CSC420: Tutorial 4

VAE and Diffusion Model

Michael Neumayr

September 28, 2025

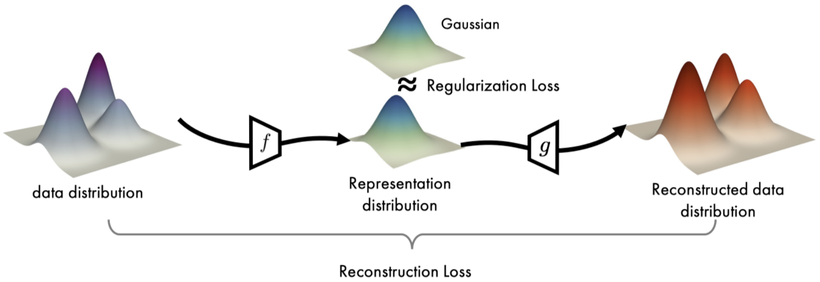
overview

- P1: VAE and Variational Inference Background
- P2: DDPM From Scratch

[0] material and visualizations inspired by and adapted from slides by David Lindell and Stephan Günnemann

vae overview

VAE conceptually:



train with ELBO objective:

$$\mathcal{L}(\theta, \phi; \mathbf{x}) = \underbrace{-\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})]}_{\text{reconstruction term}} + \underbrace{D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))}_{\text{regularization term}}$$

how did we get here

goals:

- **encoder:** we want to map high-dimensional data $\boldsymbol{x}$ to a lower-dimensional latent space $\boldsymbol{z}$
- **decoder:** we want to be able to sample the latent space $\boldsymbol{z}$ to generate new data $\boldsymbol{x}$

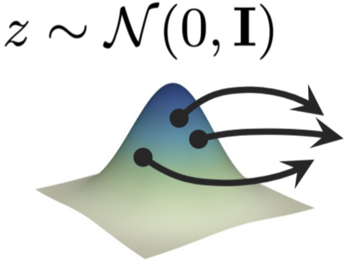
use **latent variable model** (in our case, a VAE):

- assume that a few lower-dimensional latent factors $\boldsymbol{z}$ can explain our complex data $\boldsymbol{x}$
- exploit low-dim. latent structure to facilitate modeling and sampling of distribution $p_{\theta}(\boldsymbol{x})$ over high-dim. data $\boldsymbol{x}$

main idea: true distribution $p(\boldsymbol{x})$ is “complex” (e.g. images), but the conditional distribution $p(\boldsymbol{x}|\boldsymbol{z})$ is “simple” (e.g. Gaussian)

generate data in two steps:

$\boldsymbol{z} \sim p_{\boldsymbol{\theta}}(\boldsymbol{z})$ (sample latent space $\boldsymbol{z}$)



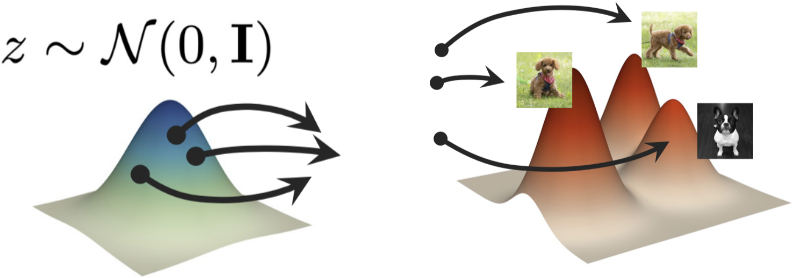
since $p_{\boldsymbol{\theta}}(\boldsymbol{z})$ standard normal (no parameters θ), use $p(\boldsymbol{z})$

how did we get here

main idea: true distribution $p(\boldsymbol{x})$ is "complex" (e.g. images), but the conditional distribution $p(\boldsymbol{x}|\boldsymbol{z})$ is "simple" (e.g. Gaussian)

generate data in two steps:

