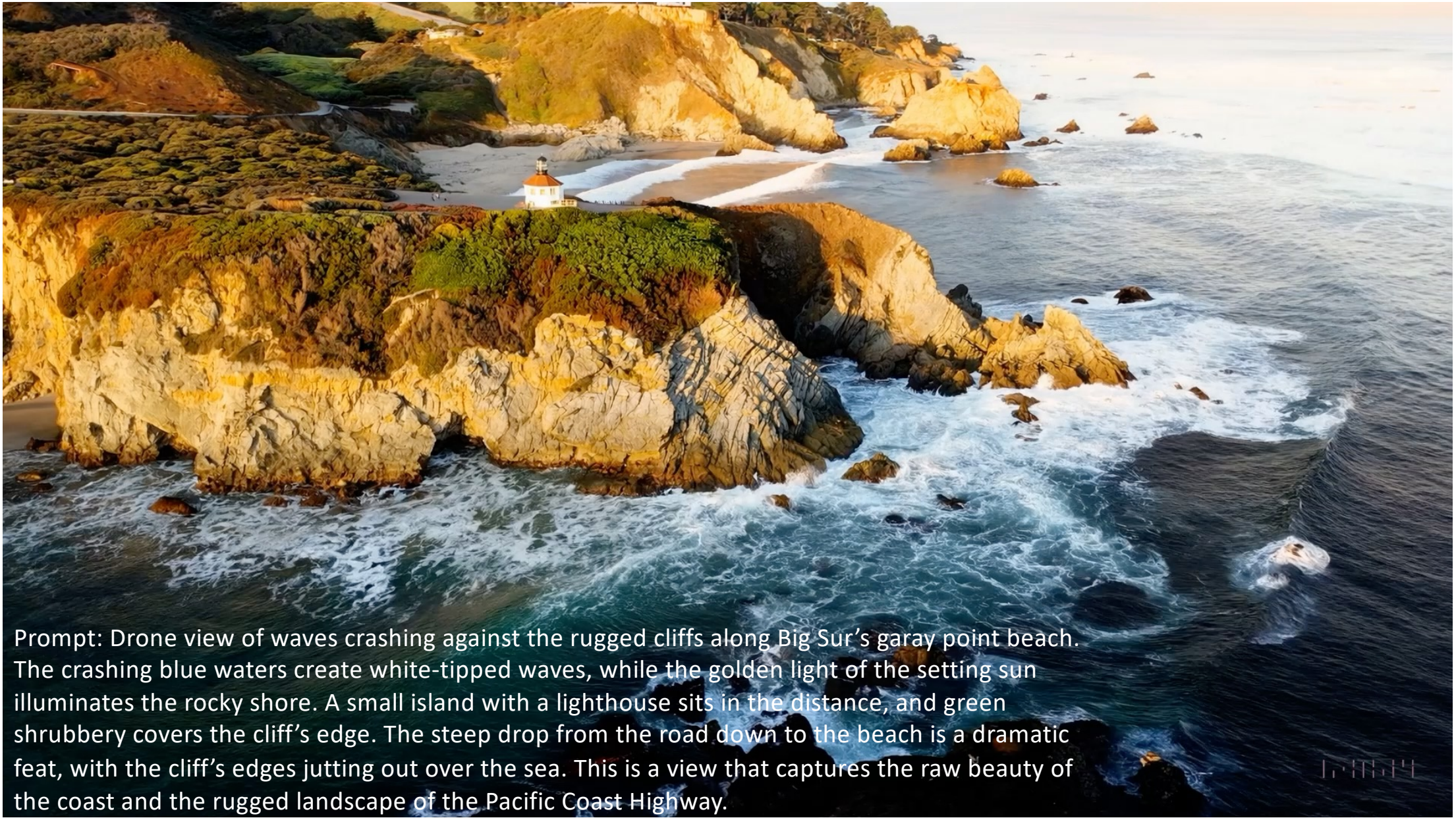
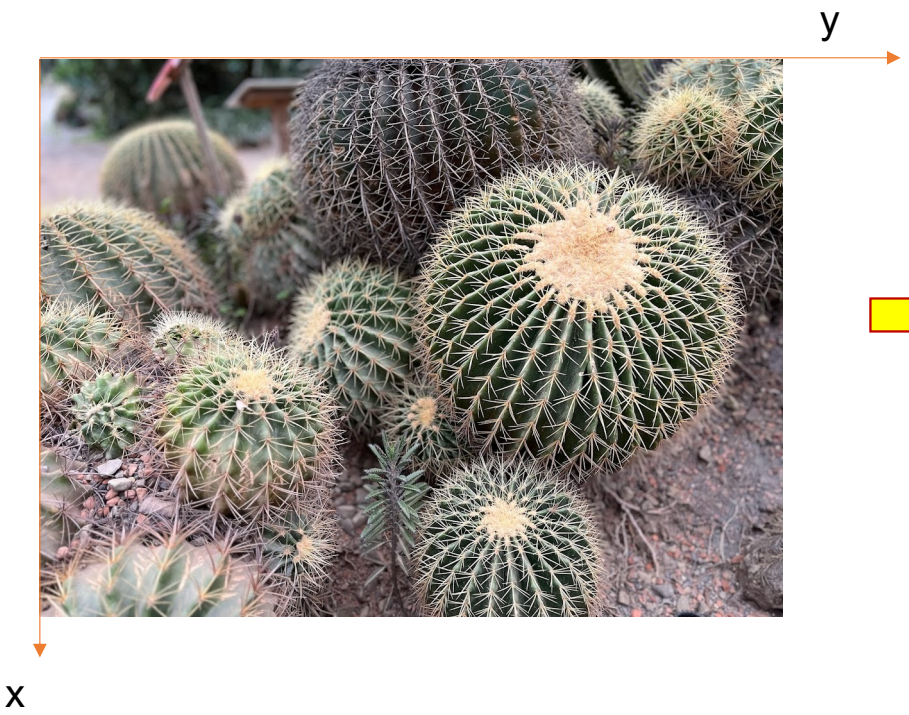


Prompt: Drone view of waves crashing against the rugged cliffs along Big Sur's garay point beach. The crashing blue waters create white-tipped waves, while the golden light of the setting sun illuminates the rocky shore. A small island with a lighthouse sits in the distance, and green shrubbery covers the cliff's edge. The steep drop from the road down to the beach is a dramatic feat, with the cliff's edges jutting out over the sea. This is a view that captures the raw beauty of the coast and the rugged landscape of the Pacific Coast Highway.

Outline

- Concept of Hidden Image Manifold
- Recent Methods for Image Generation
  - ✓ VAE (Variational Auto-Encoder)
  - ✓ GAN (Generative Adversarial Network)
  - ✓ DPM (Diffusion Probabilistic Model)

Image Data

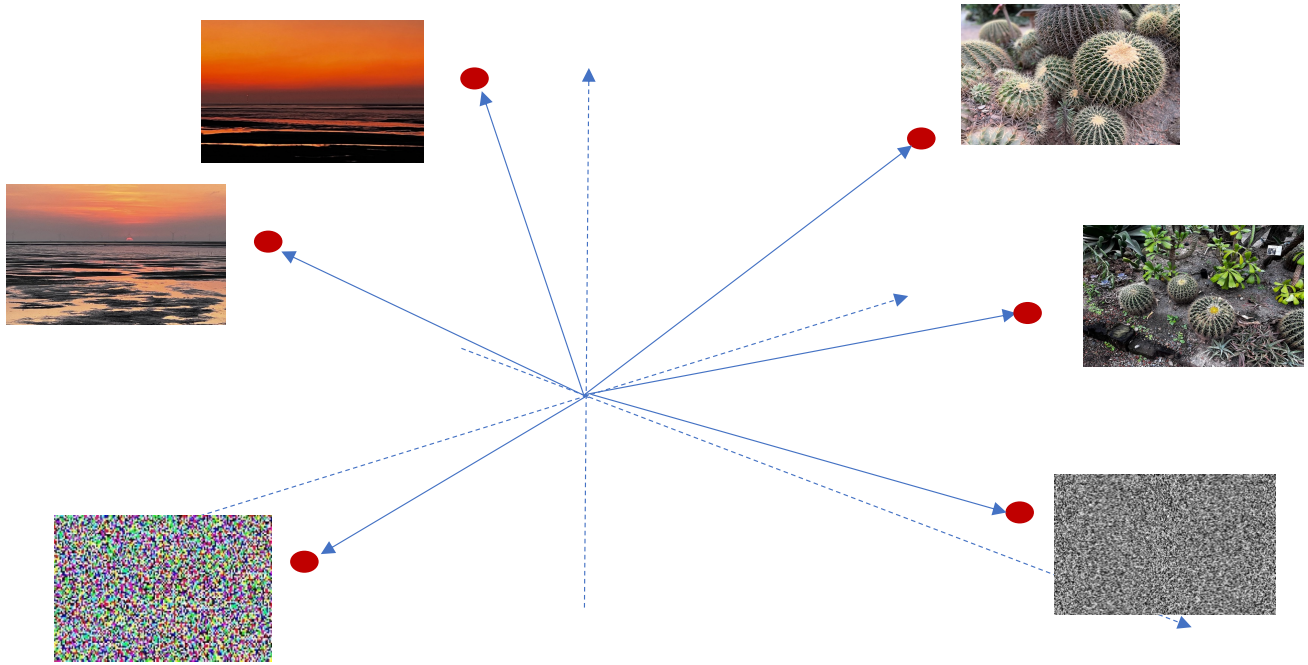


$$\begin{bmatrix} I(0,0) \\ I(0,1) \\ \vdots \\ I(0,N-1) \\ I(1,0) \\ I(1,1) \\ \vdots \\ \vdots \\ I(M-1,0) \\ I(M-1,1) \\ \vdots \\ I(M-1,N-1) \end{bmatrix} \rightarrow \begin{bmatrix} R(0,0) \\ G(0,0) \\ B(0,0) \\ R(0,1) \\ G(0,1) \\ B(0,1) \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ R(M-1,N-1) \\ G(M-1,N-1) \\ B(M-1,N-1) \end{bmatrix}$$

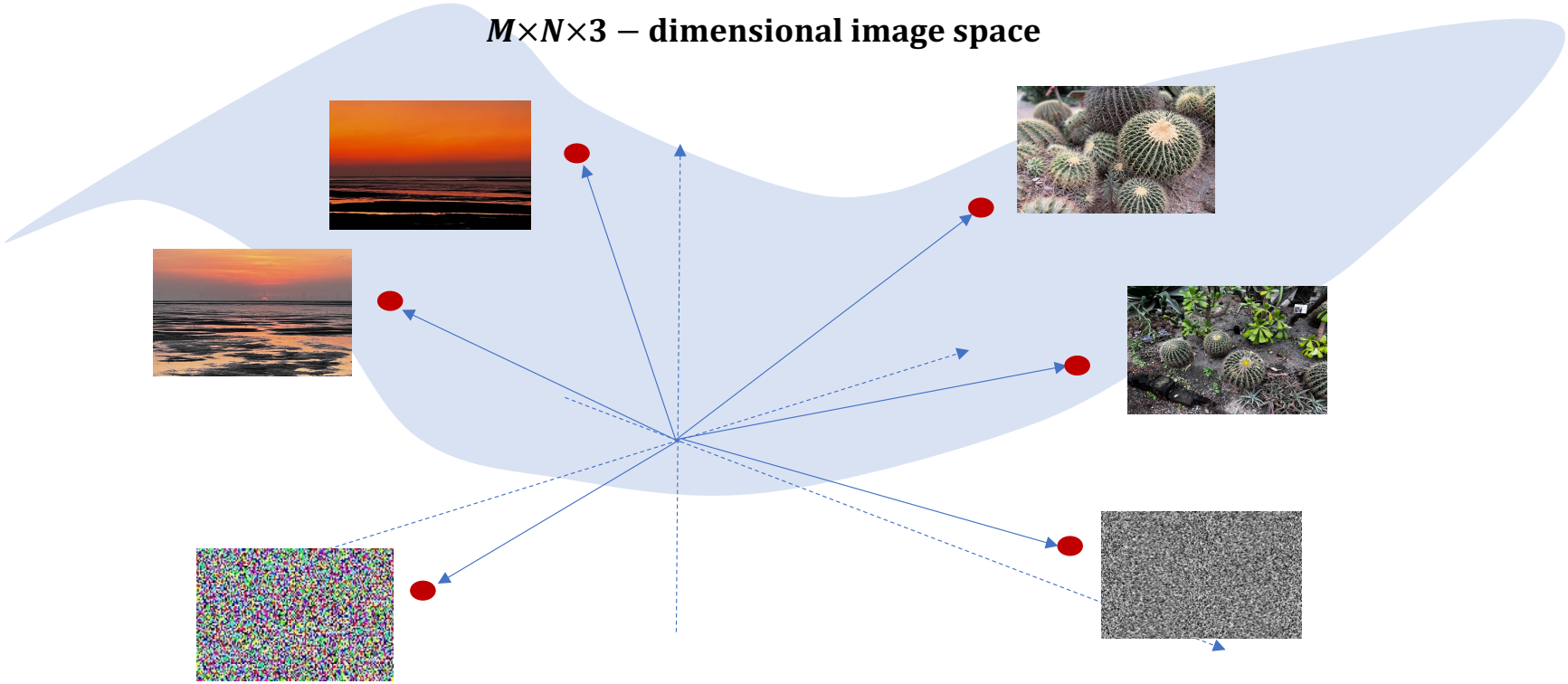
$M \times N \times 3 - dimensional$ vector

Image Data

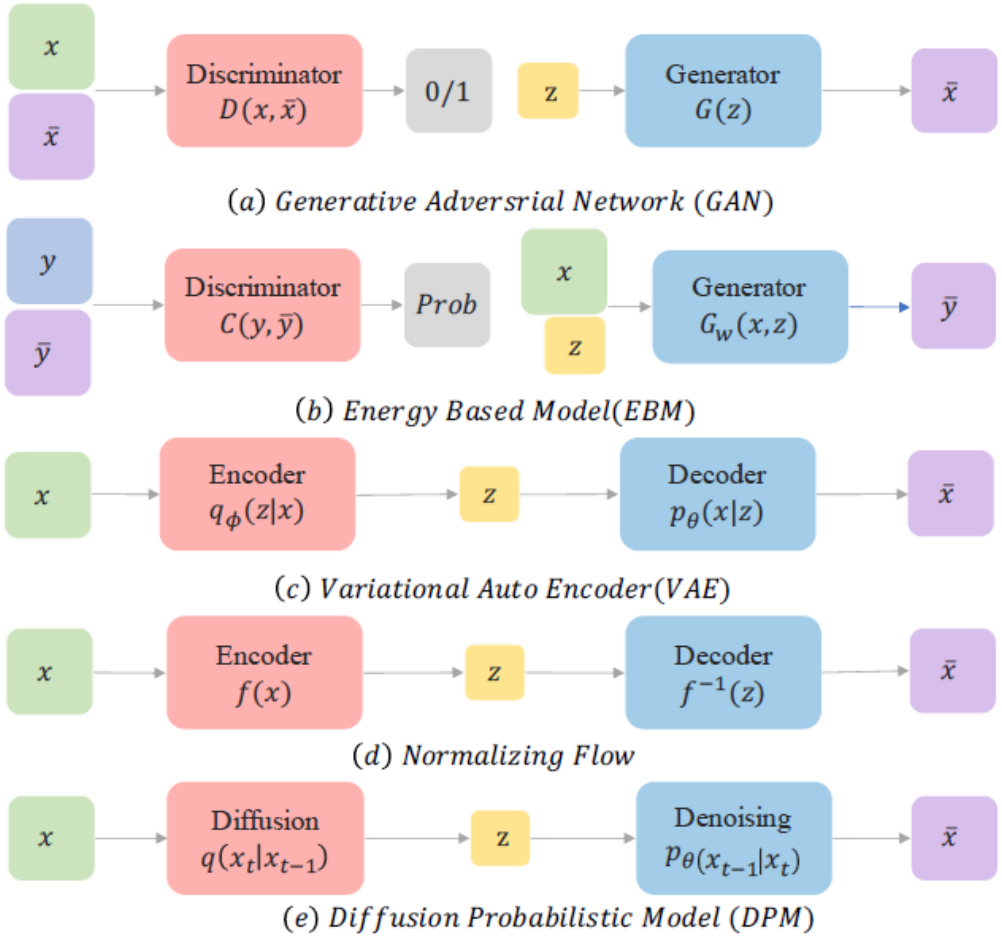
$$M \times N \times 3 - \text{dimensional image space}$$



