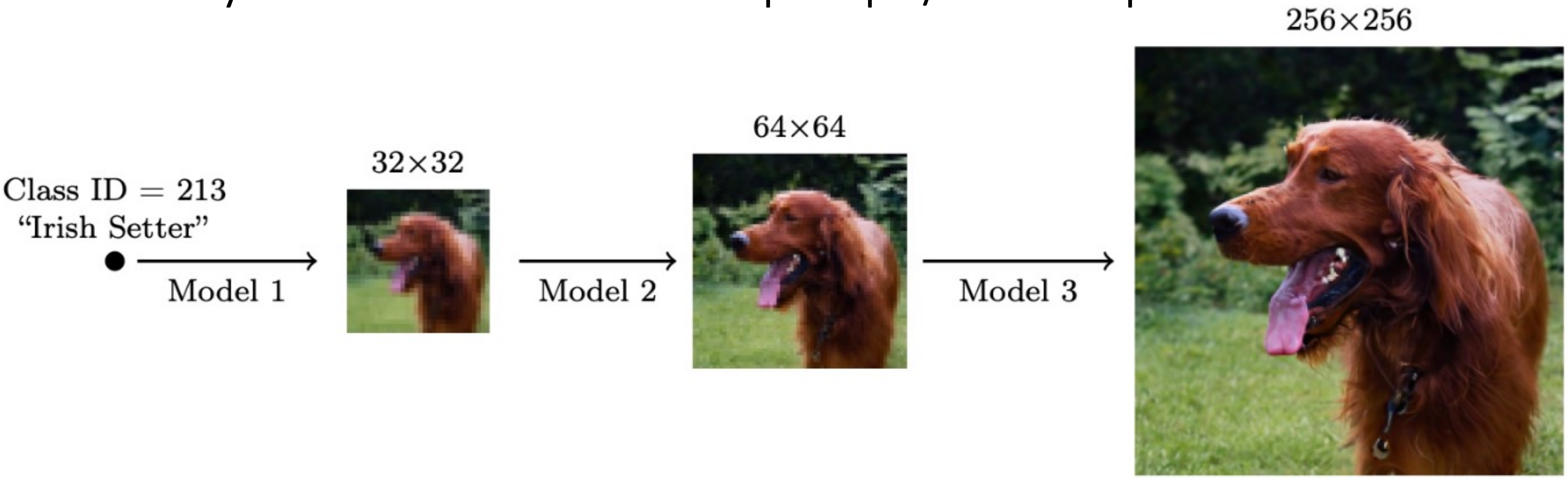
(1) Contrastive pre-training



	T_1	T_2	T_3	...	T_N
I_1	$I_1 \cdot T_1$	$I_1 \cdot T_2$	$I_1 \cdot T_3$	...	$I_1 \cdot T_N$
I_2	$I_2 \cdot T_1$	$I_2 \cdot T_2$	$I_2 \cdot T_3$	...	$I_2 \cdot T_N$
I_3	$I_3 \cdot T_1$	$I_3 \cdot T_2$	$I_3 \cdot T_3$	...	$I_3 \cdot T_N$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
I_N	$I_N \cdot T_1$	$I_N \cdot T_2$	$I_N \cdot T_3$	...	$I_N \cdot T_N$

