Diffusion Models Chapter 5: Text-Guided Diffusion Models

Generative AI and Foundation Models

Spring 2024

Department of Mathematical Sciences

Ernest K. Ryu

Seoul National University

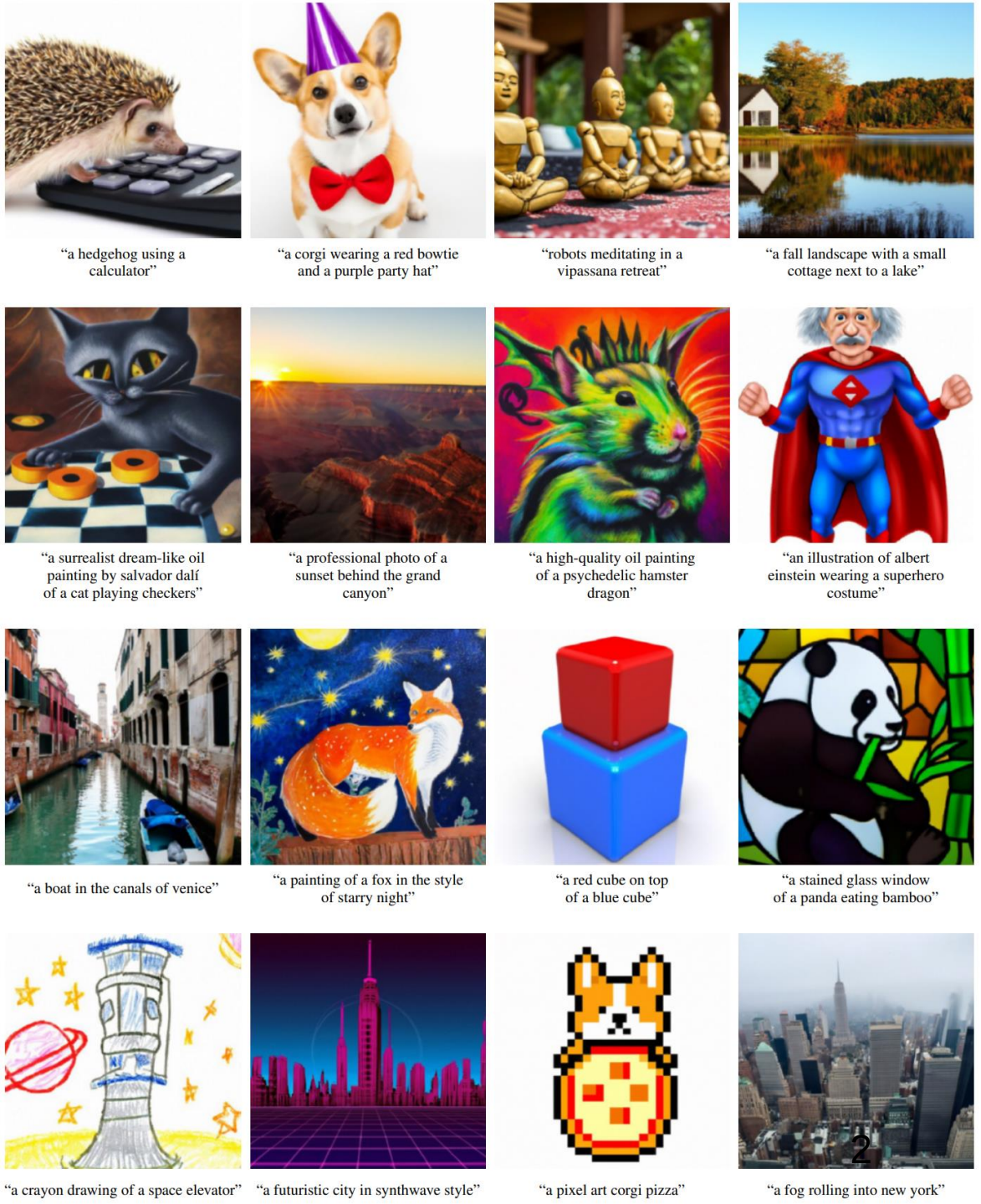
GLIDE

Guided Language to Image Diffusion for Generation and Editing (GLIDE) is a diffusion-based text-to-image model.

These models significantly outperform prior text-to-image models based on GANs.

There are 2 versions of GLIDE that use

1. CLIP guidance
2. Text-conditioning and classifier-free guidance



A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen, GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models, *ICML*, 2022.

GLIDE with CLIP guidance

The first version of GLIDE uses a “pre-trained” CLIP model to perform classifier guidance.

Since

$$\log p(C \mid X) \approx \frac{1}{\tau} f_\theta(X) \cdot g_\phi(C) + \text{constant independent of } X$$

one can consider generating images via the SDE

$$\begin{aligned} d\overline{X}_t &= (f - g^2(\nabla_{\overline{X}_t} \log p_t(\overline{X}_t) + \omega \nabla_{\overline{X}_t} \log p(C \mid \overline{X}_t)))dt + g d\bar{W}_t, \qquad \overline{X}_T \sim p_T \\ &\approx (f - g^2(\nabla_{\overline{X}_t} \log p_t(\overline{X}_t) + \frac{\omega}{\tau} \nabla_{\overline{X}_t} f_\theta(\overline{X}_t) \cdot g_\phi(C)))dt + g d\bar{W}_t, \qquad \overline{X}_T \sim p_T \end{aligned}$$

This would be convenient as we could use a pre-trained CLIP model, but this doesn’t work.

GLIDE with CLIP guidance

The problem is $\log p(C \mid X) \neq \log p(C \mid X_t)$, where X_t is a forward-corrupted version of X. We therefore need a time-dependent CLIP model such that

$$\log p_t(C \mid X_t) \approx \frac{1}{\tau} f_\theta^{(t)}(X_t) \cdot g_\phi^{(t)}(C) + \text{constant independent of } X_t$$

we can then generate images via

$$\begin{aligned} d\overline{X}_t &= (f - g^2(\nabla_{\overline{X}_t} \log p_t(\overline{X}_t) + \omega \nabla_{\overline{X}_t} \log p_t(C \mid \overline{X}_t)))dt + g d\bar{W}_t, \qquad \overline{X}_T \sim p_T \\ &\approx (f - g^2(\nabla_{\overline{X}_t} \log p_t(\overline{X}_t) + \frac{\omega}{\tau} \nabla_{\overline{X}_t} f_\theta^{(t)}(\overline{X}_t) \cdot g_\phi^{(t)}(C)))dt + g d\bar{W}_t, \qquad \overline{X}_T \sim p_T \end{aligned}$$

The time-dependent CLIP model $\left(f_\theta^{(t)}, g_\phi^{(t)}\right)$ is "pre-trained" in the sense that it is trained separately from the score network $s_\psi(X_t, t) \approx \nabla_{X_t} \log p_t(X_t)$.

GLIDE via direct text-conditioning

Direct text-conditioning uses a classifier-free guidance, and it uses a conditional error (score) network

$$\varepsilon_\theta(X_t, t, C)$$

The caption C is encoded into a sequence of K tokens, which are then processes by a transformer model. The caption embeddings are added to the time embeddings.

Furthermore, the K tokens are made available to the attention layers of the U-Net through cross attention, i.e., the K tokens are projected into key and value vectors (but not query vectors) so that the queries (corresponding to pixels) can access them.

DALL·E 2

vibrant portrait painting of Salvador Dalí with a robotic half face

a shiba inu wearing a beret and black turtleneck

a close up of a handpalm with leaves growing from it

an espresso machine that makes coffee from human souls, artstation

panda mad scientist mixing sparkling chemicals, artstation

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

a dolphin in an astronaut suit on saturn, artstation

a propaganda poster depicting a cat dressed as french emperor napoleon holding a piece of cheese

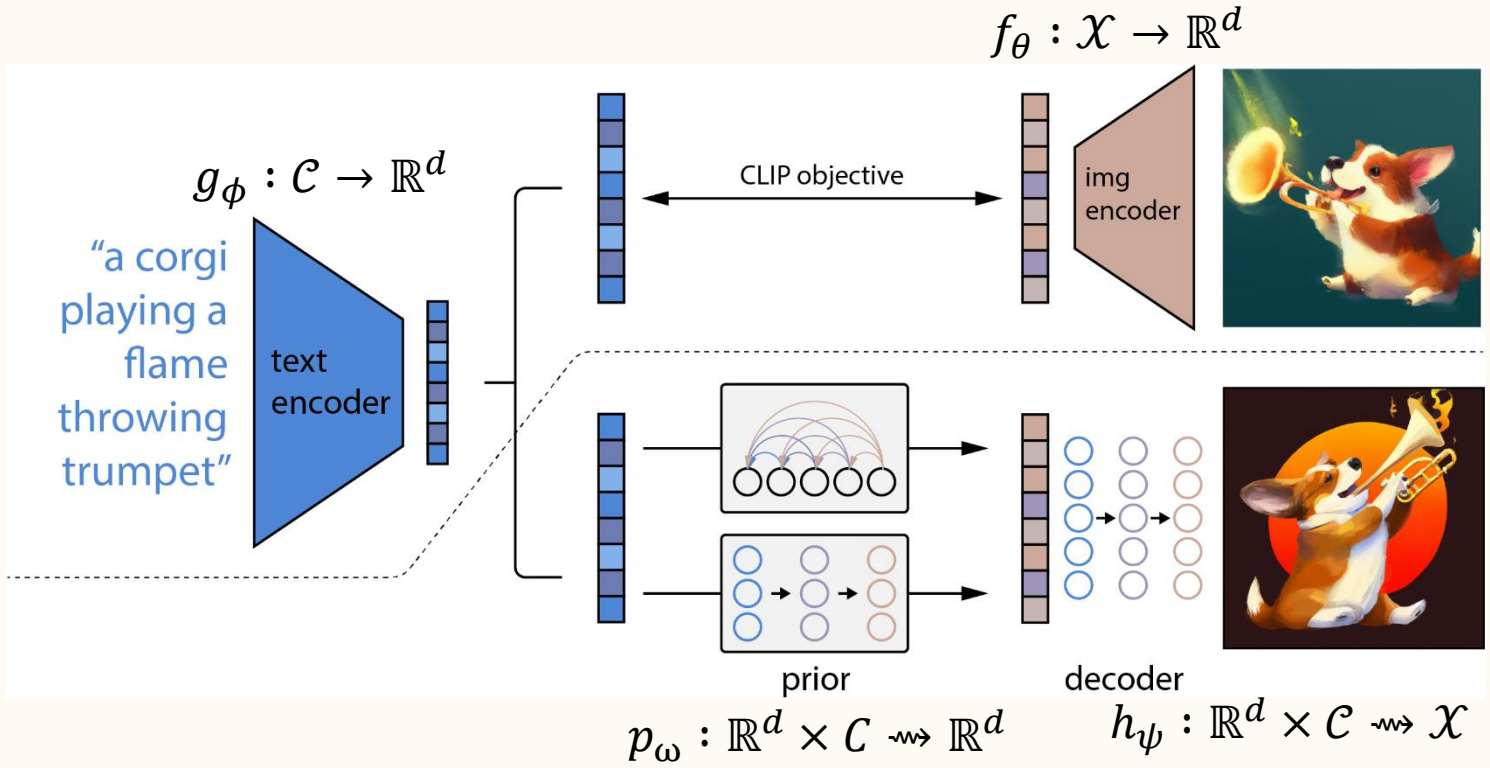
a teddy bear on a skateboard in times square