Hidden Image Manifold

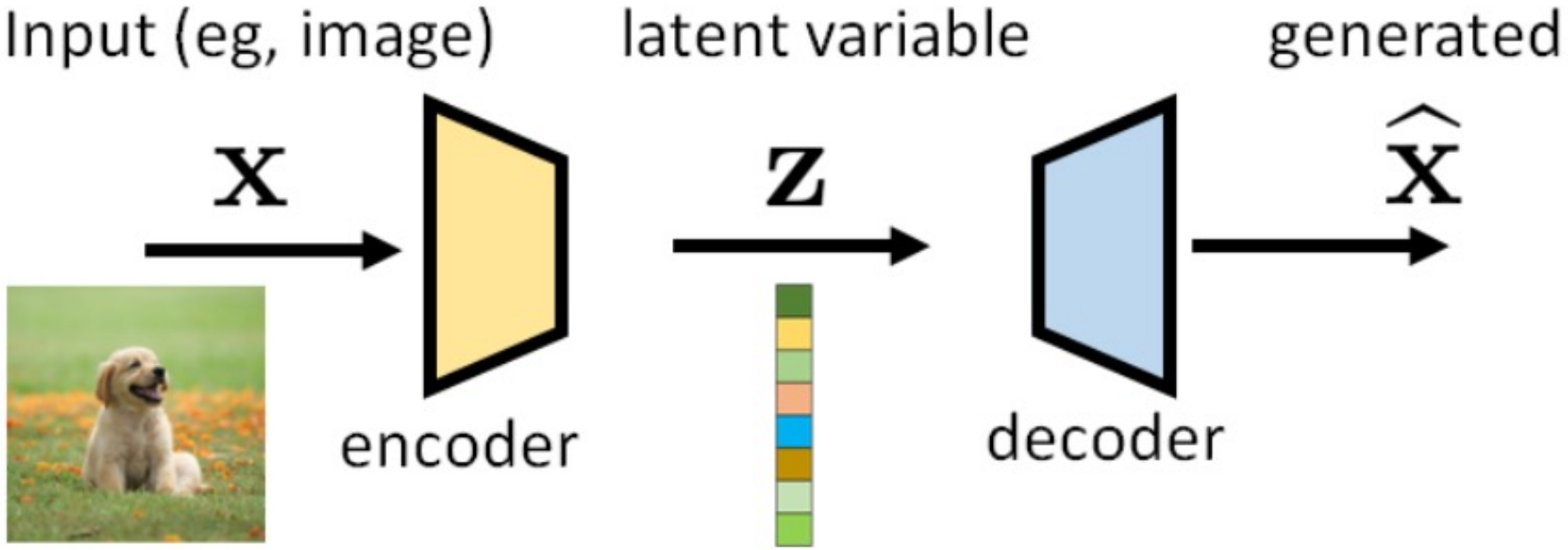


- Build a model to represent the hidden image manifold.



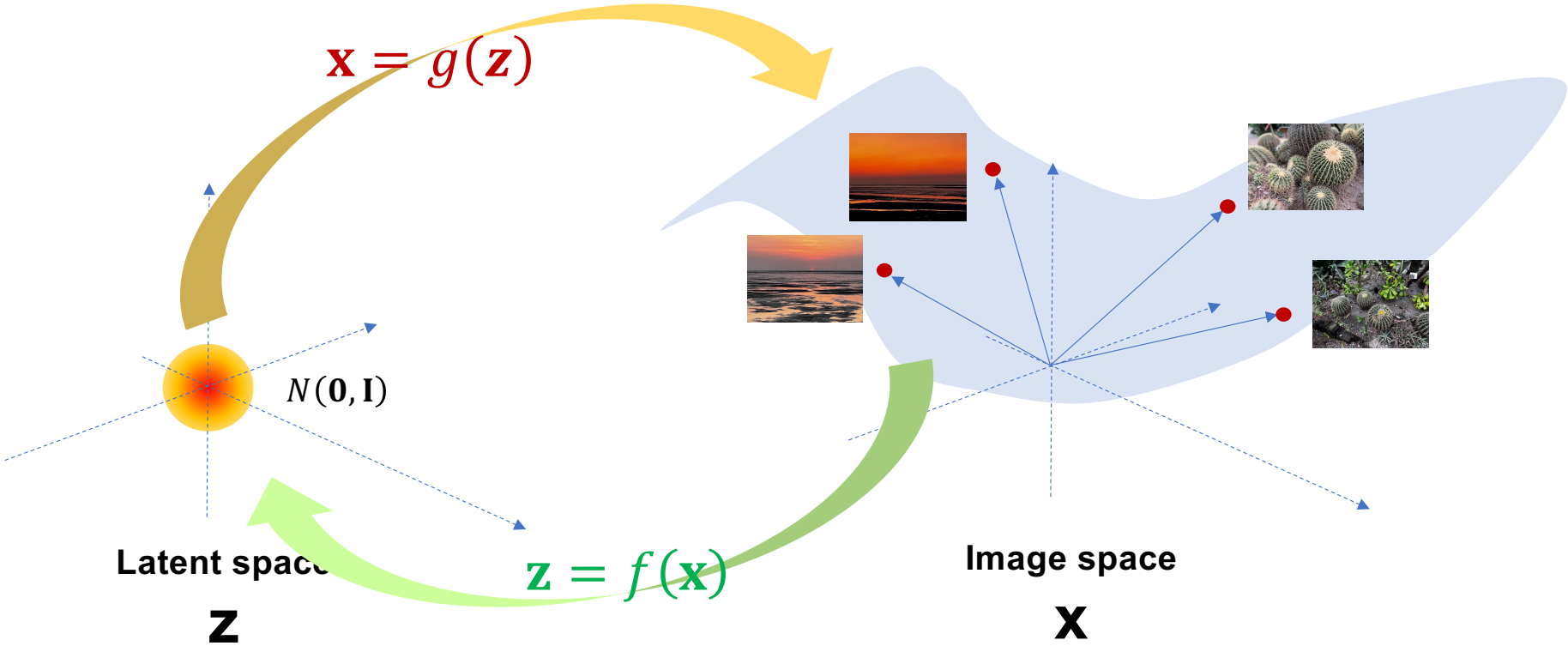
Ref: Cao, Hanqun, et al. "A survey on generative diffusion model." arXiv preprint arXiv:2209.02646 (2022).

Variational Auto-Encoder (VAE)

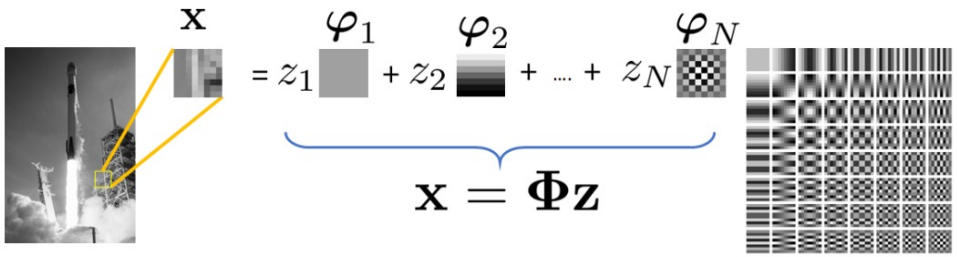


The latent variable $\mathbf{z}$ has two special roles in this setup. With respect to the input, the latent variable encapsulates the information that can be used to describe $\mathbf{x}$. The encoding procedure could be a lossy process, but our goal is to preserve the important content of $\mathbf{x}$ as much as we can. With respect to the output, the latent variable serves as the "seed" from which an image $\hat{\mathbf{x}}$ can be generated. Two different $\mathbf{z}$'s should in theory give us two different generated images.

Auto-Encoder

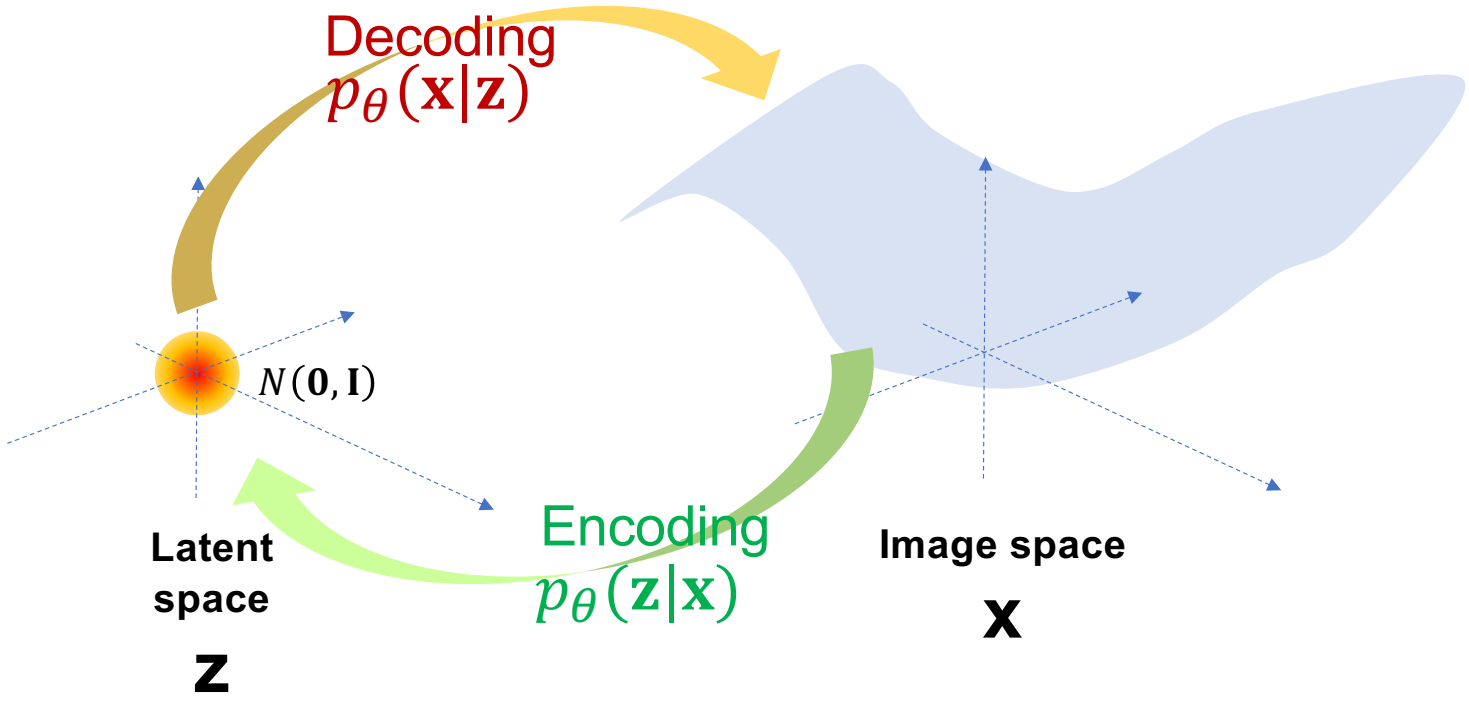


Example 1.1. Getting a latent representation of an image is not an alien thing. Back in the time of JPEG compression (which is arguably a dinosaur), we used discrete cosine transform (DCT) basis functions $\boldsymbol{\varphi}_n$ to encode the underlying image/patches of an image. The coefficient vector $\mathbf{z} = [z_1, \ldots, z_N]^T$ is obtained by projecting the image $\mathbf{x}$ onto the space spanned by the basis, via $z_n = \langle \boldsymbol{\varphi}_n, \mathbf{x} \rangle$. So, given an image $\mathbf{x}$, we can produce a coefficient vector $\mathbf{z}$. From $\mathbf{z}$, we can use the inverse transform to recover (i.e. decode) the image.



In this example, the coefficient vector $\mathbf{z}$ is the latent variable. The encoder is the DCT transform, and the decoder is the inverse DCT transform.

Variational Auto-Encoder



Variational Auto-Encoder

In VAE, we are interested in searching for the optimal probability distributions to describe **x** and **z.**

- **p(x):** The true distribution of x. It is never known.
- **p(z)**: The distribution of the latent variable. Typically, we make it a zero-mean unit-variance Gaussian **N**(0, **I**). One reason is that linear transformation of a Gaussian remains a Gaussian, and so this makes the data processing easier.
- **p(z|x)**: The conditional distribution associated with the encoder, which tells us the likelihood of z when given x. We have no access to it. The conditional distribution is not **"encoder,"** but hope **it behaves** consistently with the distribution.
- **p(x|z)**: The conditional distribution associated with the decoder, which tells us the posterior probability of getting x given z. We have no access to it.

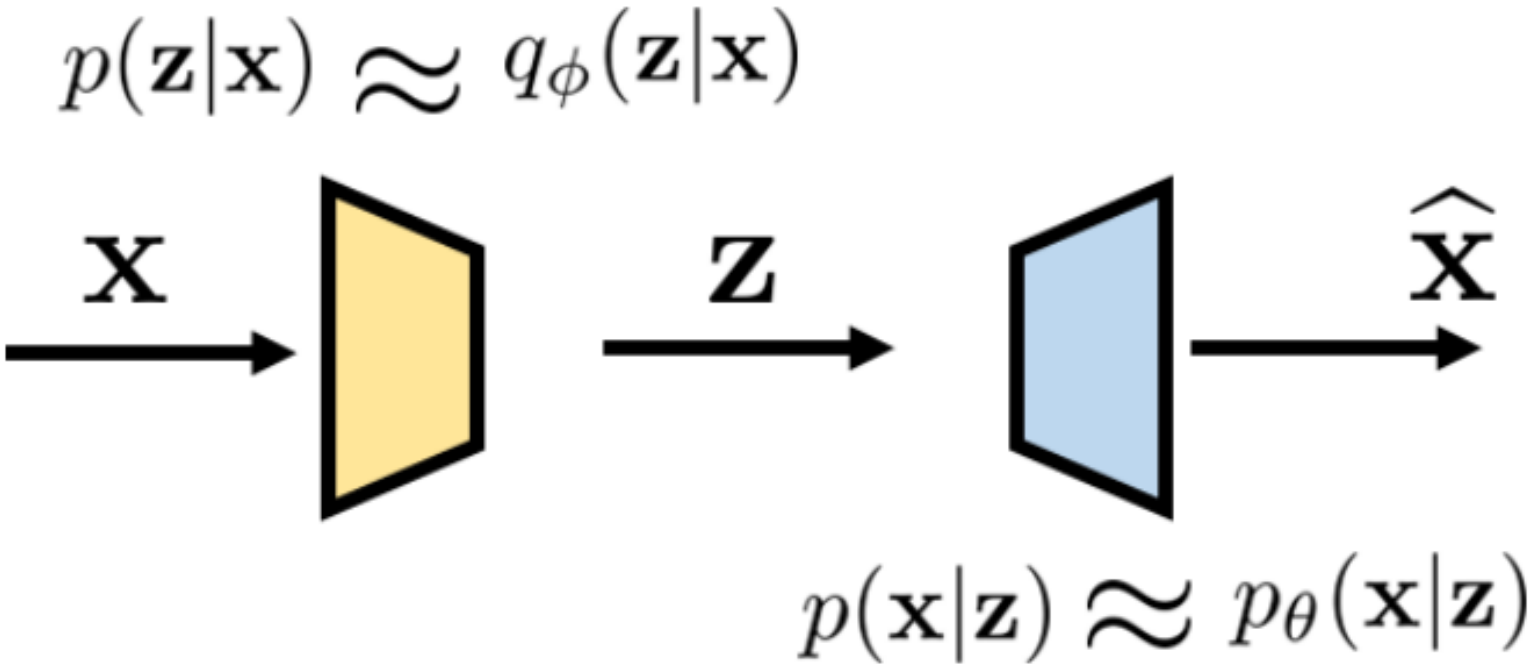
We need to impose additional structures. We consider the following two proxy distribution:

- $q_\phi(z|x)$**:** The proxy for **p(z|x)**, which is also the distribution associated with the encoder.

$$(\boldsymbol{\mu}, \boldsymbol{\sigma}^2) = \text{EncoderNetwork}_{\boldsymbol{\phi}}(\mathbf{x}),$$
$$q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z} \mid \boldsymbol{\mu}, \text{diag}(\boldsymbol{\sigma}^2)).$$

- $p_\theta(x|z)$: The proxy for **p(x|z)**, which is also the distribution associated with the decoder.