$$\boldsymbol{z} \sim p(\boldsymbol{z}) \qquad \text{(sample latent space } \boldsymbol{z}\text{)}$$
$$\boldsymbol{x} \sim p_{\theta}(\boldsymbol{x}|\boldsymbol{z}) \qquad \text{(generate data conditional on } \boldsymbol{z}\text{)}$$



how did we get here

generate data in two steps:

$$
\begin{aligned}
\boldsymbol{z} &\sim p(\boldsymbol{z}) && \text{(sample latent variable } \boldsymbol{z}\text{)} \\
\boldsymbol{x} &\sim p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) && \text{(generate data conditional on } \boldsymbol{z}\text{)}
\end{aligned}
$$

together, we get the joint distribution

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) = p(\boldsymbol{z}) p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$$

and model the full distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ by **marginalizing over $\boldsymbol{z}$**:

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \int p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) \, \mathrm{d}\boldsymbol{z} = \int p(\boldsymbol{z}) p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) \, \mathrm{d}\boldsymbol{z} = \mathbb{E}_{\boldsymbol{z} \sim p(\boldsymbol{z})}[p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z})] \tag{1}$$

our tasks in this framework

encoder: we want to map high-dimensional data $\boldsymbol{x}$ to a lower-dimensional latent space $\boldsymbol{z}$

- **inference:** with sample $\boldsymbol{x}$, find posterior distribution over $\boldsymbol{z}$

$$p_\theta(\boldsymbol{z} \mid \boldsymbol{x}) = \frac{p_\theta(\boldsymbol{x} \mid \boldsymbol{z})\, p(\boldsymbol{z})}{p_\theta(\boldsymbol{x})}$$

- but we need to model the distribution $p_\theta(\boldsymbol{x})$ for our data first

our tasks in this framework

learning: given empirical dataset $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^N$ (assume i.i.d.), find parameters θ that best explain data

- typically, we **maximize the log-likelihood** over the dataset

$$\max_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{X}) = \max_{\boldsymbol{\theta}} \frac{1}{N} \sum_{i=1}^{N} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i)$$

- for simplicity, look at a single data point $\boldsymbol{x}$, using 1 we get

$$\max_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \max_{\boldsymbol{\theta}} \log \int p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z})\, p(\boldsymbol{z})\, \mathrm{d}\boldsymbol{z} \tag{2}$$

- what is the problem?

our tasks in this framework

learning: given empirical dataset $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^N$ (assume i.i.d.), find parameters θ that best explain data

- typically, we **maximize the log-likelihood** over the dataset

$$\max_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{X}) = \max_{\boldsymbol{\theta}} \frac{1}{N} \sum_{i=1}^{N} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i)$$

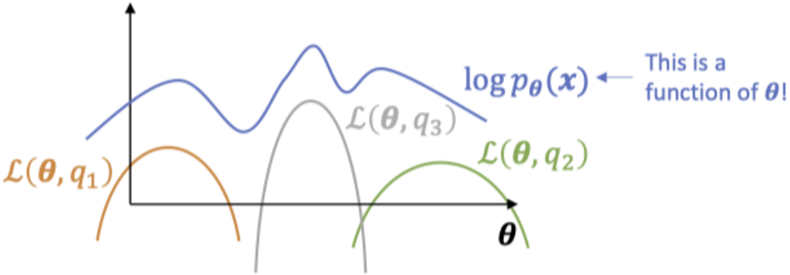
- for simplicity, look at a single data point $\boldsymbol{x}$, using 1 we get

$$\max_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \max_{\boldsymbol{\theta}} \underbrace{\log \int p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z})\, p(\boldsymbol{z})\, \mathrm{d}\boldsymbol{z}}_{f(\boldsymbol{\theta})} \tag{2}$$

- $f(\theta)$ is **intractable** to evaluate: no closed-form solution and numerical integration is infeasible ($\boldsymbol{z}$ still high-dimensional)