A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, Hierarchical text-conditional image generation with CLIP latents, *arXiv*, 2022.

DALL·E 2

The DALL·E 2 model consists of 4 neural networks:

- Image encoder f_θ
- Text encoder g_ϕ
- Image decoder h_ψ
- "Prior" p_ω



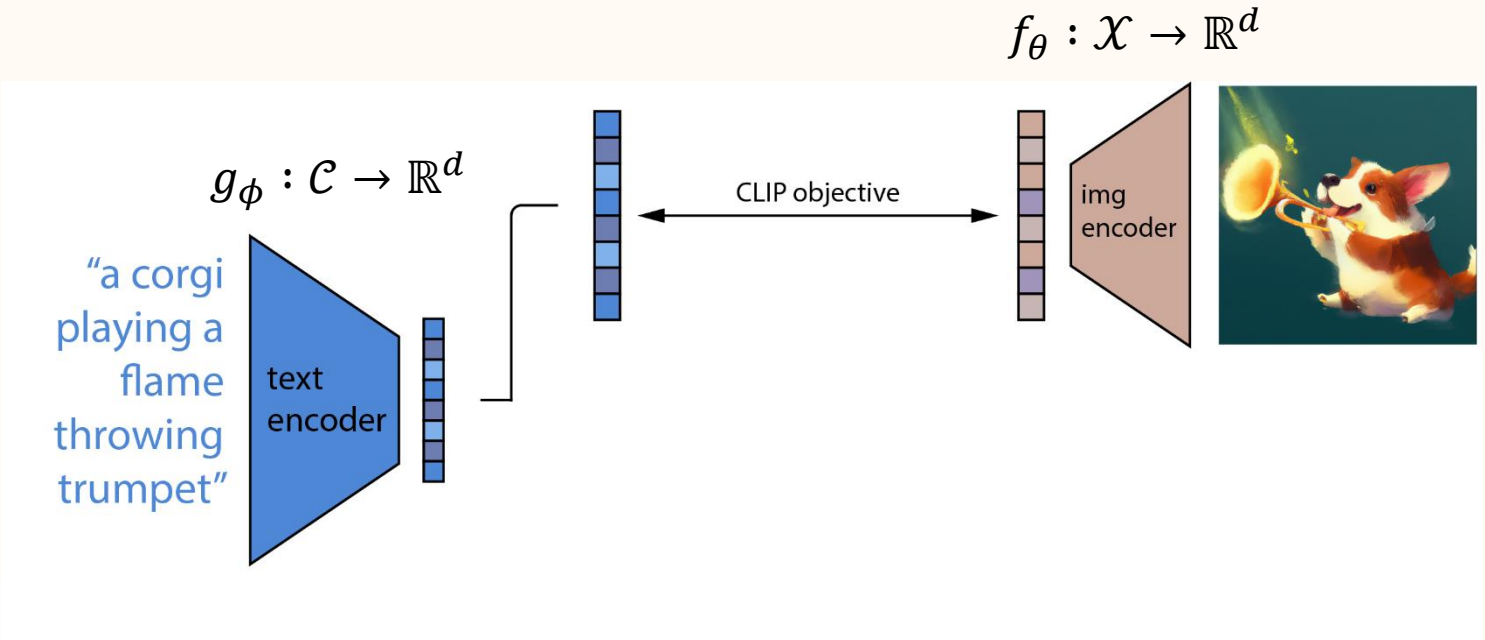
(I'm using ⇢ to denote random generation.)

A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, Hierarchical text-conditional image generation with CLIP latents, *arXiv*, 2022.

CLIP encoders f_θ and g_ϕ

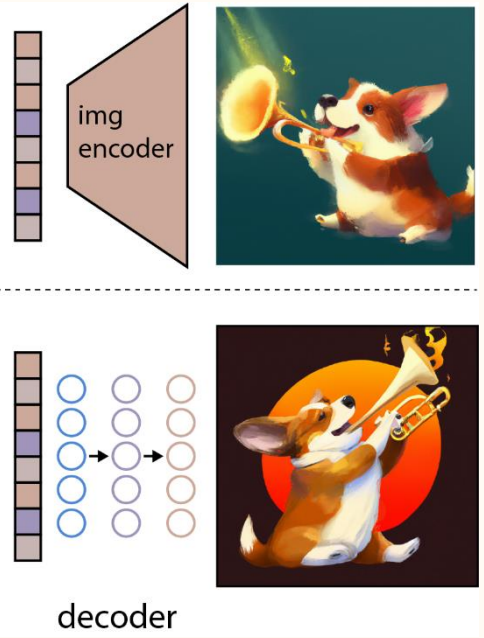
Stage 1. Train image encoder f_θ and text encoder g_ϕ as a CLIP model, or download a pre-trained CLIP model. (This CLIP model has no time dependence.)

The encoders g_ϕ and f_θ are frozen as the other networks are trained.



DALL·E 2 decoder h_ψ

$f_\theta : \mathcal{X} \to \mathbb{R}^d$



$h_\psi : \mathbb{R}^d \times \mathcal{C} \dashrightarrow \mathcal{X}$

Decoder $h_\psi(Z, C) \dashrightarrow X$ generates samples from $p(\cdot\,|f_\theta(X))$, $p(\cdot\,|f_\theta(X), C)$, or $p(\cdot\,|C)$ as a conditional diffusion model. Its use cases are:

- $h_\psi(f_\theta(X), \emptyset) \approx X$ in terms of semantic meaning.
  - Cannot and do not expect pixel-wise similarity.
  - Cool applications with the "bipartite representation" that we will see soon.
- $h_\psi(f_\theta(X), C) \approx X$ more accurately, provided that C does describe X well.
  - This will be used in final text-to-image generation.
- $h_\psi(0, C)$ generates an image corresponding to caption C.
  - Not our final text-to-image mode. (Does not work very well.)
  - Needed for classifier-free guidance.

DALL·E 2 decoder h_ψ

Stage 2: Train a conditional error (score) network

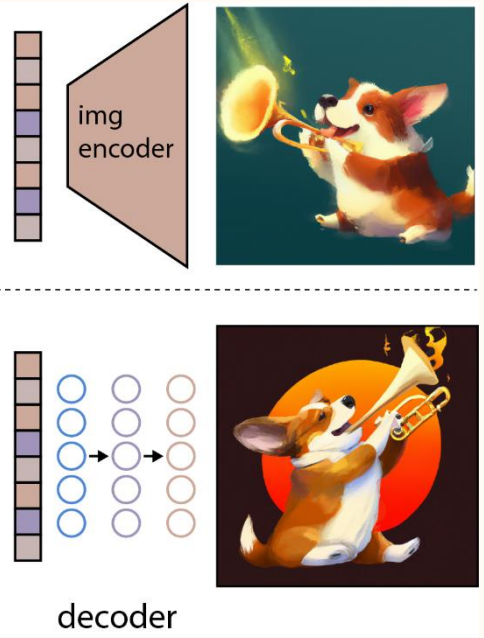
$$\varepsilon_\psi\left(X_t, t, Z^{\text{image}}, C\right)$$

with $X_0 = 0$, $Z^{\text{image}} = f_\theta(X)$, and C, given an image caption pair (X, C). Set $Z^{\text{image}} = 0$ with 10% chance and $C = \emptyset$ with 50% chance.