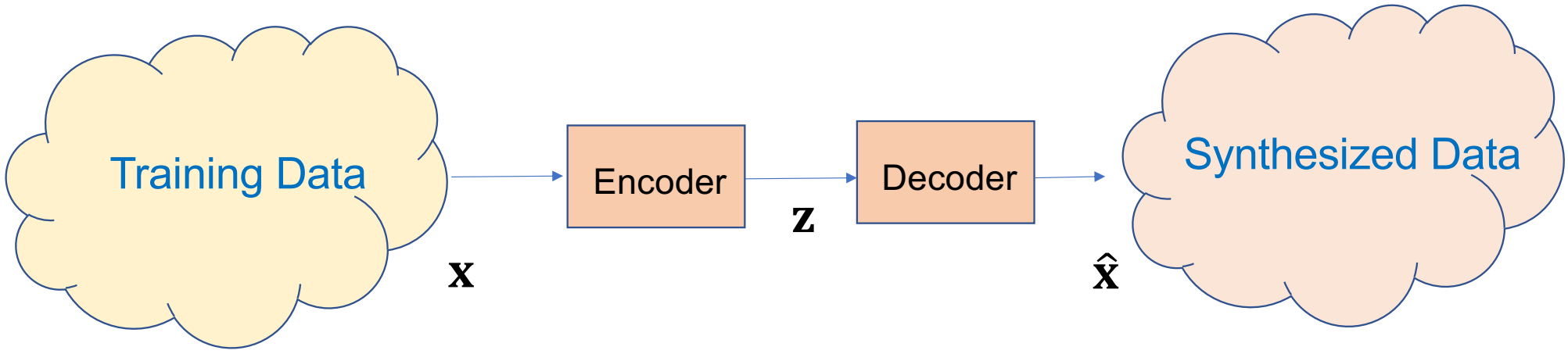
$$f_{\boldsymbol{\theta}}(\mathbf{z}) = \text{DecoderNetwork}_{\boldsymbol{\theta}}(\mathbf{z}),$$
$$p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x} \mid f_{\boldsymbol{\theta}}(\mathbf{z}), \sigma_{\text{dec}}^2 \mathbf{I}),$$



How do we use these two proxy distributions to achieve our goal of determining the encoder and the decoder?

Idea

If we treat ϕ and θ as optimization variables, then we need an objective function (or the loss function) so that we can optimize ϕ and θ through training samples.



Kullback–Leibler (KL) Divergence

In mathematical statistics, the **Kullback–Leibler (KL) divergence** (also called **relative entropy** and **I-divergence**[1]), denoted $D_{\mathrm{KL}}(P \parallel Q)$, is a type of statistical distance: a measure of how one reference probability distribution P is different from a second probability distribution Q.[2][3] Mathematically, it is defined as

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x)\ \log\left(\frac{P(x)}{Q(x)}\right).$$

Always a non-negative real number, with value 0 if and only if the two distributions in question are identical.

https://en.wikipedia.org/wiki/Kullback%E2%80%93Leibler_divergence

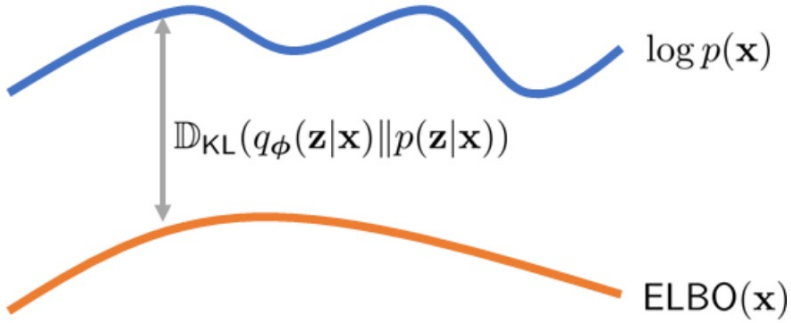
Evidence Lower Bound (ELBO)

$$\log p(\mathbf{x}) = \boxed{\text{some magical steps to be derived}}$$

$$= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] + \mathbb{D}_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p(\mathbf{z}|\mathbf{x}))$$

$$\geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right]$$

$$\stackrel{\text{def}}{=} \text{ELBO}(\mathbf{x}),$$



Proof. The trick is to use our magical proxy $q_\phi(\mathbf{z}|\mathbf{x})$ to poke around $p(\mathbf{x})$ and derive the bound.

$$\begin{aligned}
\log p(\mathbf{x}) &= \log p(\mathbf{x}) \times \underbrace{\int q_\phi(\mathbf{z}|\mathbf{x}) d\mathbf{z}}_{=1} && \text{multiply 1} \\
&= \int \underbrace{\log p(\mathbf{x})}_{\text{some constant wrt } \mathbf{z}} \times \underbrace{q_\phi(\mathbf{z}|\mathbf{x})}_{\text{distribution in } \mathbf{z}} d\mathbf{z} && \text{move } \log p(\mathbf{x}) \text{ into integral} \\
&= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x})], && (7)
\end{aligned}$$

where the last equality is the fact that $\int a \times p_Z(z) dz = \mathbb{E}[a] = a$ for any random variable Z and a scalar a.

See, we have already got $\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\cdot]$. Just a few more steps. Let's use Bayes theorem which states that $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{z}|\mathbf{x})p(\mathbf{x})$:

$$\begin{aligned}
\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x})] &= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p(\mathbf{x}, \mathbf{z})}{p(\mathbf{z}|\mathbf{x})}\right] && \text{Bayes Theorem} \\
&= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p(\mathbf{x}, \mathbf{z})}{p(\mathbf{z}|\mathbf{x})} \times \frac{q_\phi(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] && \text{Multiply and divide } q_\phi(\mathbf{z}|\mathbf{x}) \\
&= \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right]}_{\text{ELBO}} + \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{q_\phi(\mathbf{z}|\mathbf{x})}{p(\mathbf{z}|\mathbf{x})}\right]}_{\mathbb{D}_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z}|\mathbf{x}))}, && (8)
\end{aligned}$$

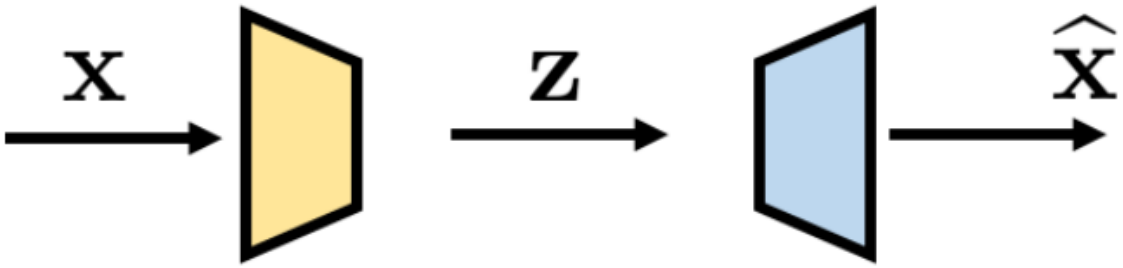
where we recognize that the first term is exactly ELBO, whereas the second term is exactly the KL divergence. Comparing Eqn (8) with Eqn (5), we complete the proof.

$$
\begin{aligned}
\text{ELBO}(\mathbf{x}) &\overset{\text{def}}{=} \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] && \text{definition} \\
&= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] && p(\mathbf{x},\mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z}) \\
&= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p(\mathbf{x}|\mathbf{z})\right] + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] && \text{split expectation} \\
&= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})\right] - \mathbb{D}_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})), && \text{definition of KL}
\end{aligned}
$$

where we replaced the inaccessible $p(\mathbf{x}|\mathbf{z})$ by its proxy $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$.

$$
\text{ELBO}(\mathbf{x}) = \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log \overbrace{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})}^{\text{a Gaussian}}]}_{\text{how good your decoder is}} - \underbrace{\mathbb{D}_{\text{KL}}\left(\overbrace{q_\phi(\mathbf{z}|\mathbf{x})}^{\text{a Gaussian}} \,\|\, \overbrace{p(\mathbf{z})}^{\text{a Gaussian}}\right)}_{\text{how good your encoder is}}.
$$

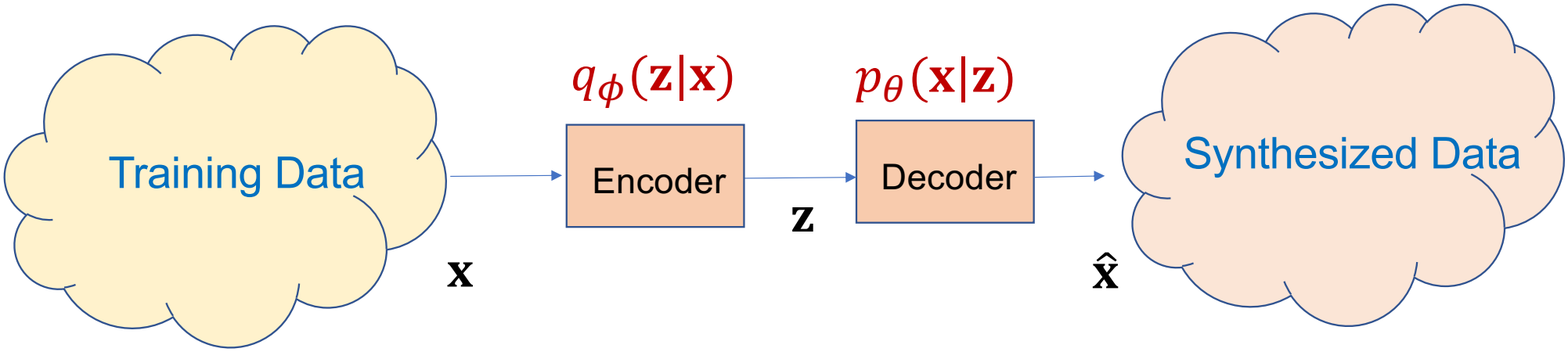
$$p(\mathbf{z}|\mathbf{x}) \approx q_\phi(\mathbf{z}|\mathbf{x})$$



$$p(\mathbf{x}|\mathbf{z}) \approx p_\theta(\mathbf{x}|\mathbf{z})$$

$$\text{ELBO}(\mathbf{x}) = \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log \overbrace{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})}^{\text{a Gaussian}}]}_{\text{how good your decoder is}} - \underbrace{\mathbb{D}_{\text{KL}}\left(\overbrace{q_\phi(\mathbf{z}|\mathbf{x})}^{\text{a Gaussian}} \,\|\, \overbrace{p(\mathbf{z})}^{\text{a Gaussian}}\right)}_{\text{how good your encoder is}}.$$

Optimization

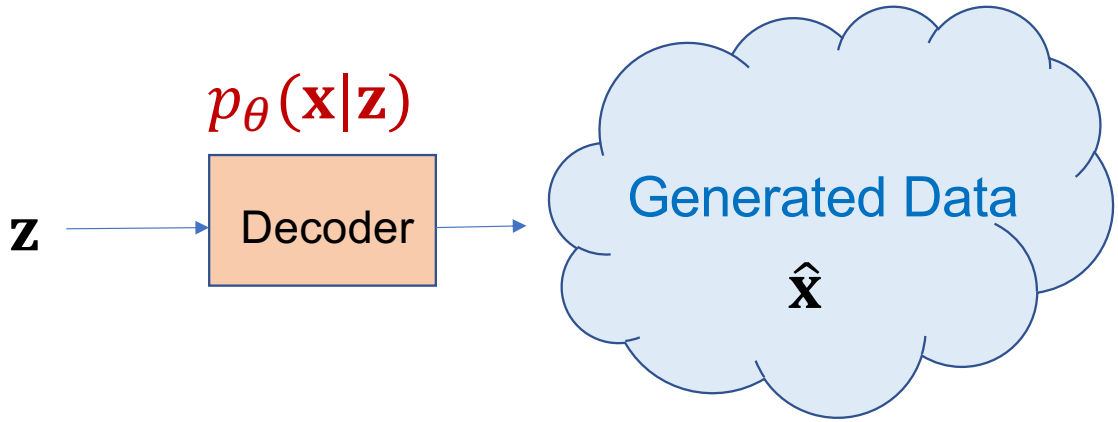


The optimization objective of VAE is to maximize the ELBO:

$$(\boldsymbol{\phi}, \boldsymbol{\theta}) = \underset{\boldsymbol{\phi},\boldsymbol{\theta}}{\operatorname{argmax}} \sum_{\mathbf{x}\in\mathcal{X}} \mathrm{ELBO}(\mathbf{x}),$$

where $\mathcal{X} = \{\mathbf{x}^{(\ell)} \mid \ell = 1, \ldots, L\}$ is the training dataset.

Image Generation



Example

$$\mathbf{x} \sim p(\mathbf{x}) = \mathcal{N}(\mathbf{x} \,|\, \boldsymbol{\mu}, \sigma^2 \mathbf{I}),$$
$$\mathbf{z} \sim p(\mathbf{z}) = \mathcal{N}(\mathbf{z} \,|\, 0, \mathbf{I}).$$

We want to build two mappings Encoder(·) and Decoder(·) using VAE.

If we knew what p(**x**) is

$$\mathbf{z} = (\mathbf{x} - \boldsymbol{\mu})/\sigma$$
$$\widehat{\mathbf{x}} = \boldsymbol{\mu} + \sigma \mathbf{z}$$

$$p(\mathbf{x}|\mathbf{z}) = \delta\left(\mathbf{x} - (\sigma\mathbf{z} + \boldsymbol{\mu})\right),$$
$$p(\mathbf{z}|\mathbf{x}) = \delta\left(\mathbf{z} - (\mathbf{x} - \boldsymbol{\mu})/\sigma\right).$$

We need to impose additional structures. We consider the following two proxy distribution:

- $q_\phi(z|x)$**:** The proxy for **p(z|x)**, which is also the distribution associated with the encoder.

$$(\boldsymbol{\mu}, \boldsymbol{\sigma}^2) = \text{EncoderNetwork}_{\boldsymbol{\phi}}(\mathbf{x}),$$
$$q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z} \mid \boldsymbol{\mu}, \text{diag}(\boldsymbol{\sigma}^2)).$$

- $p_\theta(x|z)$: The proxy for **p(x|z)**, which is also the distribution associated with the decoder.