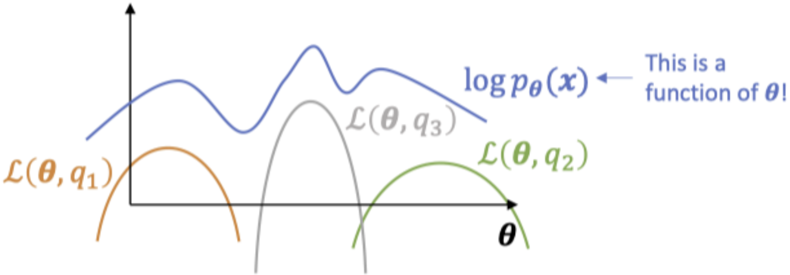
log-likelihood maximization

- $f(\theta)$ is **intractable** to evaluate: no closed-form solution and numerical integration is infeasible ($\boldsymbol{z}$ still high-dimensional)
- instead, find a **lower bound** on the log-likelihood using easy-to-evaluate functions (for us multivariate Gaussians)



log-likelihood maximization

- $f(\theta)$ is **intractable** to evaluate: no closed-form solution and numerical integration is infeasible ($\boldsymbol{z}$ still high-dimensional)
- instead, find a lower bound on the log-likelihood using easy-to-evaluate functions (for us multivariate Gaussians)



$\rightarrow$ $\underbrace{\text{variational}}_{\text{optimize over functions}}$ $\underbrace{\text{inference}}_{\text{infer latent features } \boldsymbol{z} \text{ from data } \boldsymbol{x}}$

ELBO derivation

- still looking for objective to tractably optimize $\boldsymbol{\theta}$ by substituting $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ maximization with lower bound in optim
- let $q(\boldsymbol{z})$ be an arbitrary distribution over $\boldsymbol{z}$ (choose later)

$$
\begin{aligned}
\log p_\theta(\boldsymbol{x}) &= \mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z})}\big[\log p_\theta(\boldsymbol{x})\big] \\
&= \int q(\boldsymbol{z}) \log p_\theta(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{z} && \text{| def. of expectation} \\
&= \int q(\boldsymbol{z}) \log \frac{p_\theta(\boldsymbol{x},\boldsymbol{z})}{p_\theta(\boldsymbol{z}\mid\boldsymbol{x})} \,\mathrm{d}\boldsymbol{z} && \text{| def. of cond. probability} \\
&= \int q(\boldsymbol{z}) \log \left(\frac{p_\theta(\boldsymbol{x},\boldsymbol{z})}{p_\theta(\boldsymbol{z}\mid\boldsymbol{x})} \cdot \frac{q(\boldsymbol{z})}{q(\boldsymbol{z})}\right) \mathrm{d}\boldsymbol{z} && \text{| Id. trick} \\
&= \int q(\boldsymbol{z}) \log \frac{p_\theta(\boldsymbol{x},\boldsymbol{z})}{q(\boldsymbol{z})} \,\mathrm{d}\boldsymbol{z} + \int q(\boldsymbol{z}) \log \frac{q(\boldsymbol{z})}{p_\theta(\boldsymbol{z}\mid\boldsymbol{x})} \,\mathrm{d}\boldsymbol{z} \\
&= \mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z})}\left[\log \frac{p_\theta(\boldsymbol{x},\boldsymbol{z})}{q(\boldsymbol{z})}\right] + \mathrm{D}_{\mathrm{KL}}(q(\boldsymbol{z}) \,\|\, p_\theta(\boldsymbol{z}\mid\boldsymbol{x})) && \text{| defs}
\end{aligned}
$$