Decoder

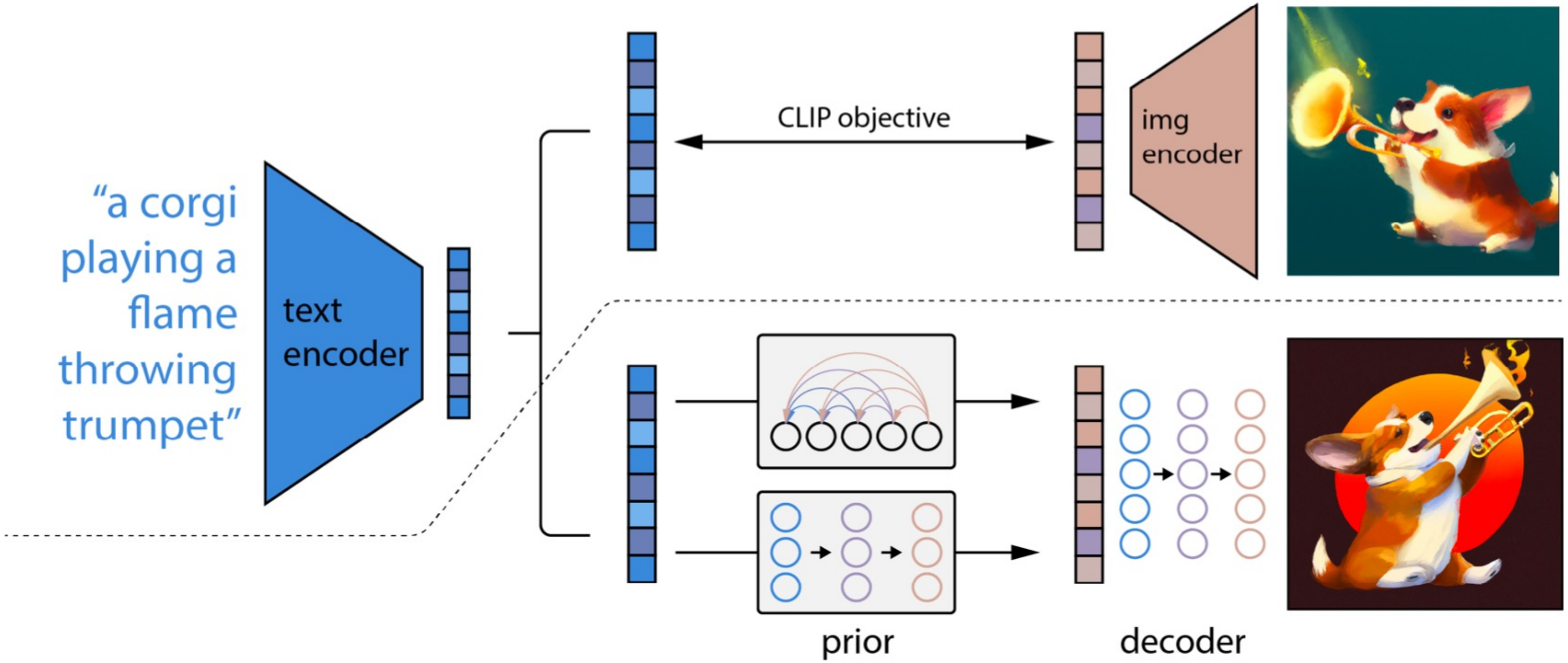
- A conditioned diffusion model
- Two diffusion upsampler
  - Modify the U-Net architecture to upsample/downsample