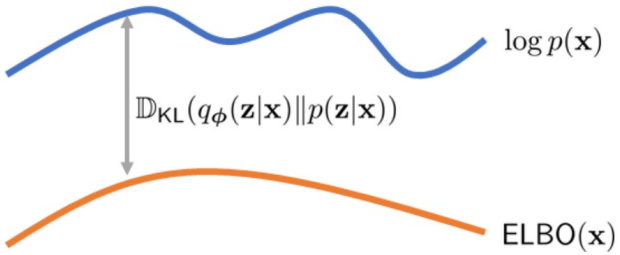
$$f_{\boldsymbol{\theta}}(\mathbf{z}) = \text{DecoderNetwork}_{\boldsymbol{\theta}}(\mathbf{z}),$$
$$p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x} \mid f_{\boldsymbol{\theta}}(\mathbf{z}), \sigma_{\text{dec}}^2 \mathbf{I}),$$

$$(\widehat{\boldsymbol{\mu}}(\mathbf{x}),\ \widehat{\sigma}(\mathbf{x})^2) = \text{Encoder}_{\boldsymbol{\phi}}(\mathbf{x}),$$
$$q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z} \mid \widehat{\boldsymbol{\mu}}(\mathbf{x}),\ \widehat{\sigma}(\mathbf{x})^2\mathbf{I}).$$
$$q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z} \mid a\mathbf{x} + \mathbf{b}, t^2\mathbf{I}).$$

$$(\widetilde{\boldsymbol{\mu}}(\mathbf{z}),\ \widetilde{\sigma}(\mathbf{z})^2) = \text{Decoder}_{\boldsymbol{\theta}}(\mathbf{z}),$$
$$p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x} \mid \widetilde{\boldsymbol{\mu}}(\mathbf{z}),\ \widetilde{\sigma}(\mathbf{z})^2\mathbf{I}).$$
$$p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{z} \mid c\mathbf{x} + \mathbf{v}, s^2\mathbf{I}).$$

Using the previous example, we can minimize the gap between log p(**x**) and ELBO(**x**) if we knew p(**z**|**x**)

$$\log p(\mathbf{x}) = \text{ELBO}(\mathbf{x}) + \mathbb{D}_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z}|\mathbf{x})) \geq \text{ELBO}(\mathbf{x})$$



$$p(\mathbf{x}|\mathbf{z}) = \delta\left(\mathbf{x} - (\sigma\mathbf{z} + \boldsymbol{\mu})\right),$$
$$p(\mathbf{z}|\mathbf{x}) = \delta\left(\mathbf{z} - (\mathbf{x} - \boldsymbol{\mu})/\sigma\right).$$

$$q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z} \mid a\mathbf{x} + \mathbf{b}, t^2\mathbf{I}).$$

$$\begin{aligned} q_\phi(\mathbf{z}|\mathbf{x}) &= \mathcal{N}(\mathbf{z} \mid \tfrac{\mathbf{x}-\boldsymbol{\mu}}{\sigma}, \ \ 0) \\ &= \delta(\mathbf{z} - \tfrac{\mathbf{x}-\boldsymbol{\mu}}{\sigma}), \end{aligned}$$

$$p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{z} \mid c\mathbf{x} + \mathbf{v}, s^2\mathbf{I}).$$

$$\begin{aligned}
\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})] &= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\log \mathcal{N}(\mathbf{x} \mid c\mathbf{z} + \mathbf{v},\ s^2\mathbf{I})] \\
&= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[-\frac{1}{2}\log 2\pi - \log s - \frac{\|\mathbf{x} - (c\mathbf{z} + \mathbf{v})\|^2}{2s^2}\right] \\
&= -\frac{1}{2}\log 2\pi - \log s - \frac{c^2}{2s^2}\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\|\mathbf{z} - \tfrac{\mathbf{x}-\mathbf{v}}{c}\|^2\right] \\
&= -\frac{1}{2}\log 2\pi - \log s - \frac{c^2}{2s^2}\mathbb{E}_{\delta\left(\mathbf{z} - \frac{\mathbf{x}-\boldsymbol{\mu}}{\sigma}\right)}\left[\|\mathbf{z} - \tfrac{\mathbf{x}-\mathbf{v}}{c}\|^2\right] \\
&= -\frac{1}{2}\log 2\pi - \log s - \frac{c^2}{2s^2}\left[\|\tfrac{\mathbf{x}-\boldsymbol{\mu}}{\sigma} - \tfrac{\mathbf{x}-\mathbf{v}}{c}\|^2\right] \\
&\leq -\frac{1}{2}\log 2\pi - \log s,
\end{aligned}$$

where the upper bound is tight if and only if the norm-square term is zero, which holds when $\mathbf{v} = \boldsymbol{\mu}$ and $c = \sigma$. For the remaining terms, it is clear that $-\log s$ is a monotonically decreasing function in s with $-\log s \to \infty$ as $s \to 0$. Therefore, when $\mathbf{v} = \boldsymbol{\mu}$ and $c = \sigma$, it follows that $\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})]$ is maximized when $s = 0$. This implies that

$$\begin{aligned}
p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) &= \mathcal{N}(\mathbf{x} \mid \sigma\mathbf{z} + \boldsymbol{\mu},\ 0) \\
&= \delta(\mathbf{x} - (\sigma\mathbf{z} + \boldsymbol{\mu})).
\end{aligned} \tag{11}$$

Limitation

$$q_{\phi}(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z} \mid \tfrac{\mathbf{x}-\boldsymbol{\mu}}{\sigma}, t^2\mathbf{I}),$$
$$p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x} \mid \sigma\mathbf{z}+\boldsymbol{\mu}, s^2\mathbf{I}).$$

$$\mathbb{D}_{\mathrm{KL}}(q_{\phi}(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})) = \mathbb{D}_{\mathrm{KL}}\left(\mathcal{N}(\mathbf{z} \mid \tfrac{\mathbf{x}-\boldsymbol{\mu}}{\sigma}, t^2\mathbf{I}) \parallel \mathcal{N}(\mathbf{z} \mid 0, \mathbf{I})\right).$$

$$q_{\phi}(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z} \mid \tfrac{\mathbf{x}-\boldsymbol{\mu}}{\sigma}, \tfrac{1}{d}\mathbf{I}).$$

where d is the dimension of **x** and **z.**

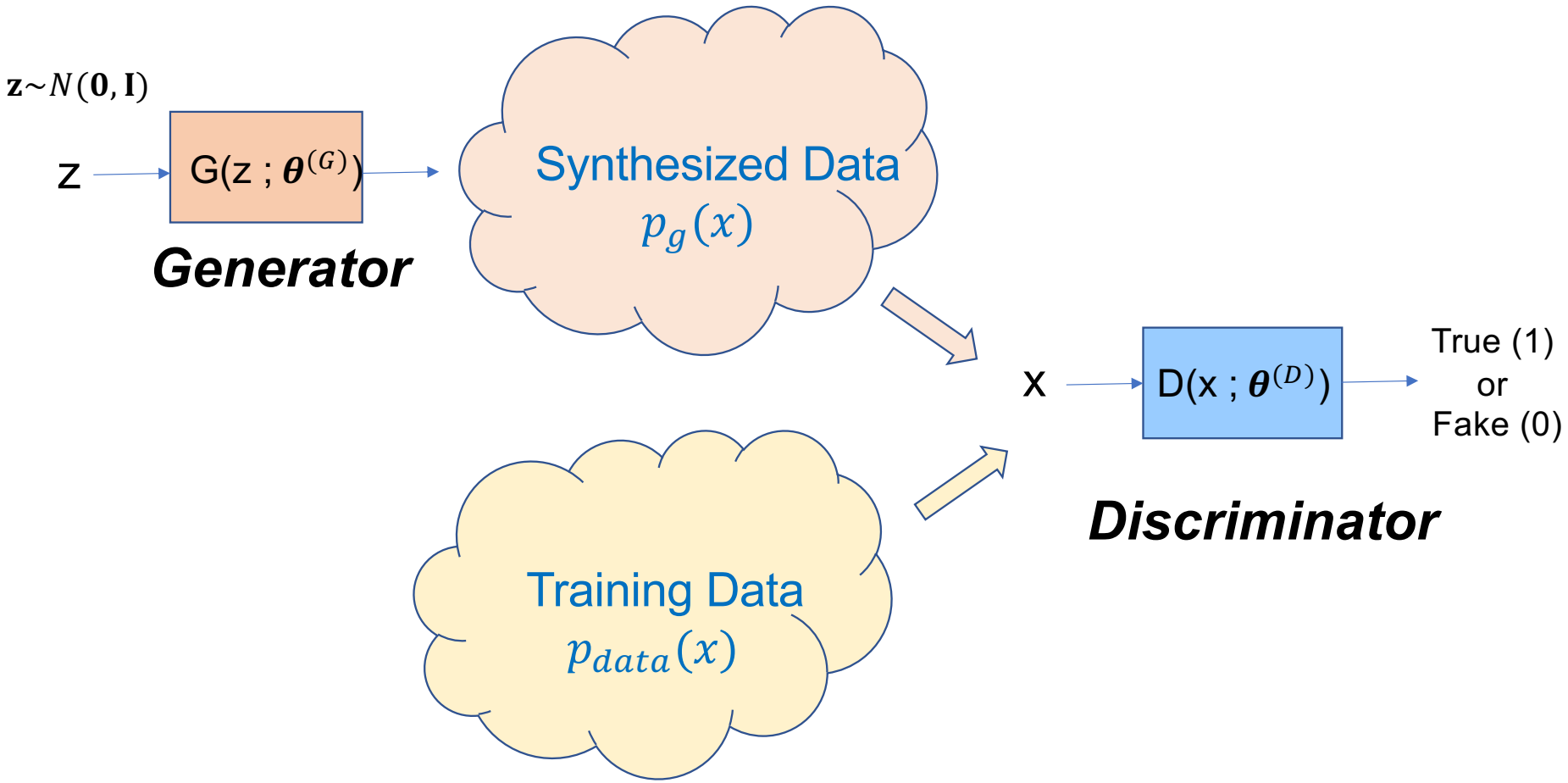
$$\mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})] = \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})}\left[\log \frac{1}{(\sqrt{2\pi s^2})^d} \exp\left\{-\frac{\|\mathbf{x}-(\sigma\mathbf{z}+\boldsymbol{\mu})\|^2}{2s^2}\right\}\right]$$

$$p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x} \mid \sigma\mathbf{z}+\boldsymbol{\mu}, \tfrac{\sigma^2}{d}\mathbf{I}).$$

While the ideal distributions are delta functions, the proxy distributions we obtain have a finite variance. This finite variance adds additional randomness to the samples generated by the VAE, because maximizing the ELBO is not the same as maximizing log p(**x**).

Generative Adversarial Networks (GAN)

- Set up a game between a **Generator** and a **Discriminator**.
  - *Generator*:
    - aim to generate samples from the same distribution of the training data.
    - trained to fool the discriminator
  - *Discriminator*:
    - Aim to distinguish training samples (real) from synthesized samples (fake).
    - based on supervised learning techniques


$\mathbf{z} \sim N(\mathbf{0}, \mathbf{I})$
z
G(z ; $\boldsymbol{\theta}^{(G)}$)
Generator
Synthesized Data
$p_g(x)$
x
D(x ; $\boldsymbol{\theta}^{(D)}$)
True (1)
or
Fake (0)
Discriminator
Training Data
$p_{data}(x)$

Generative Adversarial Network

What do you want to be?

Could we bring values to others?

YouTube LIVE

YouTube Channel：
人工智慧普適研究中心 PAIR Labs

H&J Global Chair

Registration

Time: **Nov. 28** (Thur) 10:00 AM–12:00 PM

Place: Room 329, Engineering Bldg 3, Hsinchu Campus & YouTube Live

Speaker:

Prof. Tapomayukh Bhattacharjee,
Cornell University

Title I: Towards Robot Caregiving: Building robots that work alongside human stakeholders

Abstract:

How do we build robots that can assist people with mobility limitations with activities of daily living? To successfully perform these activities, a robot needs to be able to physically interact with humans and objects in unstructured human environments. Through this talk, I will cover various projects in my lab that showcase fundamental advances in the field of physical robotic caregiving. Specifically, I will show you how we can build caregiving robots to perform activities of daily living such as bed-bathing and meal-preparation using our newly developed caregiving simulation and sensing tools as well as algorithms that leverage multimodal perception and user feedback.

Title II: Robot–assisted Feeding: How to acquire, transfer, and time bites?

Abstract:

In this talk, I will dive deep into a case study of a robot-assisted feeding system. Successful robot-assisted feeding depends on reliable bite acquisition, easy bite transfer, and appropriate bite timing. However, bite acquisition is challenging because it requires manipulation of deformable hard-to-model food items with various compliance, texture, sizes, and shapes, and thus a fixed manipulation strategy may not work. Bite transfer is not trivial because it constitutes a unique type of robot-human handover where the human needs to use the mouth which entails that the transfer should be efficient and safe even under unstructured and partially-observable physical interactions. Also, the dynamics of bite transfer changes during group dining activity and inferring the correct time to transfer a bite is a challenge. This talk will focus on algorithms and technologies used to address these issues of bite acquisition, bite transfer, and bite timing, and how we deployed these systems.

Bio:

Tapomayukh "Tapo" Bhattacharjee is an Assistant Professor in the Department of Computer Science at Cornell University where he directs the EmPRISE Lab (https://emprise.cs.cornell.edu/). He completed his Ph.D. in Robotics from Georgia Institute of Technology and was an NIH Ruth L. Kirschstein NRSA postdoctoral research associate in Computer Science & Engineering at the University of Washington. He wants to enable robots to assist people with mobility limitations with activities of daily living. His work spans the fields of human-robot interaction, haptic perception, and robot manipulation and focuses on addressing the fundamental research question on how to leverage robot-world physical interactions in unstructured human environments to perform relevant activities of daily living. He is the recipient of TRI Young Faculty Researcher Award'24, NSF CAREER Award'23, and his work has won Best Paper Award Finalist at HRI'24, Best Demo Award at HRI'24, Best RoboCup Paper Award at IROS'22, Best Paper Award Finalist and Best Student Paper Award Finalist at IROS'22, Best Technical Advances Paper Award at HRI'19, and Best Demonstration Award at NeurIPS'18. His work has also been featured in many media outlets including the BBC, Reuters, New York Times, IEEE Spectrum, and GeekWire and his robot-assisted feeding work was selected to be one of the best interactive designs of 2019 by Fast Company.

主辦單位 H&J Global Chair、UMN–NYCU Joint AI Lab、NYCU AI College、人工智慧普適研究中心

活動聯絡人 楊小姐，snail.yaching@gmail.com

Charlie Kemp

Charles C. Kemp's personal website

Company Lab Classes ▾ Blog CV

I am the chief technology officer for Hello Robot Inc., which I cofounded with Aaron Edsinger in 2017. I earned my PhD at MIT with Rod Brooks as my advisor. Prior to working full-time at Hello Robot, I was an academic. I've had the pleasure of advising outstanding PhD students, contributing to published research, writing code, and teaching. I've also invented a few robots. I am a roboticist.