This is for 64×64 images, and then we have train a cascaded diffusion model

$$64 \times 64 \to 256 \times 256 \to 1024 \times 1024$$

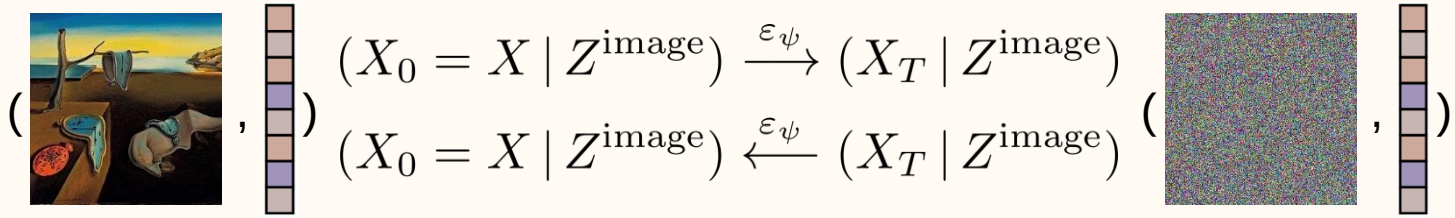
$f_\theta : \mathcal{X} \to \mathbb{R}^d$



$h_\psi : \mathbb{R}^d \times \mathcal{C} \dashrightarrow \mathcal{X}$

Bipartite representation

Consider an image X. Using $h_\psi(f_\theta(X), \emptyset)$, or equivalently $\varepsilon_\psi\left(X_t, t, Z^{\text{image}} = f_\theta(X), C = \emptyset\right)$, run the DDIM sampler forward in time to generate $\left(X_T, Z^{\text{image}}\right)$. This X_T will look like random noise, but it is a very particular noise instance as running the DDIM sampler backward in time (the usual sampling direction) starting from X_T, conditioned on Z^{image}, will generate X.

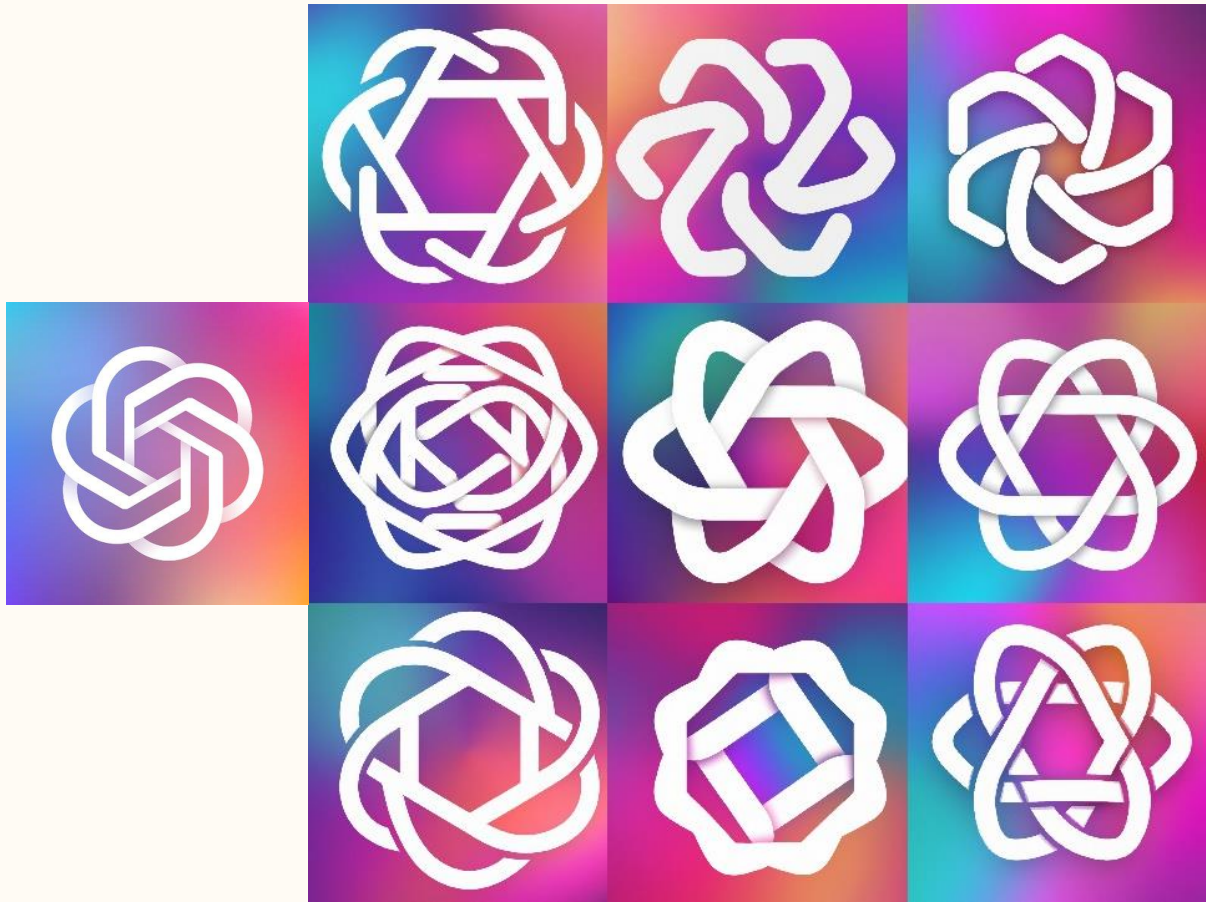


We call $\left(X_T, Z^{\text{image}}\right)$ the *bipartite representation* of X.

We can do some interesting things with it.

DALL·E 2 decoder: Variations

Given X, obtain (with DDIM sampler) its bipartite representation $\left(X_T, Z^{\text{image}}\right)$. Then sample $\left(X_0, Z^{\text{image}}\right)$ with DDPM sampler to generate variations of X. This uses f_θ and $h_\psi(Z, \emptyset)$.



DALL·E 2 decoder: Interpolations

Given $X^{(1)}$ and $X^{(2)}$, form $Z = \eta f_\theta\left(X^{(1)}\right) + (1-\eta) f_\theta\left(X^{(2)}\right)^*$ and run DDIM sampler multiple times with varying η, while fixing a particular $X_T \sim \mathcal{N}(0, I)$ to generate interpolations. This uses f_θ and $h_\psi(Z, \emptyset)$.

*Actually, the interpolation is done by spherical linear interpolation (slerp).

DALL·E 2 decoder: Text Diffs

Given (X, C) and the bipartite representation $\left(X_T, Z^{\text{image}} = f_\theta(X)\right)$, form*

$$Z = f_\theta(X) + \eta\left(g_\phi(C^{\text{new}}) - g_\phi(C)\right)$$

with $\eta > 0$, where C^{new} is a new text description. Then, run DDIM sampler multiple times with varying η to apply text diffs to the image. This uses f_θ, g_ϕ, and $h_\psi(Z, \emptyset)$.

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

*What is actually done is a spherical linear interpolation (slerp) counterpart of $Z = \eta f_\theta(C) + (1-\eta)\left(g_\phi(C^{\text{new}}) - g_\phi(C)\right)$.

DALL·E 2 decoder: Text Diffs

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

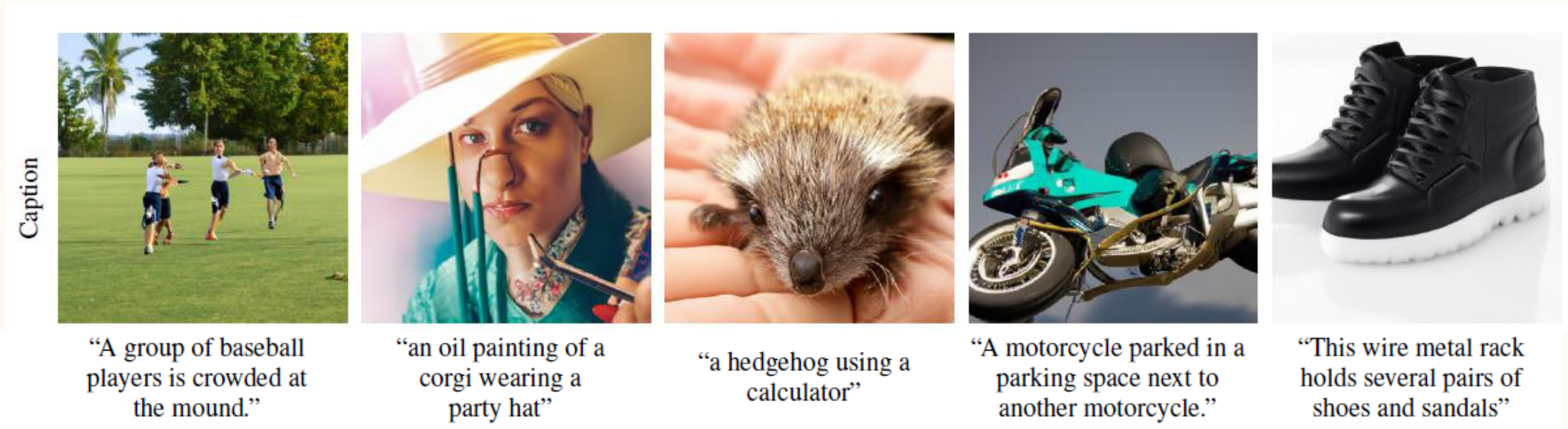
a photo of a landscape in winter → a photo of a landscape in fall

Text-to-image generation without prior

At this point, we can perform text-to-image generation given text C.

Option 1: Use $h_\psi(0, C)$.