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

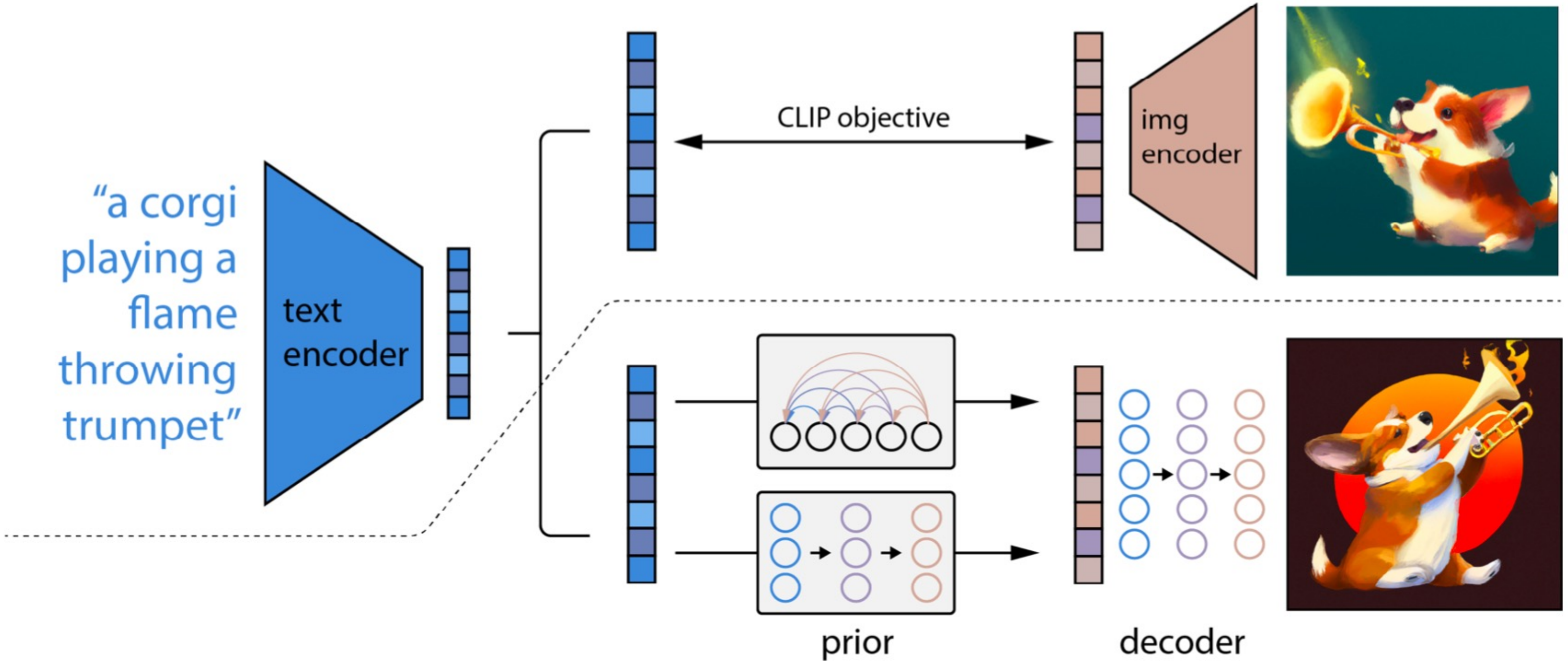
Prior

- Generate image embeddings from Gaussian noise and
  - Condition on text caption
  - Optionally, condition on CLIP text embeddings as well



Different inputs to the decoder

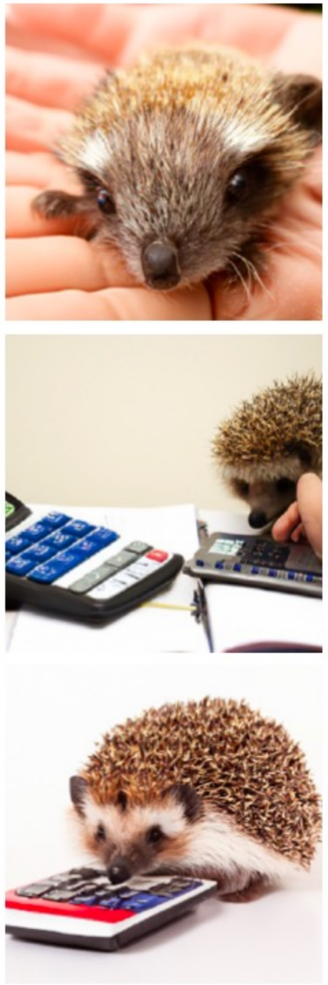
- Why not condition on captions or CLIP text embeddings?
  - Performance is not good