Robots with Purpose

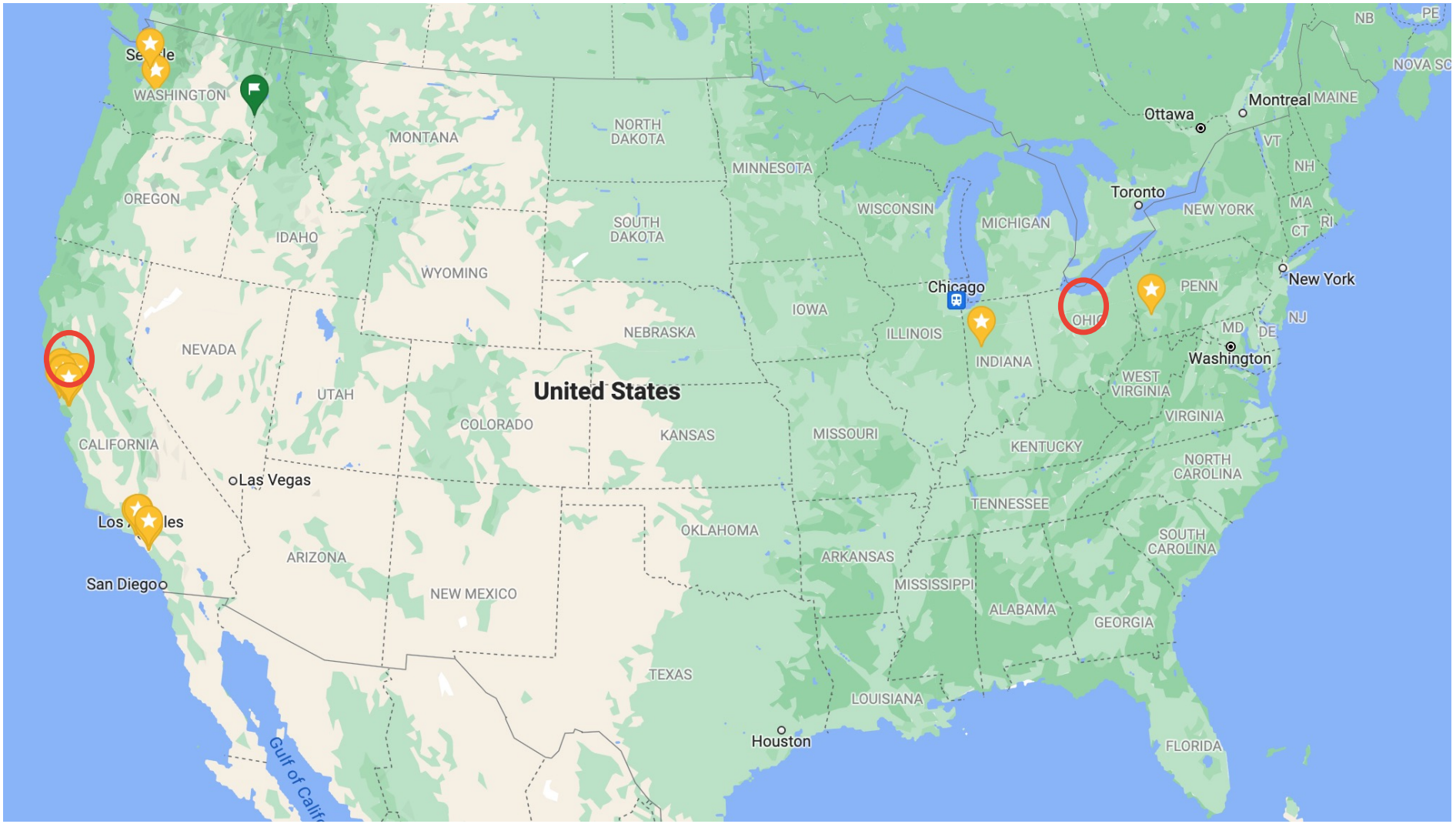
Robotics initially attracted me as an approach to tackling the grand challenges of artificial intelligence (AI). Over time, I found the pursuit of AI without purpose unsatisfying. I joined Georgia Tech in 2006, where **my research focused on enabling robots to provide intelligent physical assistance in the context of healthcare**. A full solution would likely include robots that are physically and socially intelligent in human environments, which would be consistent with notions of artificial general intelligence (AGI). The robots would also be intent on helping humans flourish, which is the type of success I'd like to see.

Henry Evans, an amazing person with severe quadriplegia, shaves himself using a PR2 robot in his home.


Our research vision is to enable robots to assist people
limitations with activities of daily living (ADLs)
1 billion individuals worldwide needed assistance with ADLs according to a survey in 2021 [Abdi et al. 2021]
Basic Activities of Daily Living (B-ADLs) – Feeding, Dressing, Transferring, etc.
Instrumental Activities of Daily Living (IADLs) – Housework, Laundry, Meal preparation, etc.

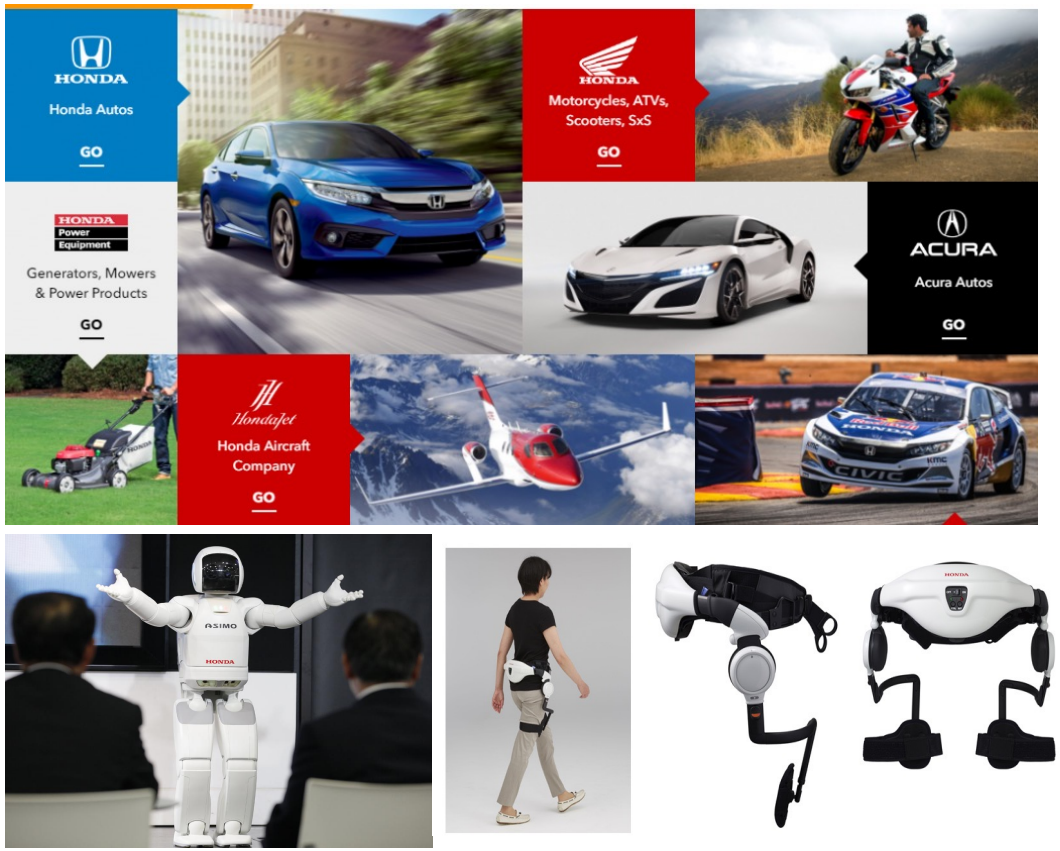

自動駕駛實驗車
Autonomous Vehicle

Spend time on Communication


WASHINGTON
OREGON
IDAHO
MONTANA
NORTH DAKOTA
SOUTH DAKOTA
MINNESOTA
WISCONSIN
MICHIGAN
WYOMING
NEBRASKA
IOWA
ILLINOIS
INDIANA
OHIO
NEVADA
UTAH
COLORADO
KANSAS
MISSOURI
CALIFORNIA
Las Vegas
ARIZONA
NEW MEXICO
OKLAHOMA
TEXAS
ARKANSAS
LOUISIANA
MISSISSIPPI
ALABAMA
GEORGIA
FLORIDA
KENTUCKY
TENNESSEE
WEST VIRGINIA
VIRGINIA
NORTH CAROLINA
SOUTH CAROLINA
PENN
NEW YORK
MAINE
VT
NH
MA
CT
RI
NJ
MD
DE
United States
Chicago
Toronto
Ottawa
Montreal
New York
Washington
Houston
San Diego
NB
PE
NOVA SC

Google Map

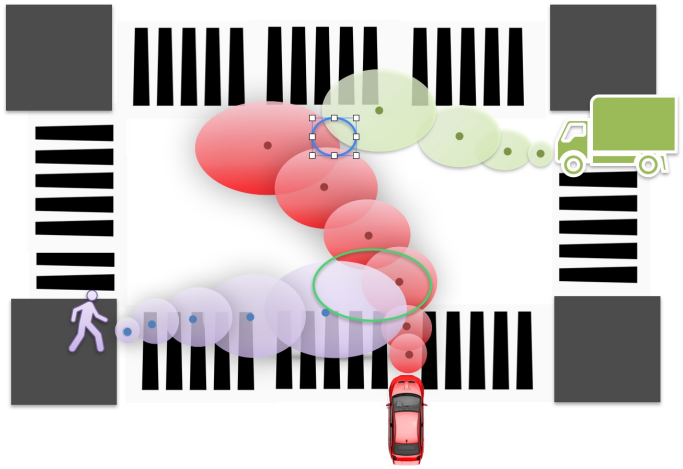
Honda Research Institute USA



https://indieseducation.com/what-are-other-product-of-honda/

Industrial Research Lab

- A very special role for a company
- Work on applied research
  - Discover new scientific knowledge that has practical aim or objective
  - For instance, how to predict the future location of traffic participants?

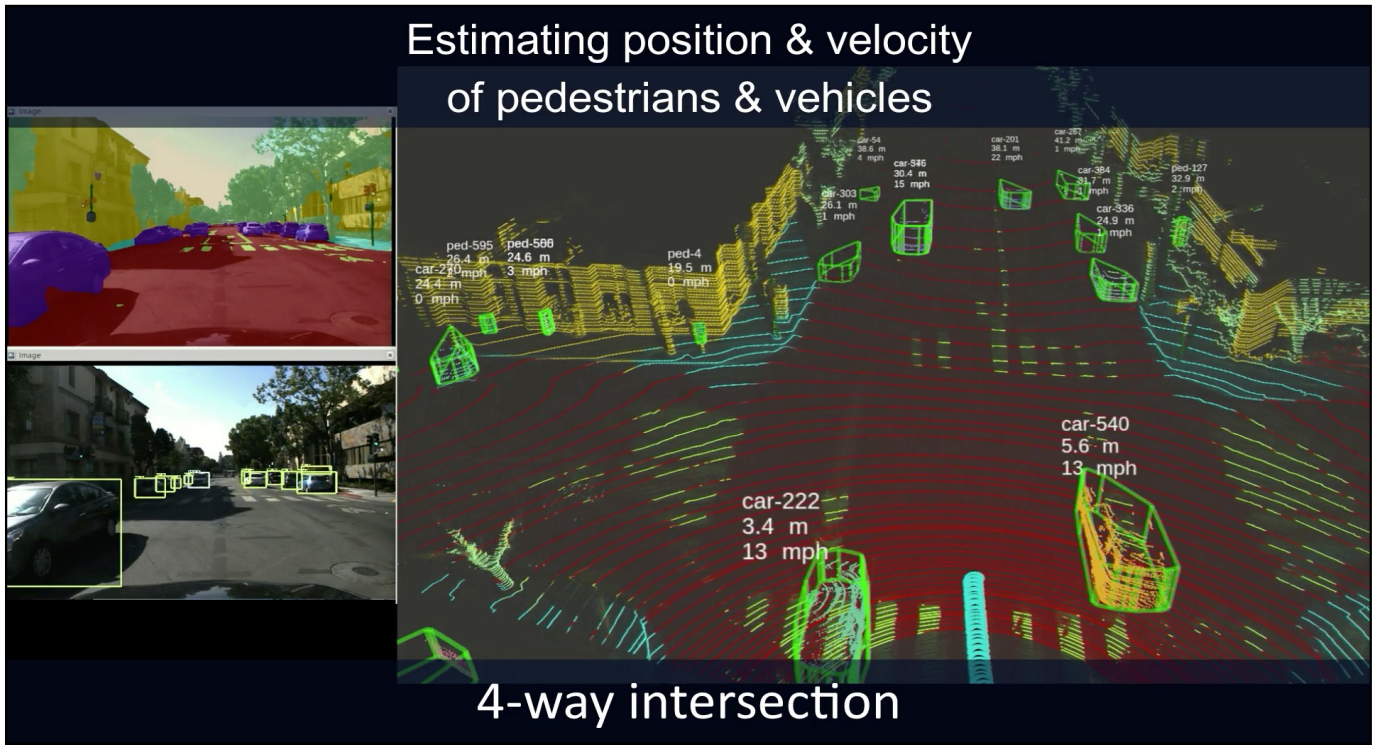


My First Project

- Goal: reduce annotation cost using multimodal data (cameras, LiDARs, and IMU)
- Start from scratch!

3 cameras, 1 64-bin LiDAR, D-GPS,
CANBUS (steering wheel, accel, brake,…)

3D Scene Analysis and Reconstruction

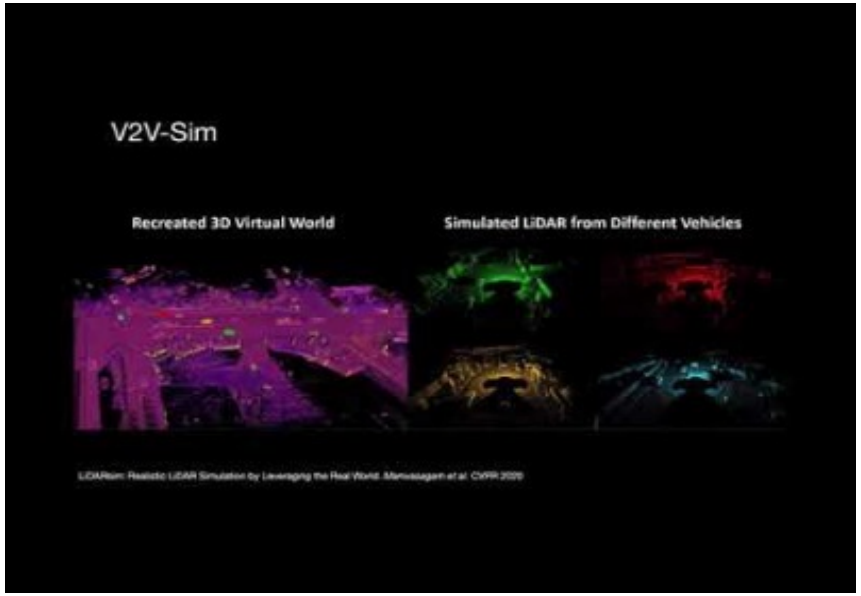


Wang, Nararayan, Patil, **Chen**, "A 3D Dynamic Scene Analysis Framework for Development of Intelligent Transportation Systems" IV 2018

Built it with a team of 5

Extension

- The company treated it as an important project
- What's next?
  - Strengthen the framework
  - Maximize its impacts for other applications
    - Digitalize real world for simulations



All the sudden, the company's direction shifted…

https://youtu.be/dA5qub7N0U8?t=67

Attention Shifting

- All the members were asked to work on another important project
- I did not fully follow what the company requested
  - I do not want to “complete different projects…”
  - I knew I do not fit in a corporate world :-)
  - Not a good team player…
- I explore the other direction and have spent the past 4 years on that (and more)!