ELBO derivation

$$\begin{aligned}
\log p_\theta(\boldsymbol{x}) &= \mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z})}\big[\log p_\theta(\boldsymbol{x})\big] \\
&= \int q(\boldsymbol{z})\, \log p_\theta(\boldsymbol{x})\, \mathrm{d}\boldsymbol{z} && \text{| def. of expectation} \\
&= \int q(\boldsymbol{z})\, \log \frac{p_\theta(\boldsymbol{x},\boldsymbol{z})}{p_\theta(\boldsymbol{z}\mid\boldsymbol{x})}\, \mathrm{d}\boldsymbol{z} && \text{| def. of cond. probability} \\
&= \int q(\boldsymbol{z})\, \log \left(\frac{p_\theta(\boldsymbol{x},\boldsymbol{z})}{p_\theta(\boldsymbol{z}\mid\boldsymbol{x})} \cdot \frac{q(\boldsymbol{z})}{q(\boldsymbol{z})}\right) \mathrm{d}\boldsymbol{z} && \text{| Id. trick} \\
&= \int q(\boldsymbol{z})\, \log \frac{p_\theta(\boldsymbol{x},\boldsymbol{z})}{q(\boldsymbol{z})}\, \mathrm{d}\boldsymbol{z} + \int q(\boldsymbol{z})\, \log \frac{q(\boldsymbol{z})}{p_\theta(\boldsymbol{z}\mid\boldsymbol{x})}\, \mathrm{d}\boldsymbol{z} \\
&= \underbrace{\mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z})}\left[\log \frac{p_\theta(\boldsymbol{x},\boldsymbol{z})}{q(\boldsymbol{z})}\right]}_{=:\ \mathrm{ELBO}(\boldsymbol{x})} + \underbrace{\mathrm{D}_{\mathrm{KL}}(q(\boldsymbol{z}) \,\|\, p_\theta(\boldsymbol{z}\mid\boldsymbol{x}))}_{\geq 0 \text{ (gap)}} && \text{| defs}
\end{aligned}$$

- for a fixed θ, we want $q_\phi(\boldsymbol{z})$ to be close to $p_\theta(\boldsymbol{z} \mid \boldsymbol{x})$

exercises around ELBO to deepen your understanding:

1. warm up: write out ELBO from

$$\mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z})}\left[\log \frac{p_\theta(\boldsymbol{x},\boldsymbol{z})}{q(\boldsymbol{z})}\right]$$

   to

$$\mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z})}[\log p_\theta(\boldsymbol{x} \mid \boldsymbol{z})] - \mathrm{D}_{\mathrm{KL}}(q(\boldsymbol{z}) \,\|\, p(\boldsymbol{z}))$$

2. prove $\log p_{\boldsymbol{\theta}}(\boldsymbol{x}) \geq \mathrm{ELBO}(\boldsymbol{x})$ (it lower bounds log-likelihood) using Jensen's inequality → Q1 (a) in assignment 2
3. prove $\log p_{\boldsymbol{\theta}}(\boldsymbol{x}) - \mathrm{ELBO}(\boldsymbol{x}) = \mathrm{D}_{\mathrm{KL}}(q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \,\|\, p_\theta(\boldsymbol{z} \mid \boldsymbol{x}))$ (it is gap between log-likelihood and KL divergence between approximate and true posterior) → Q1 (b) in assignment 2

optimizing with elbo

optimizing $\mathrm{ELBO}(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z})}[\log p_\theta(\boldsymbol{x} \mid \boldsymbol{z})] - \mathrm{D}_{\mathrm{KL}}(q(\boldsymbol{z}) \,\|\, p(\boldsymbol{z}))$

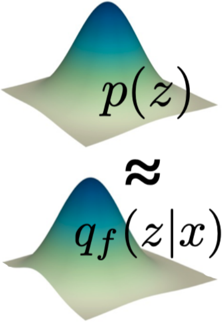
- given $\boldsymbol{x}$, we want to optimize the $\mathrm{ELBO}(\boldsymbol{x})$ wrt θ and $q(\boldsymbol{z})$
- but what are we optimizing over in the case of $q(\boldsymbol{z})$?
- choose set of tractable, parametric distributions $\mathcal{Q}$
- every distribution $q_\phi(\boldsymbol{z})$ is specified by its parameters ϕ
- best distribution $q_\phi(\boldsymbol{z}) \iff$ best parameters ϕ
- instead of optimizing $\phi^{(i)}_{\text{optimal}}$ for every data point $\boldsymbol{x}^{(i)}$, we learn a neural network that maps **every** $\boldsymbol{x}_i$ to $\phi^{(i)}_{\text{optimal}}$
  $\rightarrow$ **amortized inference**
- as $\boldsymbol{x}$ is network input, write $q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$ instead of $q_\phi(\boldsymbol{z})$:

$$\mathrm{ELBO}(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x} \mid \boldsymbol{z})] - \mathrm{D}_{\mathrm{KL}}(q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \,\|\, p(\boldsymbol{z}))$$

default choices (keep simple and standard)

prior: $p(\boldsymbol{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ (no learnable params)

- $\mathrm{D}_{\mathrm{KL}}(q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \,\|\, p(\boldsymbol{z}))$ term regularizes $q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$ to learn a distribution close to the prior $p(\boldsymbol{z})$

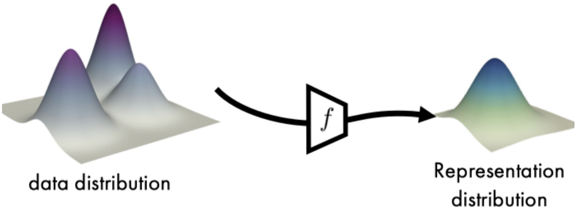


default choices (keep simple and standard)

encoder (approximate posterior):

$$q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) = \mathcal{N}\big(\boldsymbol{\mu}_\phi(\boldsymbol{x}), \mathrm{diag}(\boldsymbol{\sigma}^2_\phi(\boldsymbol{x}))\big)$$

- neural network outputs *parameters* $\boldsymbol{\mu}_\phi(\boldsymbol{x}), \boldsymbol{\sigma}^2_\phi(\boldsymbol{x})$
- given a k-dimensional latent vector $\boldsymbol{z}$, what dimension is the output tensor of the encoder?

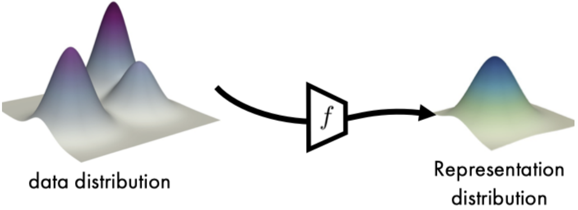


default choices (keep simple and standard)

encoder (approximate posterior):

$$q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) = \mathcal{N}\big(\boldsymbol{\mu}_\phi(\boldsymbol{x}), \mathrm{diag}(\boldsymbol{\sigma}^2_\phi(\boldsymbol{x}))\big)$$

- neural network outputs *parameters* $\boldsymbol{\mu}_\phi(\boldsymbol{x}), \boldsymbol{\sigma}^2_\phi(\boldsymbol{x})$
- given a k-dimensional latent vector $\boldsymbol{z}$, what dimension is the output tensor of the encoder?
- for every image, the encoder outputs a latent vector with dimension (k, 2): per-latent dimension mean and variance



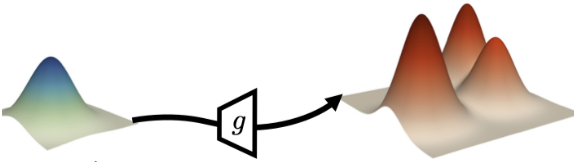
default choices (keep simple and standard)

decoder (likelihood):

- pick by data type, for example:

$$p_\theta(\boldsymbol{x} \mid \boldsymbol{z}) = \begin{cases} \mathcal{N}\big(\boldsymbol{\mu}_\theta(\boldsymbol{z}), \mathbf{I}\big), & \boldsymbol{x} \in \mathbb{R}^D \\ \text{Bernoulli}\big(\boldsymbol{\pi}_\theta(\boldsymbol{z})\big), & \boldsymbol{x} \in \{0,1\}^D \end{cases}$$