Caption only

Caption + CLIP text embedding

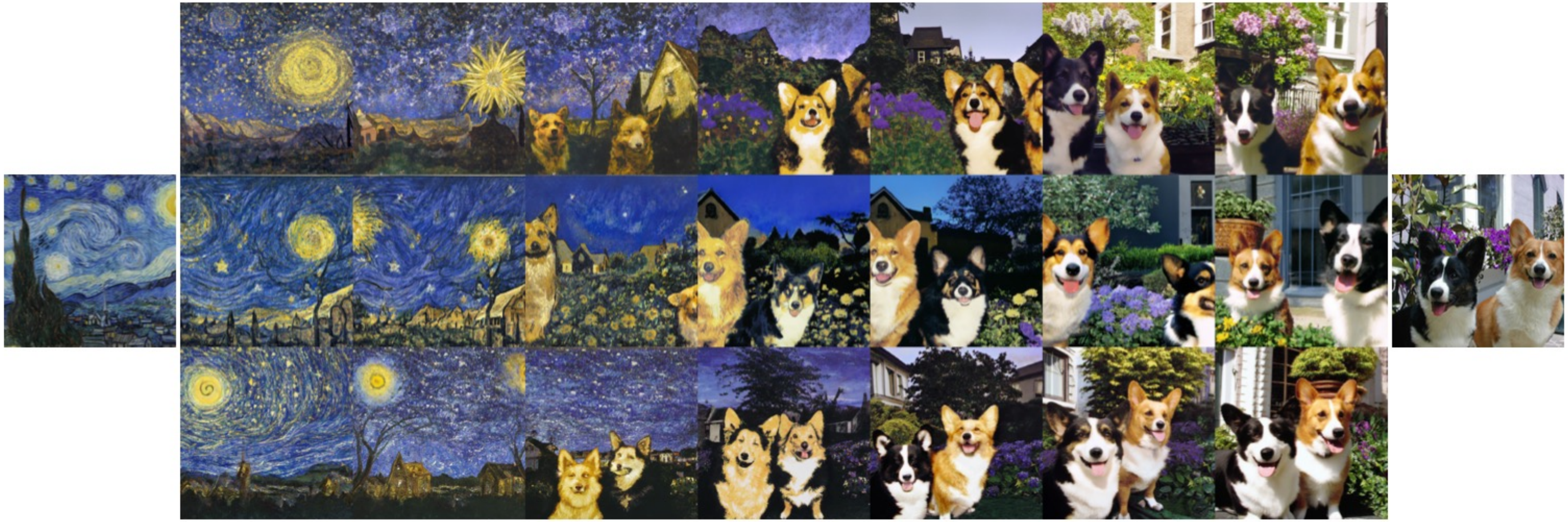
Caption + CLIP image embedding



"a hedgehog using a calculator"

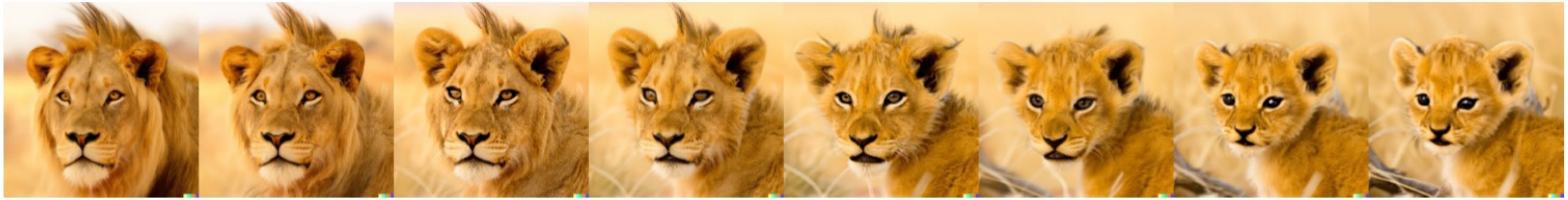
Interpolation

- Interpolate image embeddings (for testing the decoder only)