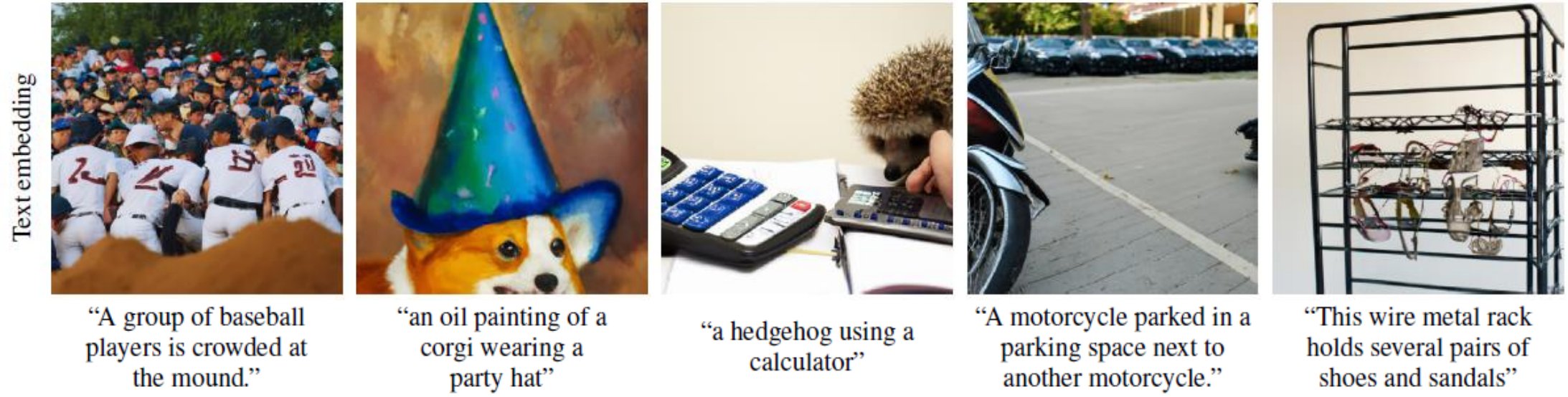
- Doesn't work very well.



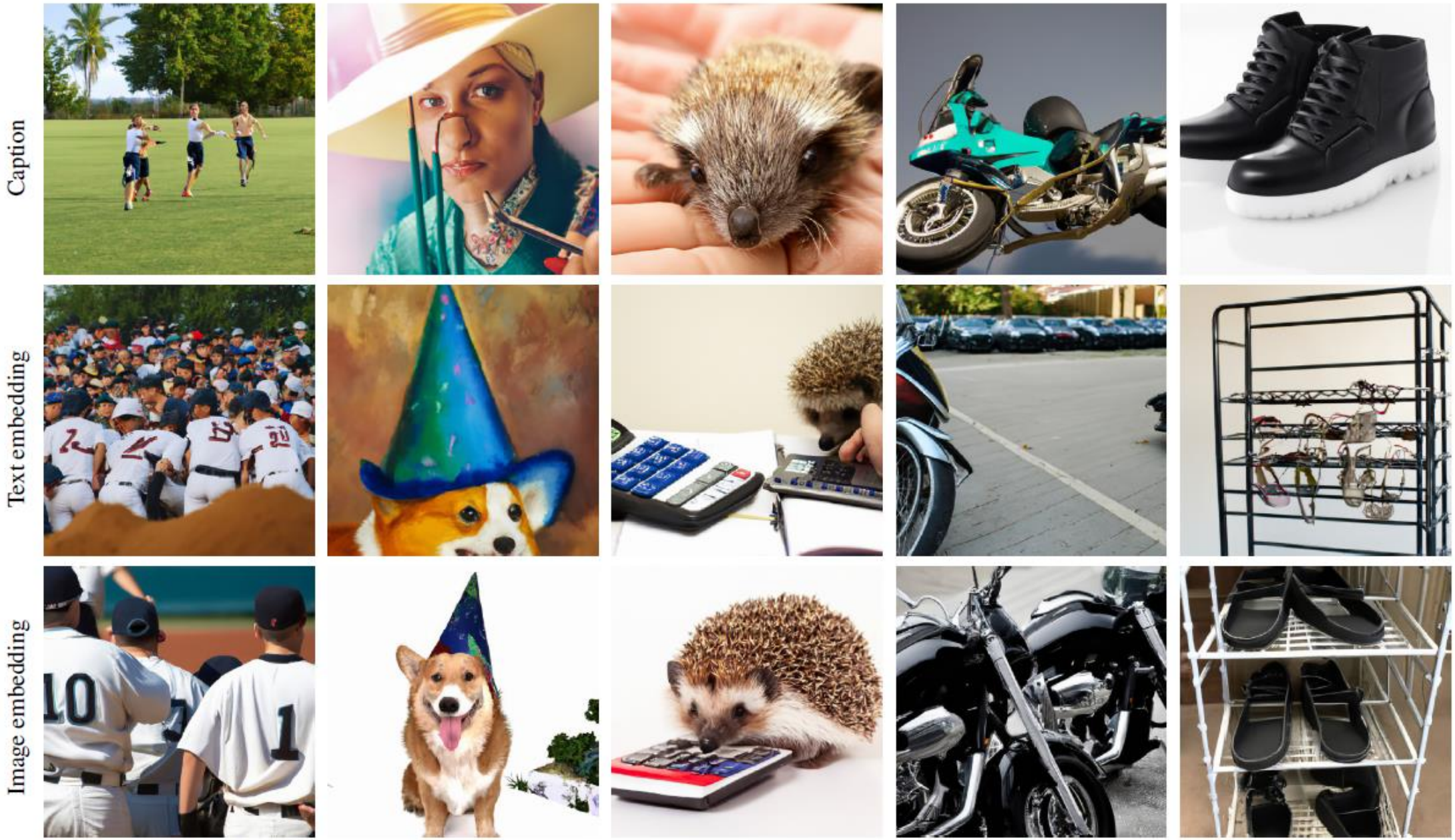
Text-to-image generation without prior

Option 2: Use $h_\psi(g_\phi(C), C)$.

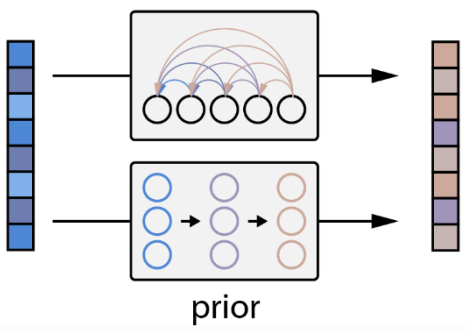
- This ignores the fact that $g_\phi(C)$ is a text embedding while h_ψ expects an image embedding $f_\theta(X)$. There is a mismatch.
- This work better, but not as good as option 3.



Option 3:
Use prior p_ω.



DALL·E 2 prior



We want a process to transform Z^{text} into Z^{image}.

$p_\omega : \mathbb{R}^d \times C \rightsquigarrow \mathbb{R}^d$

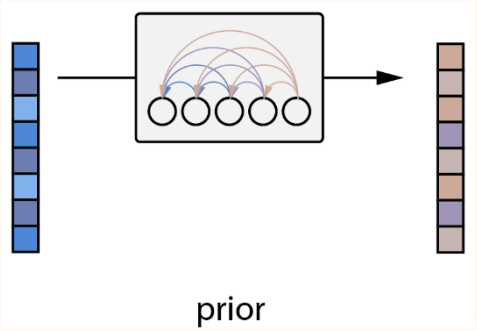
Prior $p_\omega(Z^{\text{text}}, C) \rightsquigarrow Z^{\text{image}}$ generates samples from $p\left(Z^{\text{image}} \middle| Z^{\text{text}}, C\right)$, which is mathematically equivalent to sampling X given C and obtaining $Z^{\text{image}} = f_\theta(X)$.

Note that $Z^{\text{text}} = g_\phi(C)$. So, mathematically speaking, the conditioning on Z^{text} is redundant (only conditioning on C would contain the same “information” in theory) but the CLIP-pre-trained (and frozen) features $g_\phi(C)$ are beneficial in practice.

DALL·E 2 prior

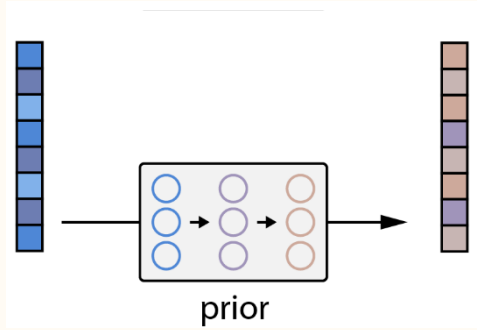
There are two approaches.

The first is an autoregressive approach, much alike pixelCNN. This doesn't work very well.