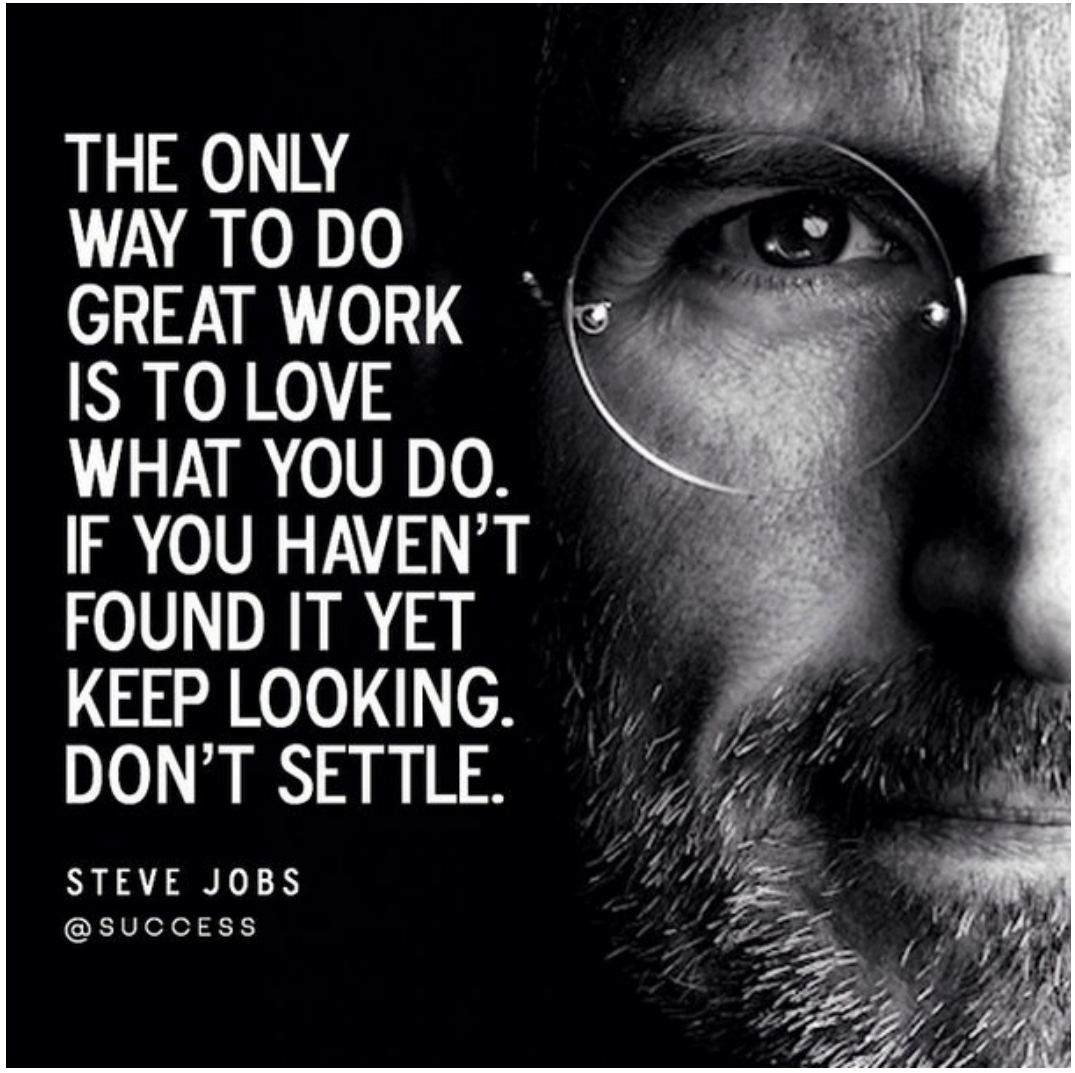
I was right because the topic ended again…

The direction becomes important…

我終於找到我可以侃侃而談的主題 :-)

Takeaway

- While it is a research lab, its directions are set according to the company's need
  - **It is a good time to ask yourself "What do you want?"**
- Be a good team player
- Spend your time to discuss alternatives! Don't keep them in your mind!


THE ONLY
WAY TO DO
GREAT WORK
IS TO LOVE
WHAT YOU DO.
IF YOU HAVEN'T
FOUND IT YET
KEEP LOOKING.
DON'T SETTLE.
STEVE JOBS
@SUCCESS

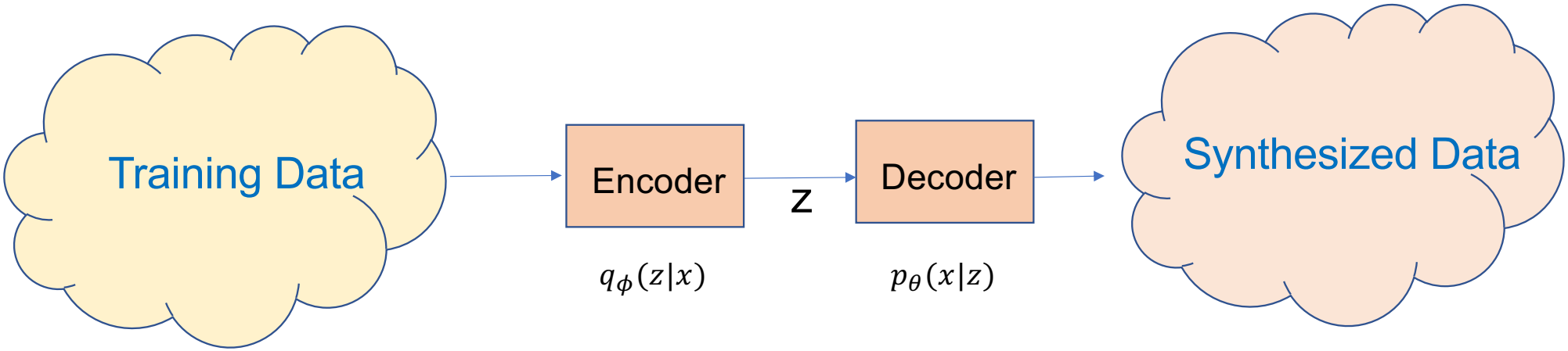
Reference

- Kingma, Diederik P., and Max Welling. "Auto-encoding variational bayes. ICLR 2014

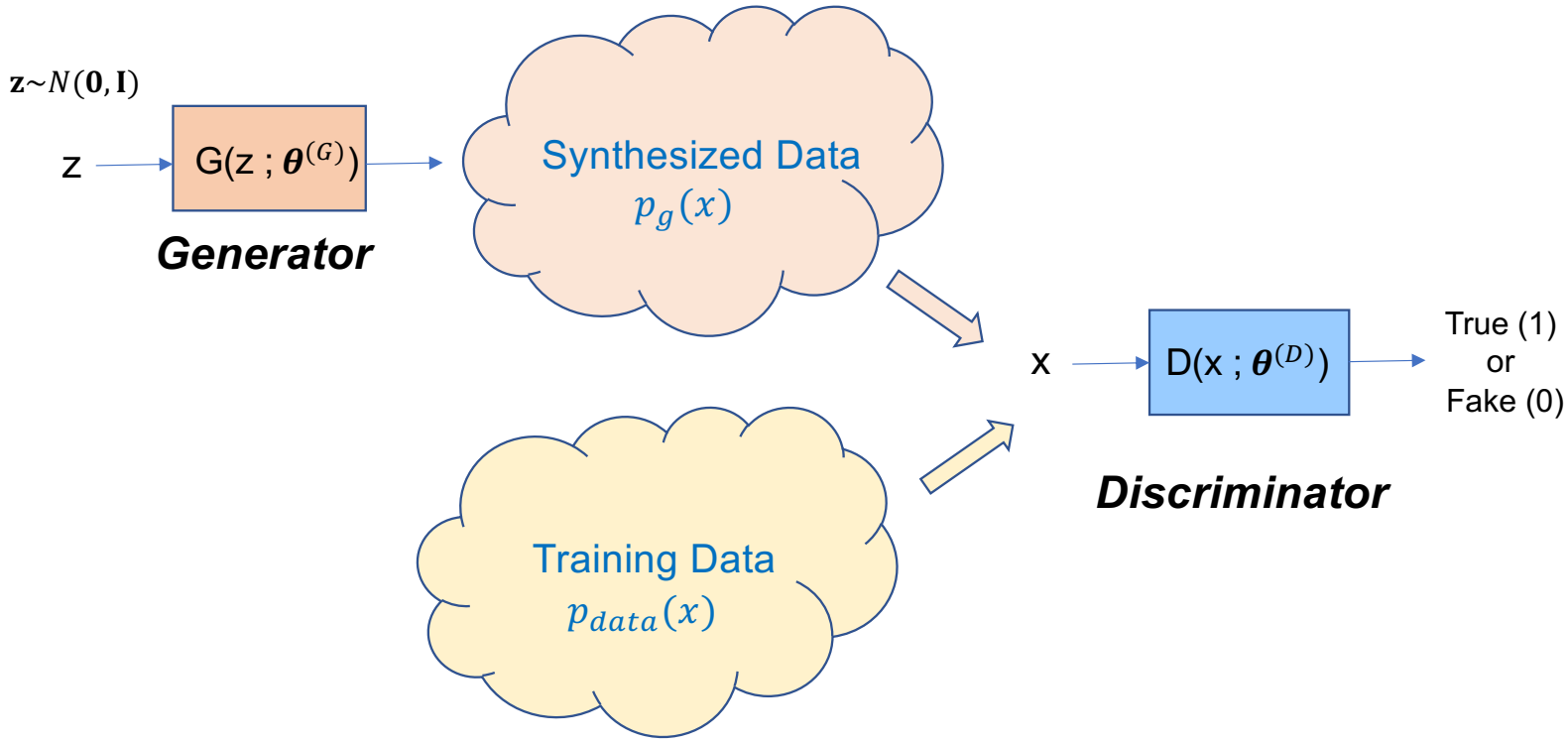
- Rezende, Danilo Jimenez, Shakir Mohamed, and Daan Wierstra. "Stochastic backpropagation and variational inference in deep latent Gaussian models." *ICML 2014.*

Diffusion Probabilistic Model (DPM)

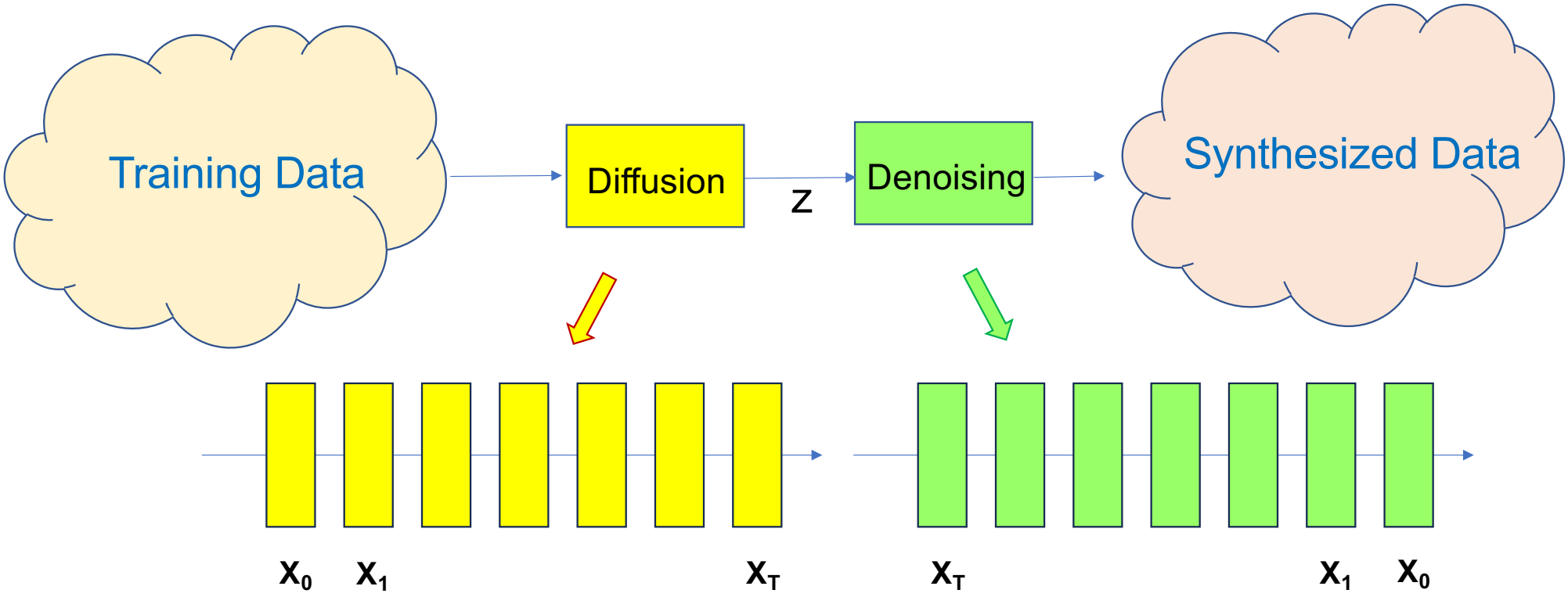
VAE (Variational Auto-Encoder)