Interpolation

- Interpolate text embeddings

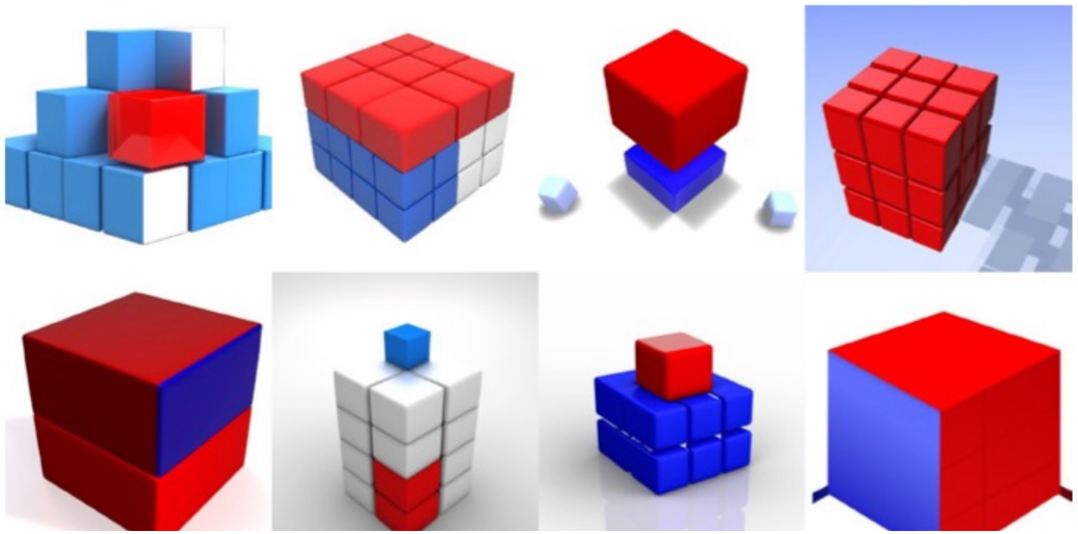


a photo of an adult lion → a photo of lion cub

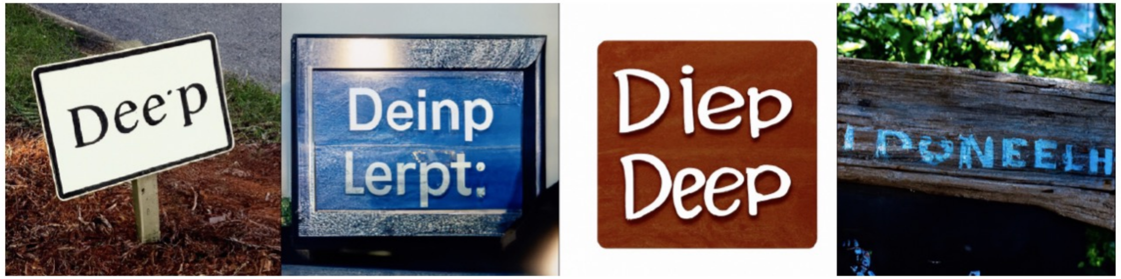
Limitations

- Bind two separate objects (cubes) to two separate attributes (colors)
- Produce coherent text

"A red cube on top of a blue cube"



"A sign that says deep learning"



Summary

- CLIP = contrastive learning between images and text
- DALL-E = discrete VAE (VQ-VAE) + Transformer
- Guided diffusion = Diffusion model + Learnable variance + Classifier guidance
- GLIDE = Diffusion model + Classifier-free guidance
- DALL-E 2 = Diffusion model + CLIP embeddings

Disco Diffusion

- Quick Start on using AI to render images using Disco Diffusion
  - https://www.youtube.com/watch?v=wIw59kAU6u8
  - An introduction for beginners, the only requirement is a Google drive.
- Demo
  - https://colab.research.google.com/github/alembics/disco-diffusion/blob/main/Disco_Diffusion.ipynb
  - or simply google search disco diffusion

Importance factors for image generation

- Model: domain specific is better
- Prompt, image size, diffusion steps, etc

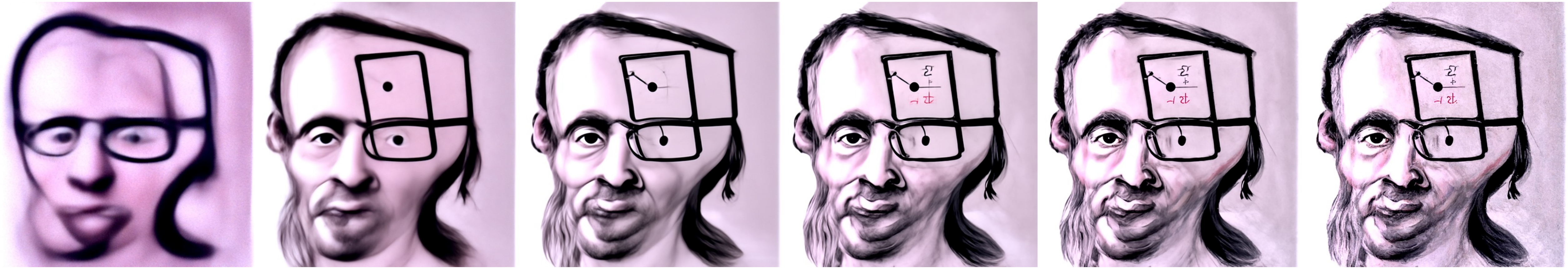
"A beautiful painting of a singular lighthouse, shining its light across a tumultuous sea of blood by Greg Rutkowski and Thomas Kinkade, Trending on artstation."

Example online

Example by me

Disco Diffusion example

- Prompt: “A portrait of the smartest mathematician”
- Diffusion generation:



~8 mins to generate a 600 X 600 image using a 250-step diffusion sampling on a single Tesla T4

Disco Diffusion example

- Observations:
  - Style: black-white, sketch
  - Features: glasses, beard, geometric shapes, formulas, etc.