$p_\omega : \mathbb{R}^d \times C \rightsquigarrow \mathbb{R}^d$

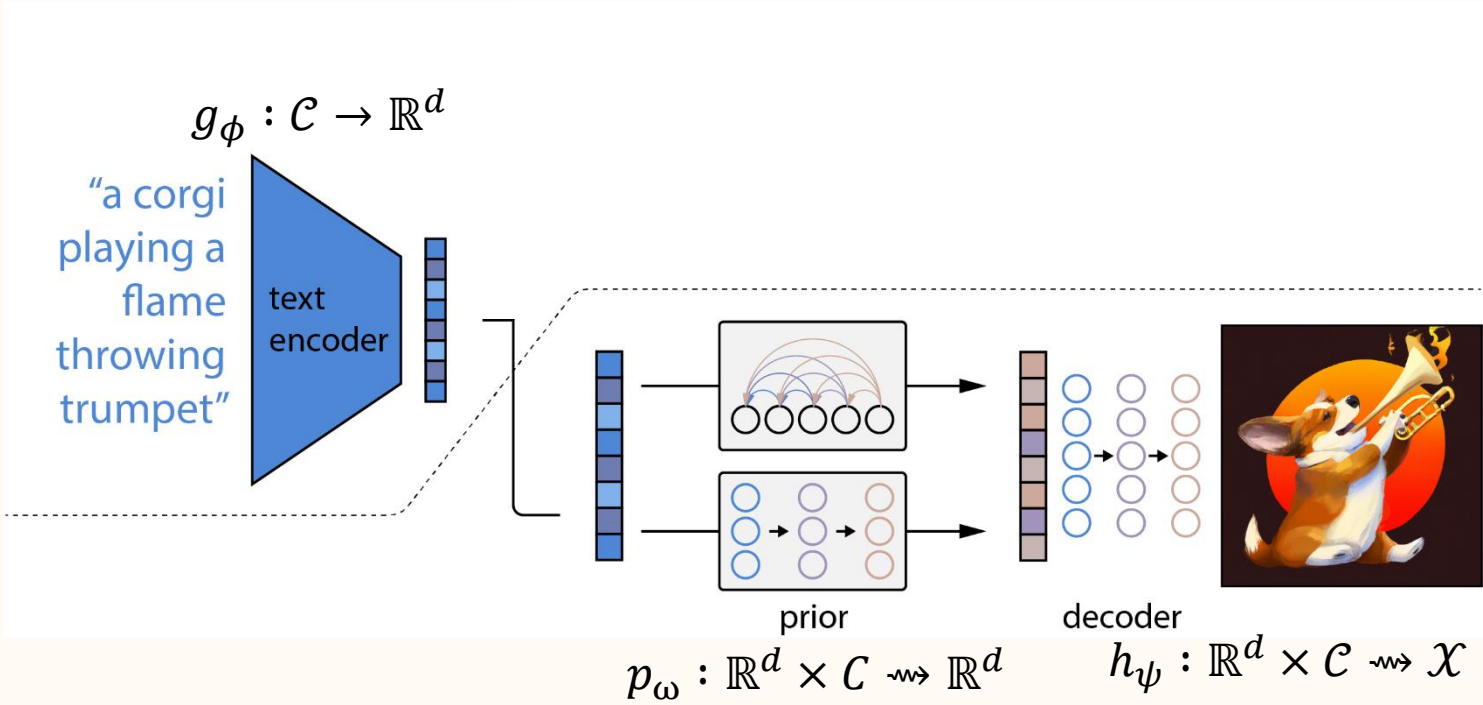
The second is based on diffusion. The architecture is a pure transformer model, not a U-Net. Since the CLIP latents are not images, the inductive biases of the convolution layers are likely not beneficial. Therefore, a U-Net is not expected to work well and it experimentally doesn't.



DALL·E 2 text-to-image generation

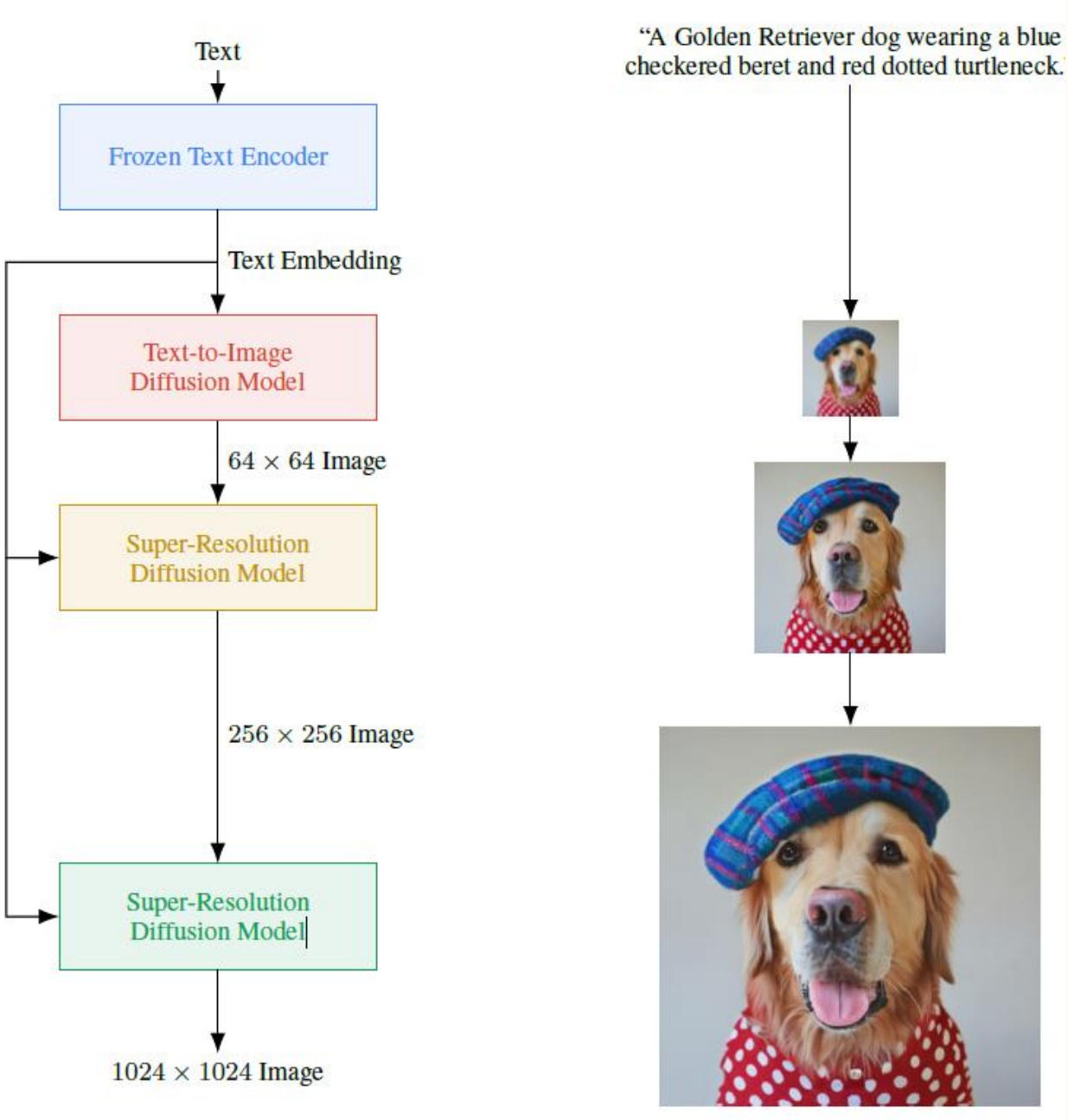
Given text C,

1. Compute $Z^{\text{text}} = g_\phi(C)$
2. Generate $p_\omega(Z^{\text{text}}, C) \rightsquigarrow Z^{\text{image}}$.
3. Generate $h_\psi\left(Z^{\text{image}}, C\right) \rightsquigarrow X$.



Imagen

Imagen is a simple cascaded diffusion model with a pre-trained large language model to encode the input text into text embeddings.



C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. Denton, S. K. S. Ghasemipour, B. K. Ayan, S. S. Mahdavi, R. G. Lopes, T. Salimans, J. Ho, D. J. Fleet, and M. Norouzi, Photorealistic text-to-Image diffusion models with deep language understanding, *NeurIPS*, 2022.

Imagen

Sprouts in the shape of text 'Imagen' coming out of a fairytale book.

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

A high contrast portrait of a very happy fuzzy panda dressed as a chef in a high end kitchen making dough. There is a painting of flowers on the wall behind him.

Teddy bears swimming at the Olympics 400m Butterfly event.

A cute corgi lives in a house made out of sushi.

A cute sloth holding a small treasure chest. A bright golden glow is coming from the chest.

A brain riding a rocketship heading towards the moon.

A dragon fruit wearing karate belt in the snow.

A strawberry mug filled with white sesame seeds. The mug is floating in a dark chocolate sea.

Imagen

A chromeplated cat sculpture placed on a Persian rug.

Android Mascot made from bamboo.

Intricate origami of a fox and a unicorn in a snowy forest.

A transparent sculpture of a duck made out of glass.

A raccoon wearing cowboy hat and black leather jacket is behind the backyard window. Rain droplets on the window.

A bucket bag made of blue suede. The bag is decorated with intricate golden paisley patterns. The handle of the bag is made of rubies and pearls.

Three spheres made of glass falling into ocean. Water is splashing. Sun is setting.

Vines in the shape of text 'Imagen' with flowers and butterflies bursting out of an old TV.

A strawberry splashing in the coffee in a mug under the starry sky.

Imagen training and generation

Stage 0. Pre-train large language model, such as Text-To-Text Transfer Transformer (T5), on plain text without any images. Once trained, freeze the large language model.

Stage 1. Train cascaded diffusion model with image-caption pairs.

Stage 2. Generate image with classifier-free guidance using dynamic thresholding.

C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, *JMLR*, 2020.

In classifier-free guidance, using a large guidance scale parameter is necessary for text-image alignment. However, this worsens the perceptual image quality (image fidelity).

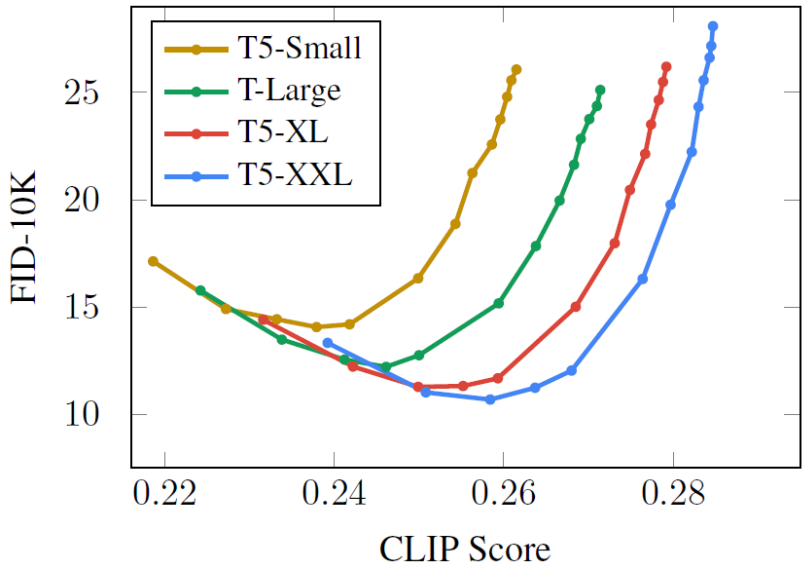
Problem is that large guidance gradients cause image to saturate. Dynamic thresholding mitigates this issue. Roughly speaking, dynamic thresholding gradually pushes (rather than clipping) the pixel values to be within the appropriate range.