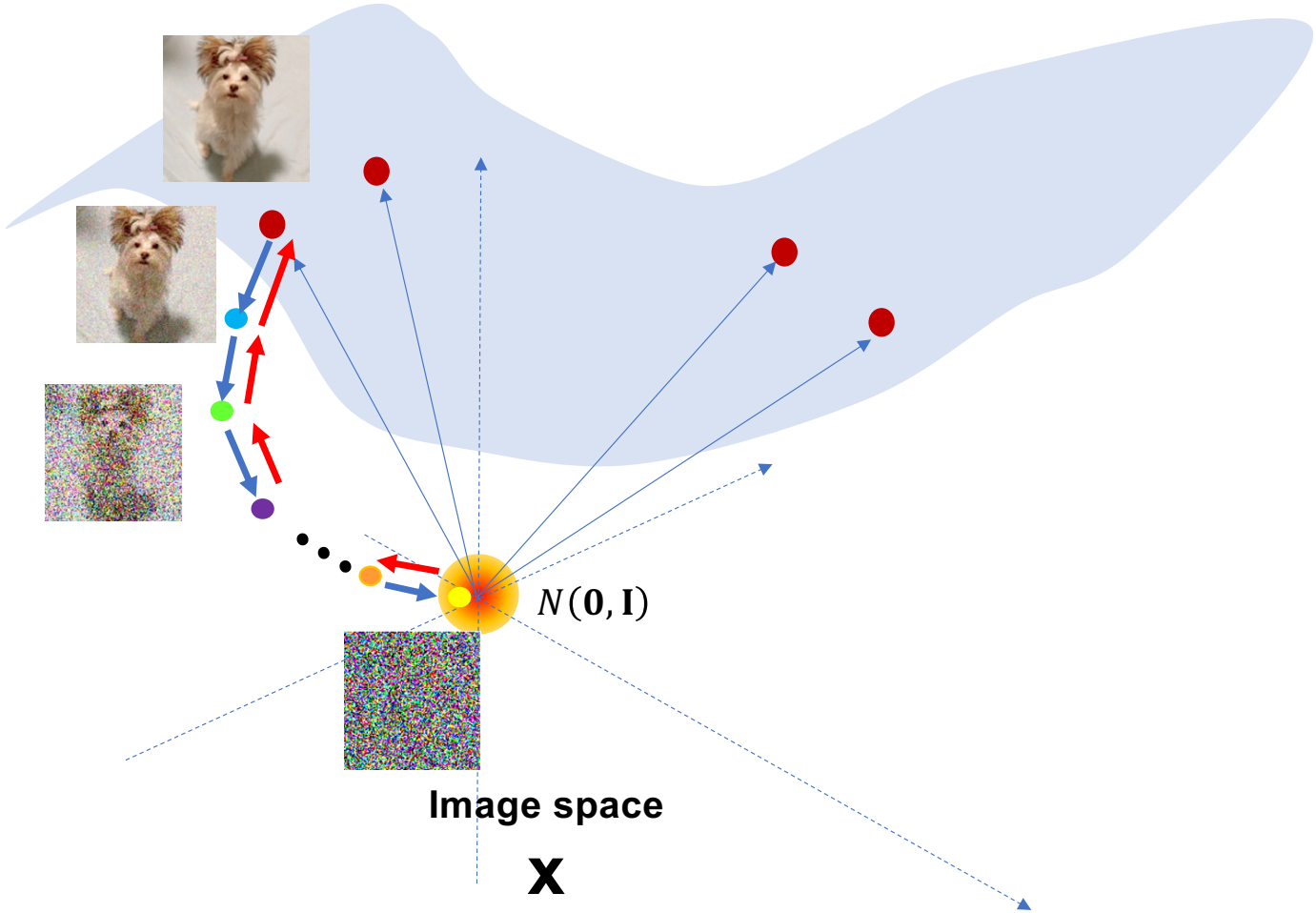


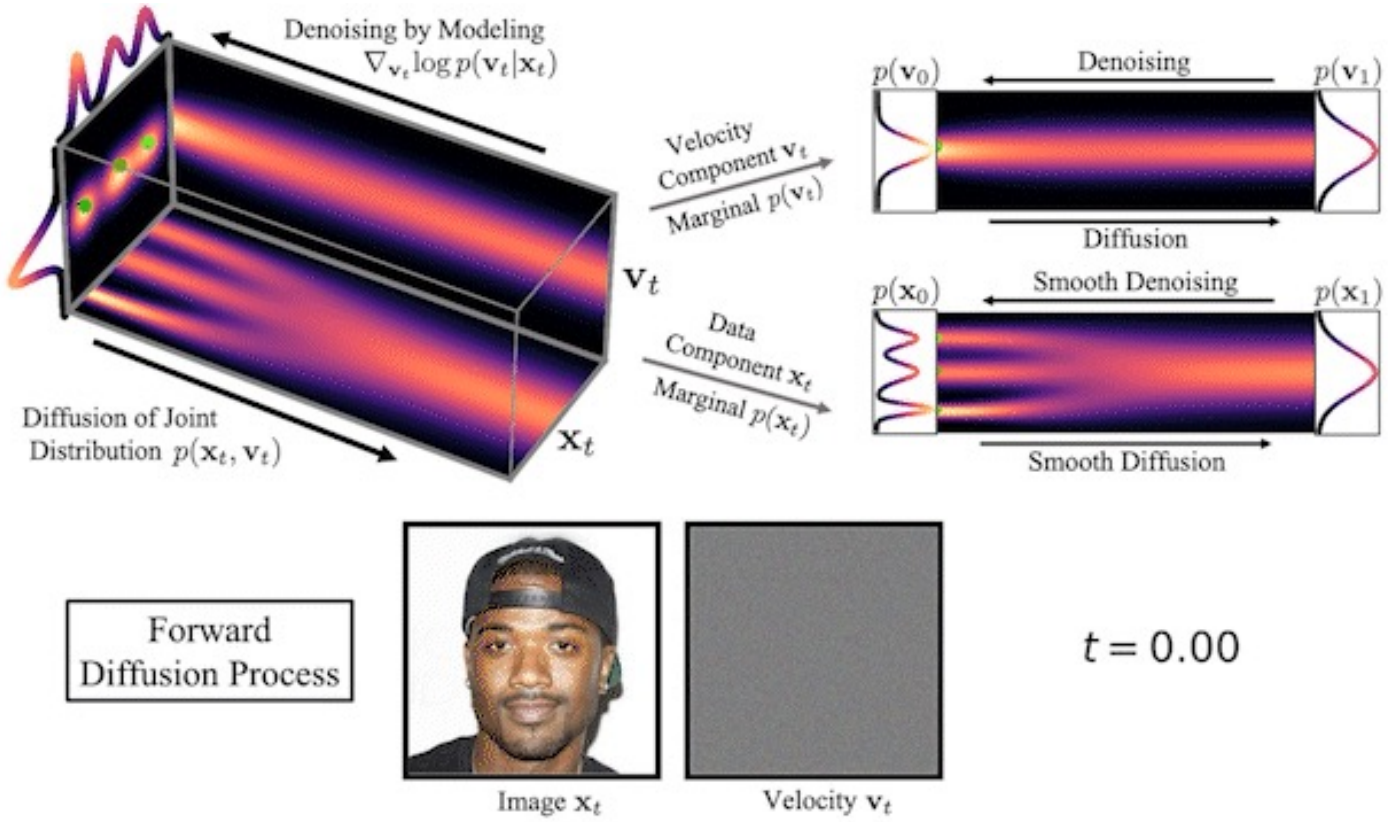
GAN (Generative Adversarial Network)



DPM (Diffusion Probabilistic Model)




$N(\mathbf{0}, \mathbf{I})$
Image space
x



https://developer.nvidia.com/blog/improving-diffusion-models-as-an-alternative-to-gans-part-2/

- $\mathbf{x}_0$: It is the original image, which is the same as $\mathbf{x}$ in VAE.
- $\mathbf{x}_T$: It is the latent variable, which is the same as $\mathbf{z}$ in VAE. As explained above, we choose $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$ for simplicity, tractability, and computational efficiency.
- $\mathbf{x}_1, \ldots, \mathbf{x}_{T-1}$: They are the intermediate states. They are also the latent variables, but they are not white Gaussian.

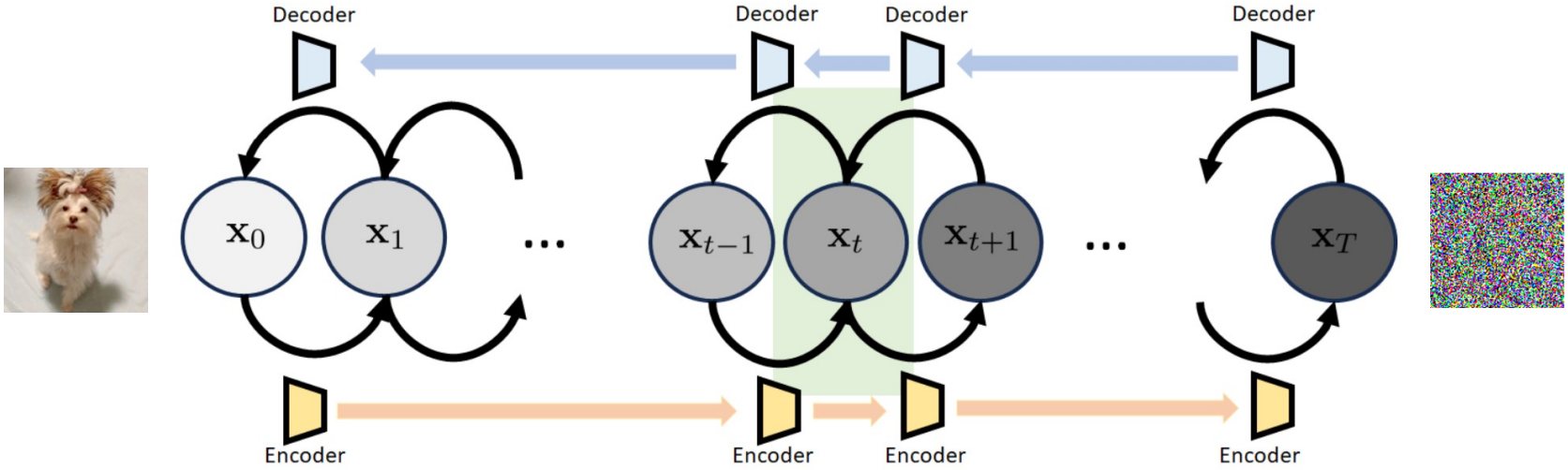


Figure 6: Variational diffusion model by Kingma et al [22]. In this model, the input image is $\mathbf{x}_0$ and the white noise is $\mathbf{x}_T$. The intermediate variables (or states) $\mathbf{x}_1, \ldots, \mathbf{x}_{T-1}$ are latent variables. The transition from $\mathbf{x}_{t-1}$ to $\mathbf{x}_t$ is analogous to the forward step (encoder) in VAE, whereas the transition from $\mathbf{x}_t$ to $\mathbf{x}_{t-1}$ is analogous to the reverse step (decoder) in VAE. In variational diffusion models, the input dimension and the output dimension of the encoders/decoders are identical.

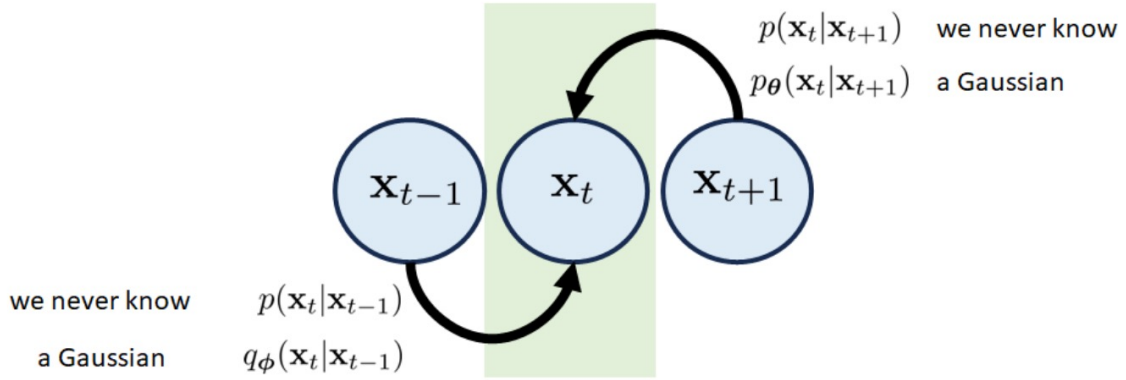


Figure 7: The transition block of a variational diffusion model consists of three nodes. The transition distributions $p(\mathbf{x}_t|\mathbf{x}_{t+1})$ and $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ are not accessible, but we can approximate them by Gaussians.

Markov Chain Structure: the transition distributions are only dependent on its immediate previous stage

$$q_\phi(\mathbf{x}_t|\mathbf{x}_{t-1}) \stackrel{\text{def}}{=} \mathcal{N}(\mathbf{x}_t \,|\, \sqrt{\alpha_t}\mathbf{x}_{t-1}, (1-\alpha_t)\mathbf{I})$$

$$\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{(1-\alpha_t)}\boldsymbol{\epsilon}, \qquad \text{where } \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}).$$

Theorem 2.1. (Why $\sqrt{\alpha}$ and $1-\alpha$?) Suppose that $q_{\boldsymbol{\phi}}(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t \,|\, a\mathbf{x}_{t-1}, b^2\mathbf{I})$ for some constants a and b. If we want to choose a and b such that the distribution of $\mathbf{x}_t$ will become $\mathcal{N}(0, \mathbf{I})$, then it is necessary that

$$a = \sqrt{\alpha} \qquad \text{and} \qquad b = \sqrt{1-\alpha}.$$

Therefore, the transition distribution is

$$q_{\boldsymbol{\phi}}(\mathbf{x}_t|\mathbf{x}_{t-1}) \stackrel{\text{def}}{=} \mathcal{N}(\mathbf{x}_t \,|\, \sqrt{\alpha}\mathbf{x}_{t-1}, (1-\alpha)\mathbf{I}). \tag{39}$$

Proof. We want to show that $a = \sqrt{\alpha}$ and $b = \sqrt{1-\alpha}$. For the distribution shown in Eqn (38), the equivalent sampling step is:

$$\mathbf{x}_t = a\mathbf{x}_{t-1} + b\boldsymbol{\epsilon}_{t-1}, \qquad \text{where} \qquad \boldsymbol{\epsilon}_{t-1} \sim \mathcal{N}(0, \mathbf{I}). \tag{40}$$

We can carry on the recursion to show that

$$\begin{aligned}
\mathbf{x}_t &= a\mathbf{x}_{t-1} + b\boldsymbol{\epsilon}_{t-1} \\
&= a(a\mathbf{x}_{t-2} + b\boldsymbol{\epsilon}_{t-2}) + b\boldsymbol{\epsilon}_{t-1} && \text{(substitute } \mathbf{x}_{t-1} = a\mathbf{x}_{t-2} + b\boldsymbol{\epsilon}_{t-2}) \\
&= a^2\mathbf{x}_{t-2} + ab\boldsymbol{\epsilon}_{t-2} + b\boldsymbol{\epsilon}_{t-1} && \text{(regroup terms)} \\
&= \vdots \\
&= a^t\mathbf{x}_0 + b\underbrace{\left[\boldsymbol{\epsilon}_{t-1} + a\boldsymbol{\epsilon}_{t-2} + a^2\boldsymbol{\epsilon}_{t-3} + \ldots + a^{t-1}\boldsymbol{\epsilon}_0\right]}_{\stackrel{\text{def}}{=}\mathbf{w}_t}.
\end{aligned} \tag{41}$$

The finite sum above is a sum of independent Gaussian random variables. The mean vector $\mathbb{E}[\mathbf{w}_t]$ remains zero because everyone has a zero mean. The covariance matrix (for a zero-mean vector) is

$$\begin{aligned}
\text{Cov}[\mathbf{w}_t] &\stackrel{\text{def}}{=} \mathbb{E}[\mathbf{w}_t\mathbf{w}_t^T] \\
&= b^2(\text{Cov}(\boldsymbol{\epsilon}_{t-1}) + a^2\text{Cov}(\boldsymbol{\epsilon}_{t-2}) + \ldots + (a^{t-1})^2\text{Cov}(\boldsymbol{\epsilon}_0)) \\
&= b^2(1 + a^2 + a^4 + \ldots + a^{2(t-1)})\mathbf{I} \\
&= b^2 \cdot \frac{1 - a^{2t-1}}{1 - a^2}\mathbf{I}.
\end{aligned}$$

As $t \to \infty$, $a^t \to 0$ for any $0 < a < 1$. Therefore, at the limit when $t = \infty$,

So, if we want $\lim_{t\to\infty} \text{Cov}[\mathbf{w}_t] = \mathbf{I}$ (so that the distribution of $\mathbf{x}_t$ will approach $\mathcal{N}(0, \mathbf{I})$), then we need

$$1 = \frac{b^2}{1 - a^2},$$

or equivalently $b = \sqrt{1 - a^2}$. Now, if we let $a = \sqrt{\alpha}$, then $b = \sqrt{1-\alpha}$. This will give us

$$\mathbf{x}_t = \sqrt{\alpha}\mathbf{x}_{t-1} + \sqrt{1-\alpha}\boldsymbol{\epsilon}_{t-1}. \tag{42}$$

Given the transition probability, we know that if $\mathbf{x}_t \sim q_{\boldsymbol{\phi}}(\mathbf{x}_t|\mathbf{x}_{t-1})$ then

$$\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{(1-\alpha_t)}\boldsymbol{\epsilon}, \qquad \text{where } \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}).$$

Theorem 2.2. (Conditional Distribution $q_{\boldsymbol{\phi}}(\mathbf{x}_t|\mathbf{x}_0)$). The conditional distribution $q_{\boldsymbol{\phi}}(\mathbf{x}_t|\mathbf{x}_0)$ is given by

$$q_{\boldsymbol{\phi}}(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t \,|\, \sqrt{\overline{\alpha}_t}\mathbf{x}_0, \;\; (1-\overline{\alpha}_t)\mathbf{I}), \tag{43}$$

where $\overline{\alpha}_t = \prod_{i=1}^{t} \alpha_i$.