- neural network outputs *parameters* $\boldsymbol{\mu}_\theta(\boldsymbol{z})$ or $\boldsymbol{\pi}_\theta(\boldsymbol{z})$



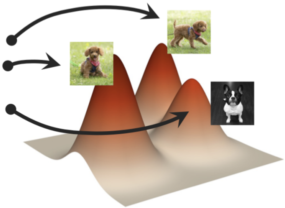
default choices (keep simple and standard)

decoder (likelihood):

- pick by data type, for example:

$$p_\theta(\boldsymbol{x} \mid \boldsymbol{z}) = \begin{cases} \mathcal{N}\big(\boldsymbol{\mu}_\theta(\boldsymbol{z}), \mathbf{I}\big), & \boldsymbol{x} \in \mathbb{R}^D \\ \text{Bernoulli}\big(\boldsymbol{\pi}_\theta(\boldsymbol{z})\big), & \boldsymbol{x} \in \{0,1\}^D \end{cases}$$

- neural network outputs *parameters* $\boldsymbol{\mu}_\theta(\boldsymbol{z})$ or $\boldsymbol{\pi}_\theta(\boldsymbol{z})$
- retrieve samples from the distribution using the parameters



additional slides

bayes terminology recap

$$\underbrace{p_{\boldsymbol{\theta}}(\boldsymbol{z} \mid \boldsymbol{x})}_{\text{posterior}} = \frac{\overbrace{p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z})}^{\text{likelihood}} \overbrace{p(\boldsymbol{z})}^{\text{prior}}}{\underbrace{p_{\boldsymbol{\theta}}(\boldsymbol{x})}_{\text{(evidence)}}}$$

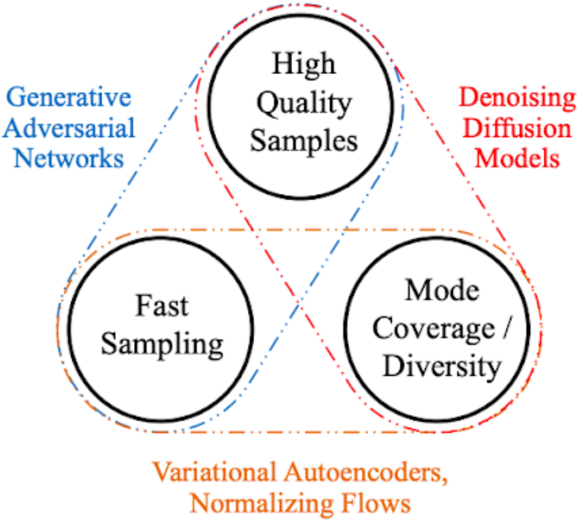
Kullback-Leibler divergence recap

- KL divergence from $q(\boldsymbol{z})$ to $p(\boldsymbol{z})$ is defined as

$$\mathrm{D}_{\mathrm{KL}}(q(\boldsymbol{z}) \,\|\, p(\boldsymbol{z})) \;:=\; \int q(\boldsymbol{z}) \log \frac{q(\boldsymbol{z})}{p(\boldsymbol{z})} \, d\boldsymbol{z}$$

- properties:
  - asymmetric, $\mathrm{D}_{\mathrm{KL}}(q(\boldsymbol{z}) \,\|\, p(\boldsymbol{z})) \neq \mathrm{D}_{\mathrm{KL}}(p(\boldsymbol{z}) \,\|\, q(\boldsymbol{z}))$ in general
  - nonnegative, $\mathrm{D}_{\mathrm{KL}}(q(\boldsymbol{z}) \,\|\, p(\boldsymbol{z})) \geq 0$
  - $\mathrm{D}_{\mathrm{KL}}(q(\boldsymbol{z}) \,\|\, p(\boldsymbol{z})) = 0 \iff p = q$ almost everywhere

high level understanding



https://developer.nvidia.com/blog/improving-diffusion-models-as-an-alternative-to-gans-part-1/