(a) No thresholding. (b) Static thresholding. (c) Dynamic thresholding.

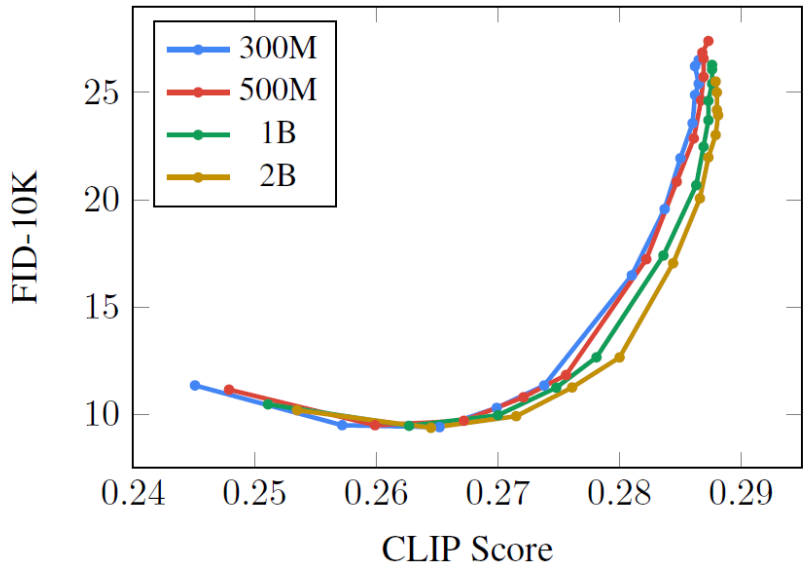
Scaling text encoder > scaling U-Net

Interesting observation:

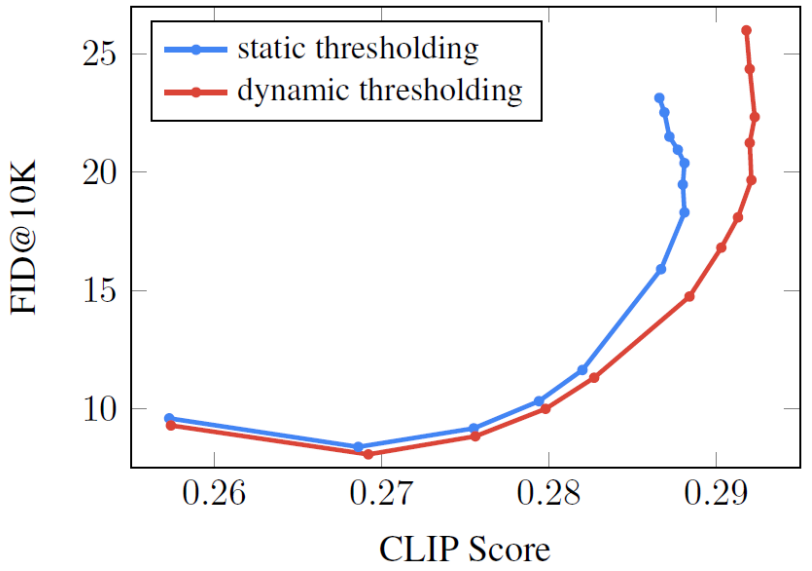
Scaling text encoder is more important than scaling error (score) network.



(a) Impact of encoder size.



(b) Impact of U-Net size.



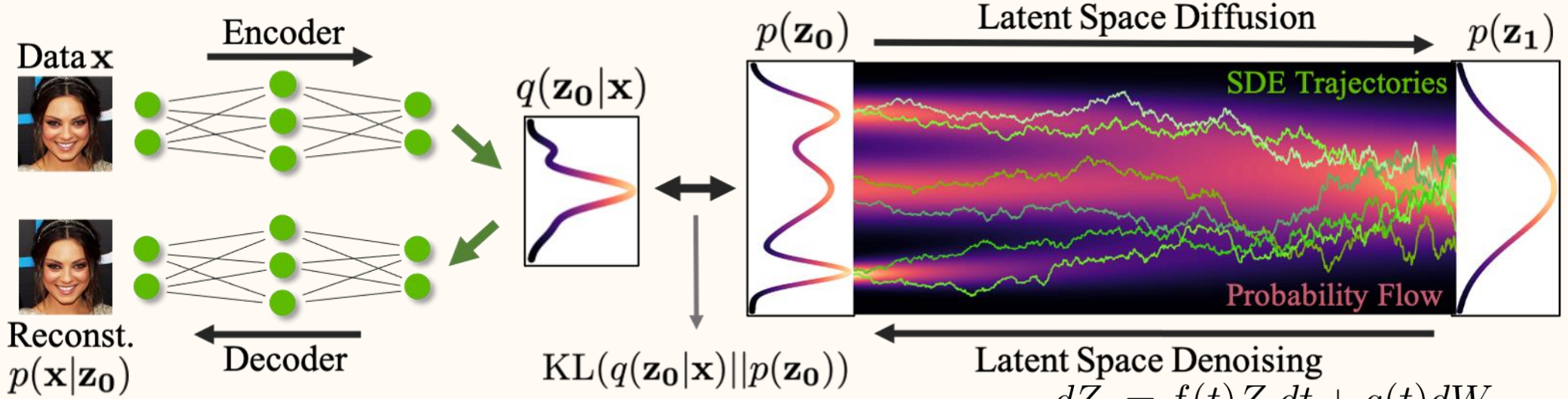
(c) Impact of thresholding.

Latent diffusion model

Standard diffusion directly operates on image.

- Perhaps inefficient to perform the many (≈ 1000) steps of diffusion on full image.
- Limits applicability. E.g. how can we diffuse to generate sentences?

Solution: Diffusion on latent variables of a variational autoencoder.

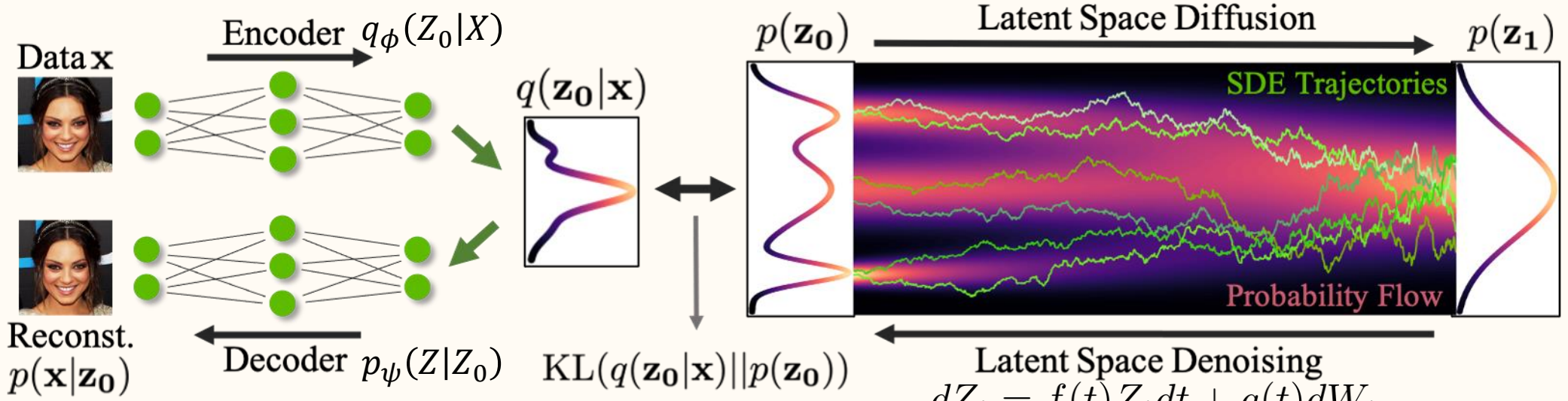


A. Vahdat, K. Kreis, and J. Kautz, Score-based generative modeling in latent space, *NeurIPS*, 2021.

Variational lower bound (VLB)

Decompose the VLB into three terms.

$$\begin{aligned}\mathrm{VLB}_{\phi,\theta,\psi}(X) &= \mathbb{E}_{Z_0\sim q_\phi(\cdot\,|\,X)}\left[-\log p_\psi(X\,|\,Z_0)\right] + D_{\mathrm{KL}}(q_\phi(\cdot\,|\,X)\|p_\theta(\cdot)) \\ &= \underbrace{\mathbb{E}_{Z_0\sim q_\phi(\cdot\,|\,X)}\left[-\log p_\psi(X\,|\,Z_0)\right]}_{\text{reconstruction term}} + \underbrace{\mathbb{E}_{Z_0\sim q_\phi(\cdot\,|\,X)}\left[\log q_\phi(Z_0\,|\,X)\right]}_{\text{negative encoder entropy}} + \underbrace{\mathbb{E}_{Z_0\sim q_\phi(\cdot\,|\,X)}\left[-\log p_\theta(Z_0)\right]}_{\text{cross-entropy}}\end{aligned}$$