Forward Pass

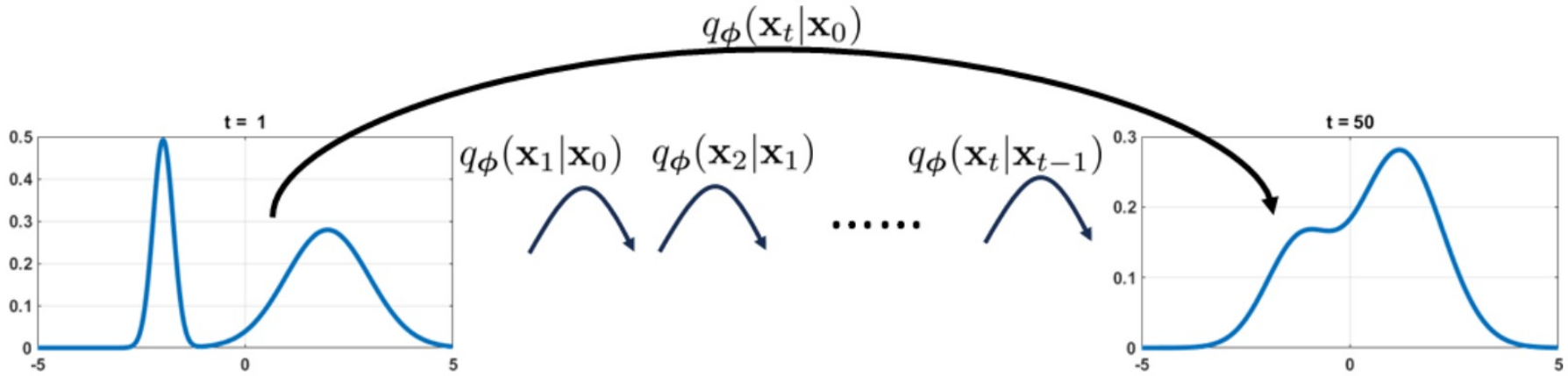
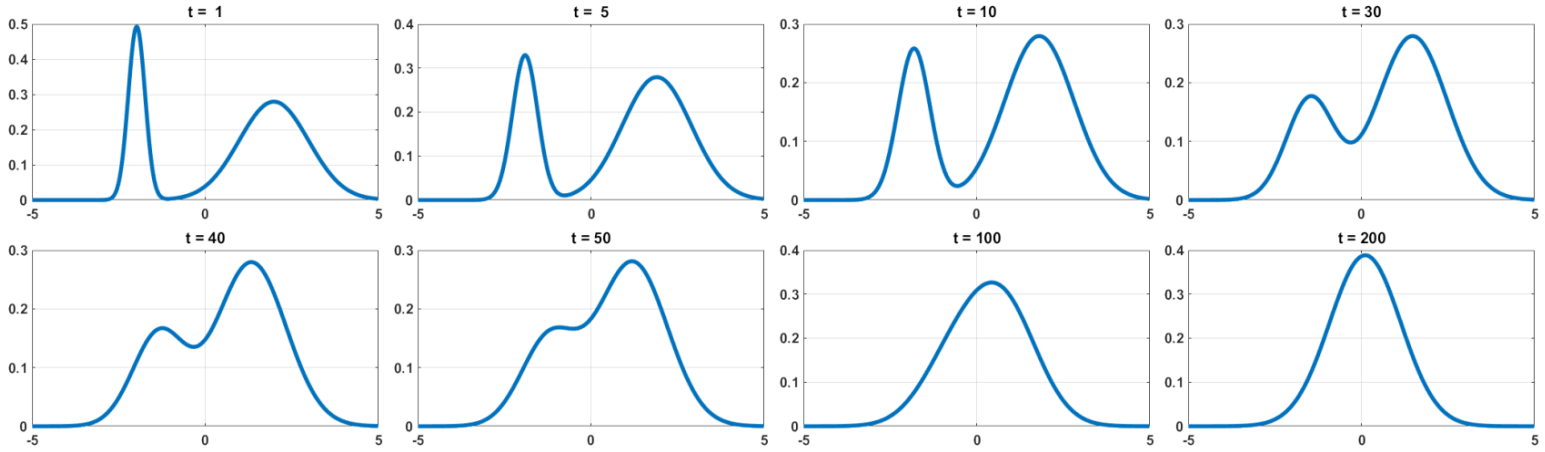


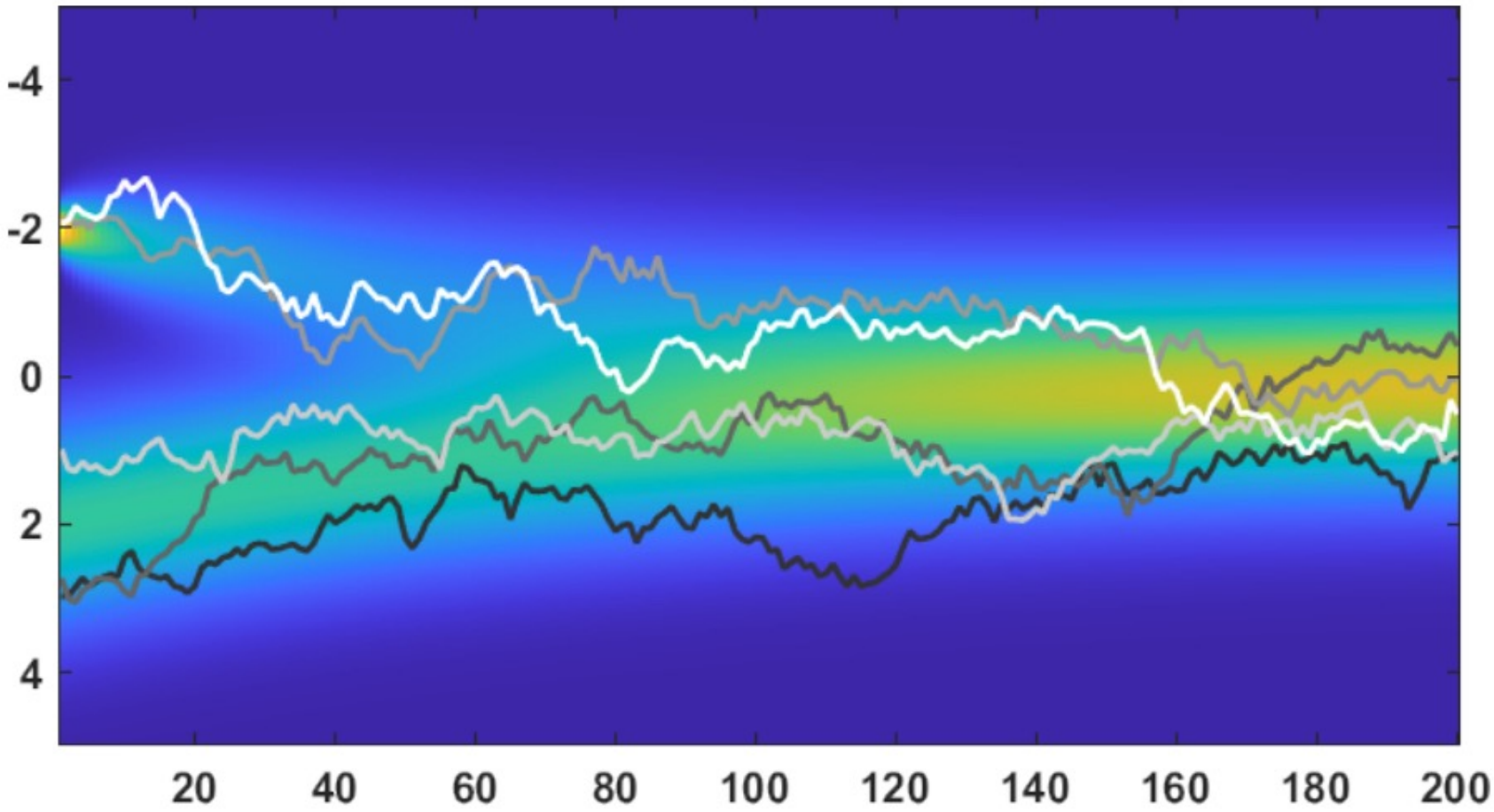
Figure 10: The difference between $q_\phi(\mathbf{x}_t|\mathbf{x}_{t-1})$ and $q_\phi(\mathbf{x}_t|\mathbf{x}_0)$.

Example

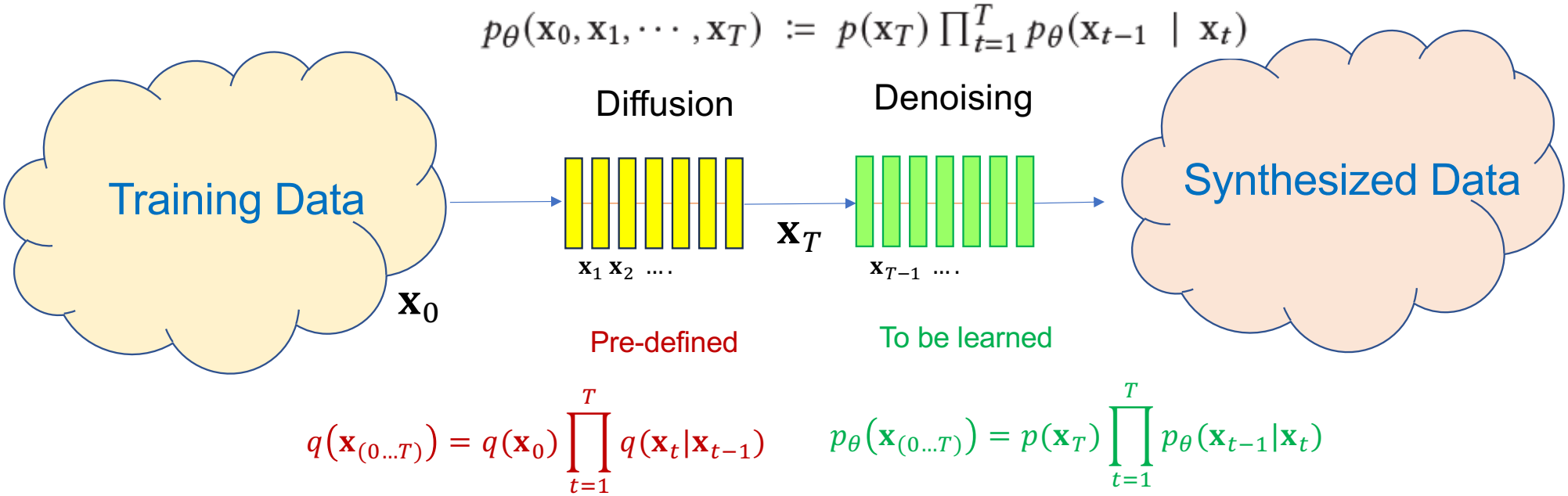
$$\mathbf{x}_0 \sim p_0(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \sigma_k^2 \mathbf{I})$$

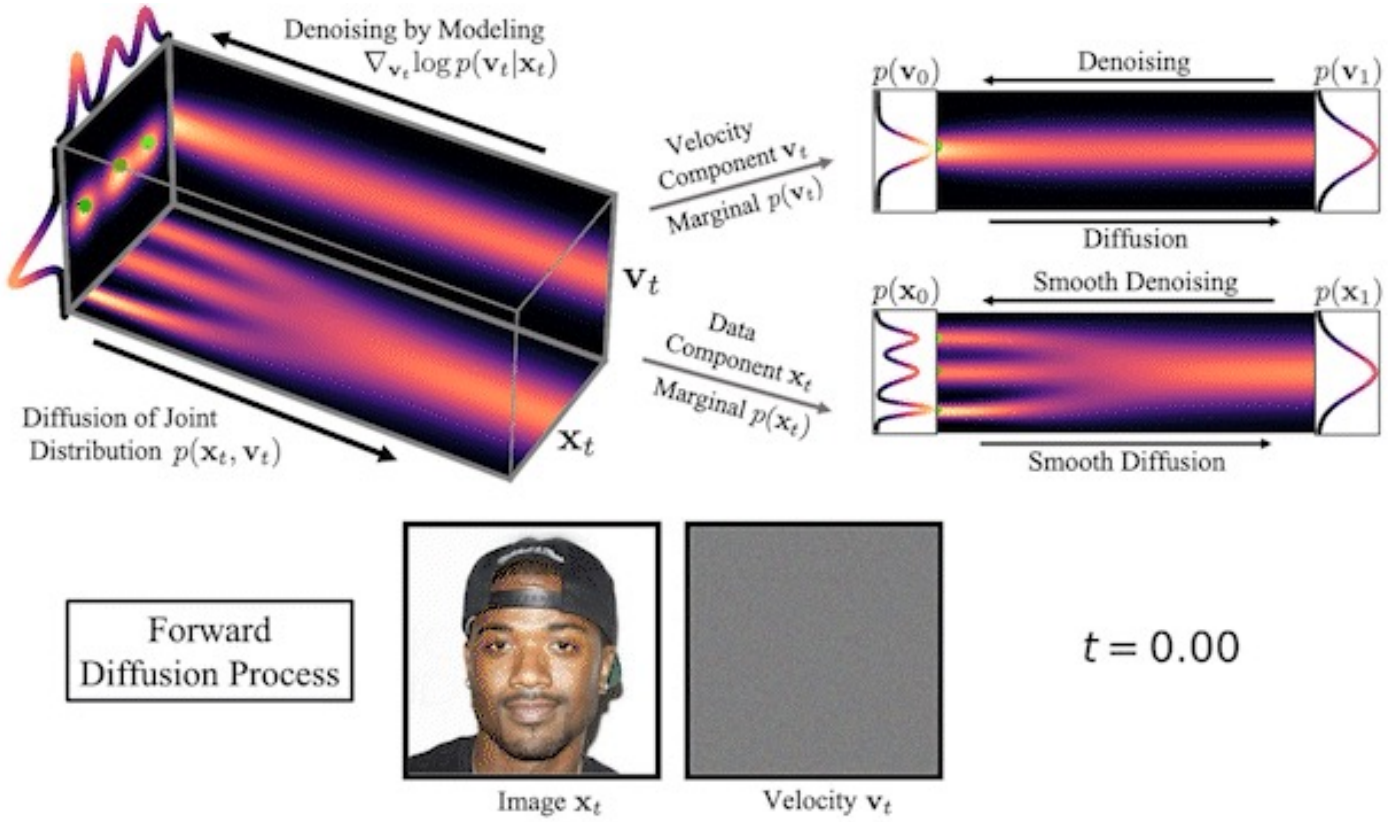
$$\begin{aligned}\mathbf{x}_t \sim p_t(\mathbf{x}) &= \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x} \mid \sqrt{\overline{\alpha}_t}\boldsymbol{\mu}_k, (1-\overline{\alpha}_t)\mathbf{I} + \overline{\alpha}_t \sigma_k^2 \mathbf{I}) \\ &= \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x} \mid \sqrt{\alpha^t}\boldsymbol{\mu}_k, (1-\alpha^t)\mathbf{I} + \alpha^t \sigma_k^2 \mathbf{I}), \qquad \text{if } \alpha_t = \alpha \text{ so that } \overline{\alpha}_t = \prod_{i=1}^{t} \alpha = \alpha^t.\end{aligned} \tag{48}$$




-4
-2
0
2
4
20
40
60
80
100
120
140
160
180
200

DPM (Diffusion Probabilistic Model)





https://developer.nvidia.com/blog/improving-diffusion-models-as-an-alternative-to-gans-part-2/