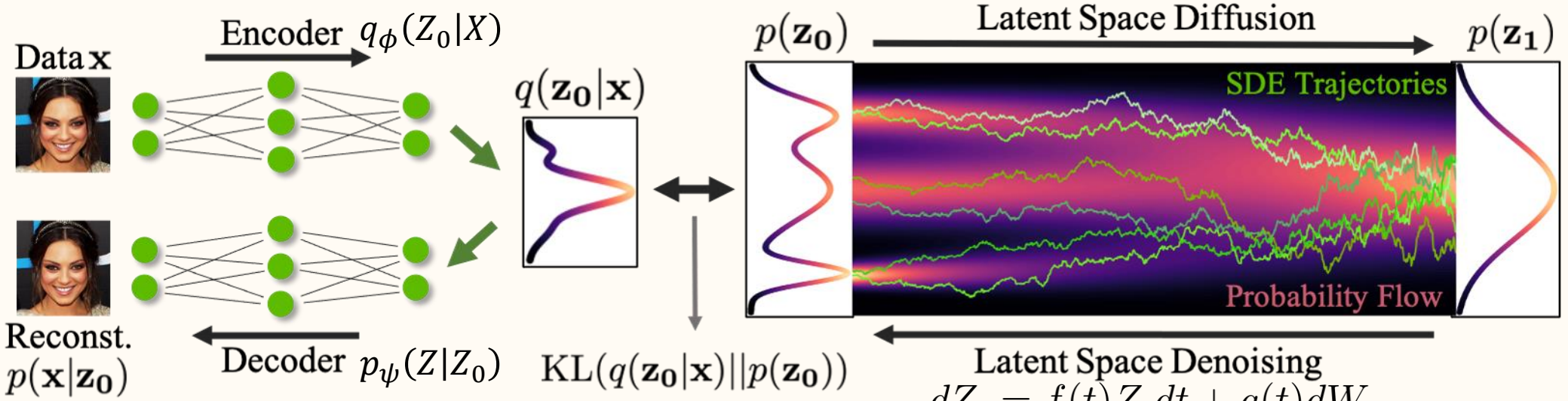


$$dZ_t = f(t)Z_t dt + g(t)dW_t$$

Prior: $p_\theta(Z_0)$ with $Z_T \sim \mathcal{N}(0, I)$

$$\begin{aligned}\mathrm{VLB}_{\phi,\theta,\psi}(X) &= \mathbb{E}_{Z_0\sim q_\phi(\cdot\,|\,X)}\left[-\log p_\psi(X\,|\,Z_0)\right] + D_{\mathrm{KL}}(q_\phi(\cdot\,|\,X)\|p_\theta(\cdot)) \\ &= \underbrace{\mathbb{E}_{Z_0\sim q_\phi(\cdot\,|\,X)}\left[-\log p_\psi(X\,|\,Z_0)\right]}_{\text{reconstruction term}} + \underbrace{\mathbb{E}_{Z_0\sim q_\phi(\cdot\,|\,X)}\left[\log q_\phi(Z_0\,|\,X)\right]}_{\text{negative encoder entropy}} + \underbrace{\mathbb{E}_{Z_0\sim q_\phi(\cdot\,|\,X)}\left[-\log p_\theta(Z_0)\right]}_{\text{cross-entropy}}\end{aligned}$$

Under the standard VAE setup, $q_\phi(\cdot\,|X) = \mathcal{N}\left(\mu_\phi(X), \Sigma_\phi(X)\right)$ and $p_\psi(\cdot\,|Z_0) = \mathcal{N}\left(f_\psi(Z), \sigma^2 I\right)$.
So sampling $Z_0 \sim q_\phi(\cdot\,|X)$ and forming the first two with the reparameterization trick and backprop is straightforward.

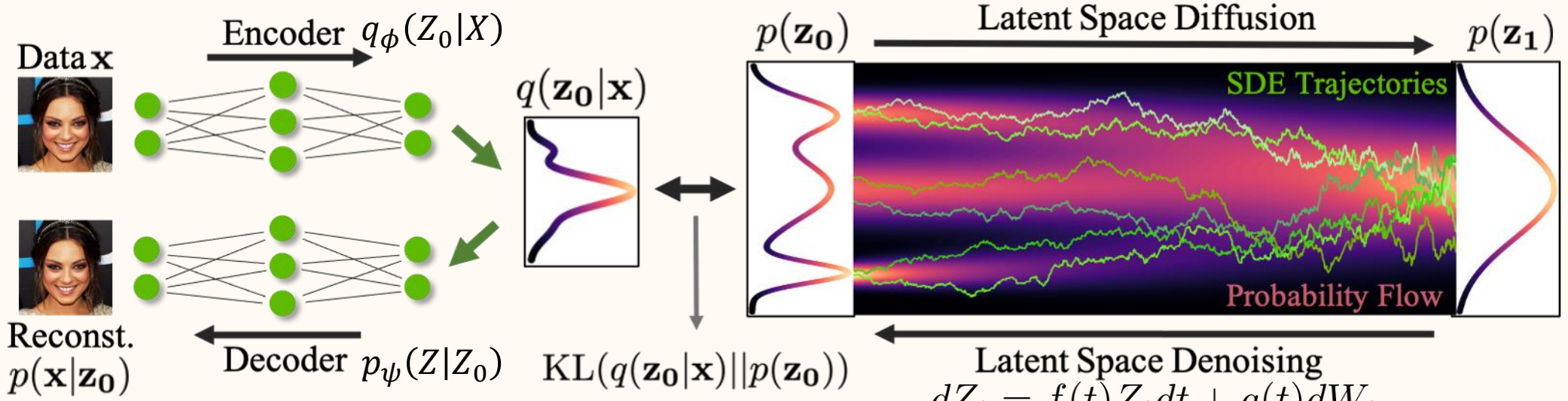


We can deal with the cross-entropy via score matching.

$$\mathrm{CE}(q_\phi(\cdot|X)\|p_\theta(\cdot)) = \mathop{\mathbb{E}}_{t\sim\mathcal{U}[0,1]}\left[\frac{w(t)}{2}\mathop{\mathbb{E}}_{\substack{Z_0\sim q_\phi(\cdot\,|\,X)\\ \varepsilon\sim\mathcal{N}(0,T)\\ Z_t=\mu_t(Z_0)+\sigma_t\varepsilon}}\left[\|\varepsilon-\varepsilon_\theta(Z_t,t)\|^2\right]\right]+\frac{d_Z}{2}\log\left(2\pi e\sigma_0\right),$$

where $\mu_t(Z_0)$ is the mean of Z_t conditioned on Z_0 under the SDE $dZ_t = f(t)Z_t dt + g(t)dW_t$.

(Need to use the reparameterization trick for $Z_0 \sim q_\phi(\cdot\,|X)$ to be able to backprop with respect to ϕ.)

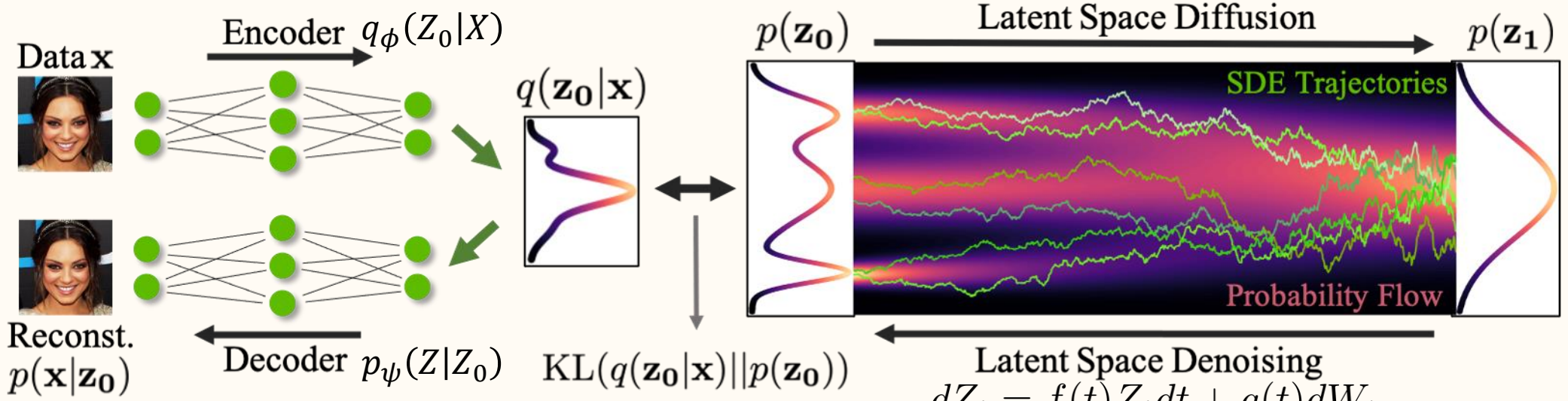


Latent diffusion model: Training

Stage 0. Pre-train VAE with prior $p_z = \mathcal{N}(0, I)$. (q_ϕ, p_ψ)

Stage 1. End-to-end train VAE with diffusion model. $(q_\phi, p_\psi, p_\theta)$

- Training only p_θ is okay, but joint training provides improvement.



Latent diffusion model: Training

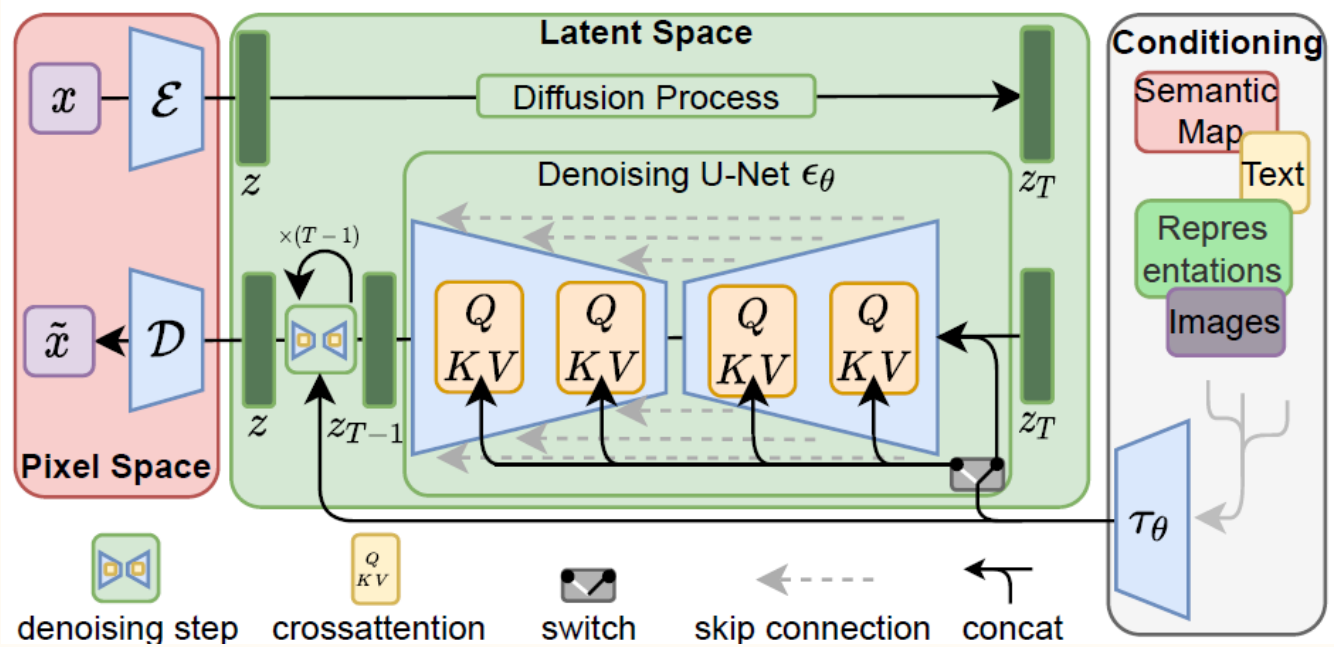
Since VAE is pretrained with $p_z = \mathcal{N}(0, I)$, the terminal marginal of the diffusion is chosen to be $p(Z_1) = \mathcal{N}(0, I)$, and if we choose the SDE $dZ_t = f(t)Z_t dt + g(t)dW_t$ to be the VP-SDE, the training of p_θ in Stage 1 should be much easier than the standard diffusion.

In standard diffusion, the distributions of X_0 and X_1 are significantly different. In this setup, the distributions of Z_0 and Z_1 are very similar.

Stable Diffusion

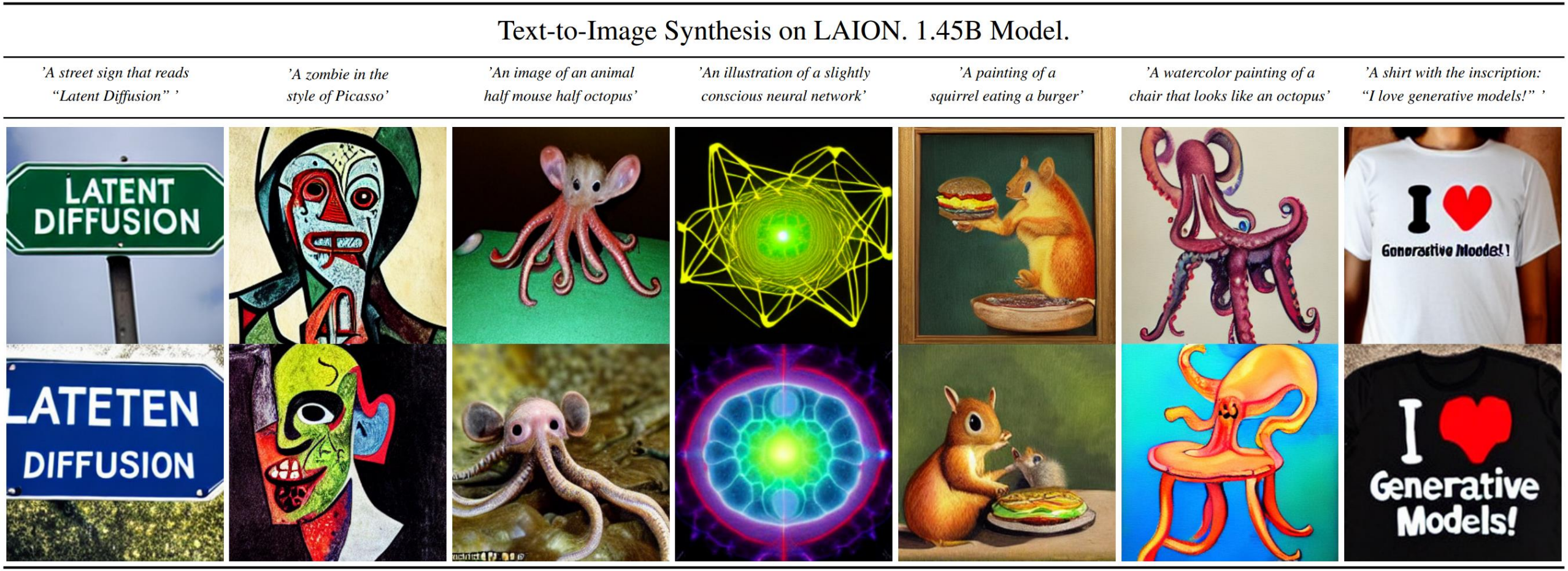
Latent diffusion model with pre-trained and frozen autoencoder.

Then conditional diffusion model trained on latent variables.



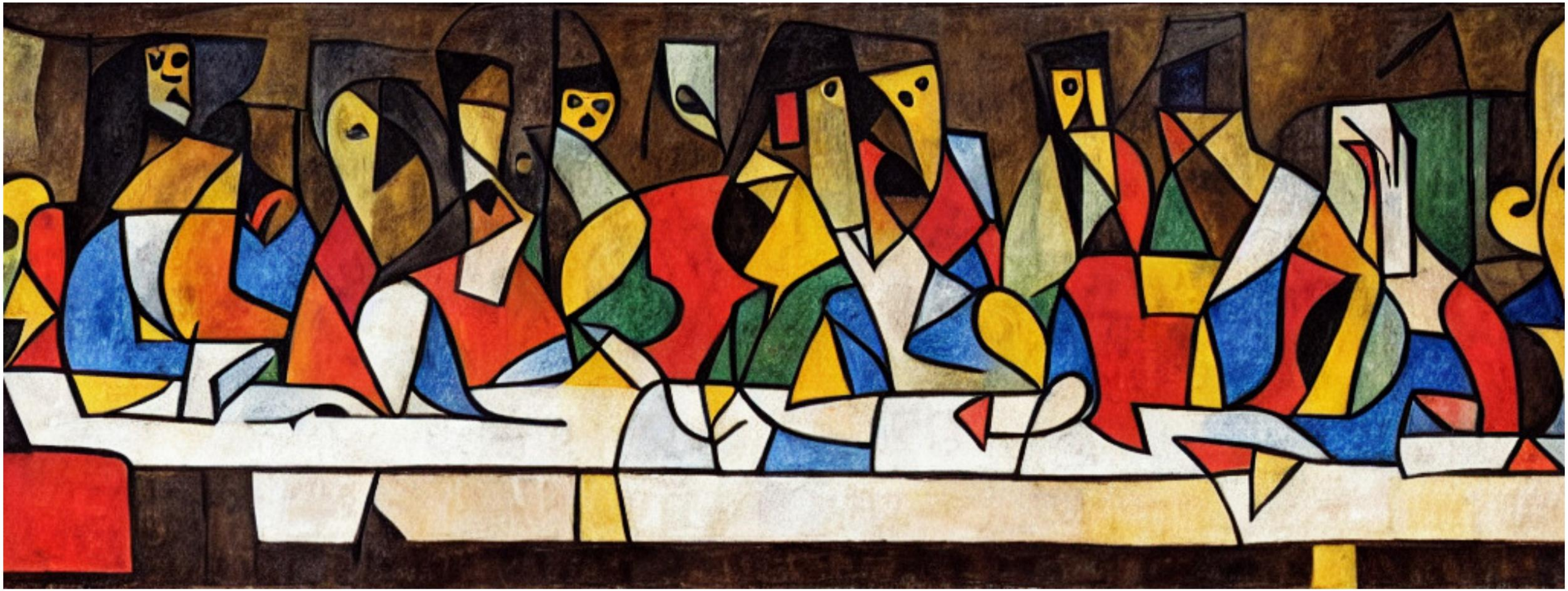
R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, High-resolution image synthesis with latent diffusion models, *CVPR*, 2022.

Stable Diffusion: Image samples



Stable Diffusion: Image samples

'A painting of the last supper by Picasso.'



Stable Diffusion: Image samples

'An oil painting of a latent space.'

'An epic painting of Gandalf the Black summoning thunder and lightning in the mountains.'

Stable Diffusion: Open source

An updated version of the model presented in the paper by Rombach et al. was released under the name *Stable Diffusion*. This has lead to many innovations.

R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, High-resolution image synthesis with latent diffusion models, *CVPR*, 2022.