

Deep Generative Models

Lecture 8

Roman Isachenko

Moscow Institute of Physics and Technology

2022

Recap of previous lecture

Let our data \mathbf{y} comes from discrete distribution $\Pi(\mathbf{y})$.

Discrete model

- ▶ Use **discrete** model (e.x. $P(\mathbf{y}|\theta) = \text{Cat}(\pi(\theta))$).
- ▶ Minimize any suitable divergence measure $D(\Pi, P)$.

Continuous model

Use **continuous** model (e.x. $p(\mathbf{x}|\theta) = \mathcal{N}(\mu_\theta(\mathbf{x}), \sigma_\theta^2(\mathbf{x}))$), but

- ▶ **discretize** model (make the model outputs discrete): transform $p(\mathbf{x}|\theta)$ to $P(\mathbf{y}|\theta)$;
- ▶ **dequantize** data (make the data continuous): transform $\Pi(\mathbf{y})$ to $\pi(\mathbf{x})$.

Model discretization through CDF

$$F(\mathbf{x}|\theta) = \int_{-\infty}^{\mathbf{x}} p(\mathbf{x}'|\theta) d\mathbf{x}'; \quad P(\mathbf{y}|\theta) = F(\mathbf{y} + 0.5|\theta) - F(\mathbf{y} - 0.5|\theta)$$

Recap of previous lecture

Uniform dequantization bound

Let dequantize discrete distribution $\Pi(\mathbf{y})$ to continuous distribution $\pi(\mathbf{x})$ in the following way: $\mathbf{x} = \mathbf{y} + \mathbf{u}$, where $\mathbf{u} \sim U[0, 1]$.

Theorem

Fitting continuous model $p(\mathbf{x}|\theta)$ on uniformly dequantized data is equivalent to maximization of a lower bound on log-likelihood for a discrete model:

$$P(\mathbf{y}|\theta) = \int_{U[0,1]} p(\mathbf{y} + \mathbf{u}|\theta) d\mathbf{u}$$

Variational dequantization bound

Introduce variational dequantization noise distribution $q(\mathbf{u}|\mathbf{y})$ and treat it as an approximate posterior.

$$\log P(\mathbf{y}|\theta) \geq \int q(\mathbf{u}|\mathbf{y}) \log \frac{p(\mathbf{y} + \mathbf{u}|\theta)}{q(\mathbf{u}|\mathbf{y})} d\mathbf{u} = \mathcal{L}(q, \theta).$$

Recap of previous lecture

Theorem

$$\frac{1}{n} \sum_{i=1}^n KL(q(\mathbf{z}|\mathbf{x}_i)||p(\mathbf{z})) = KL(q_{\text{agg}}(\mathbf{z})||p(\mathbf{z})) + \mathbb{I}_q[\mathbf{x}, \mathbf{z}].$$

ELBO surgery

$$\frac{1}{n} \sum_{i=1}^n \mathcal{L}_i(q, \theta) = \underbrace{\frac{1}{n} \sum_{i=1}^n \mathbb{E}_{q(\mathbf{z}|\mathbf{x}_i)} \log p(\mathbf{x}_i|\mathbf{z}, \theta)}_{\text{Reconstruction loss}} - \underbrace{\mathbb{I}_q[\mathbf{x}, \mathbf{z}]}_{\text{MI}} - \underbrace{KL(q_{\text{agg}}(\mathbf{z})||p(\mathbf{z}))}_{\text{Marginal KL}}$$

Optimal prior

$$KL(q_{\text{agg}}(\mathbf{z})||p(\mathbf{z})) = 0 \quad \Leftrightarrow \quad p(\mathbf{z}) = q_{\text{agg}}(\mathbf{z}) = \frac{1}{n} \sum_{i=1}^n q(\mathbf{z}|\mathbf{x}_i).$$

The optimal prior distribution $p(\mathbf{z})$ is aggregated posterior $q(\mathbf{z})$.

Recap of previous lecture

- ▶ Standard Gaussian $p(\mathbf{z}) = \mathcal{N}(0, I) \Rightarrow$ over-regularization;
- ▶ $p(\mathbf{z}) = q_{\text{agg}}(\mathbf{z}) = \frac{1}{n} \sum_{i=1}^n q(\mathbf{z}|\mathbf{x}_i) \Rightarrow$ overfitting and highly expensive.

ELBO revisiting

$$\frac{1}{n} \sum_{i=1}^n \mathcal{L}_i(q, \theta) = \text{RL} - \text{MI} - \text{KL}(q_{\text{agg}}(\mathbf{z}) || p(\mathbf{z}|\lambda))$$

It is Forward KL with respect to $p(\mathbf{z}|\lambda)$.

ELBO with flow-based VAE prior

$$\begin{aligned} \mathcal{L}(\phi, \theta) &= \mathbb{E}_{q(\mathbf{z}|\mathbf{x}, \phi)} [\log p(\mathbf{x}|\mathbf{z}, \theta) + \log p(\mathbf{z}|\lambda) - \log q(\mathbf{z}|\mathbf{x}, \phi)] \\ &= \mathbb{E}_{q(\mathbf{z}|\mathbf{x}, \phi)} \left[\log p(\mathbf{x}|\mathbf{z}, \theta) + \underbrace{\left(\log p(f(\mathbf{z}, \lambda)) + \log |\det(\mathbf{J}_f)| \right)}_{\text{flow-based prior}} - \log q(\mathbf{z}|\mathbf{x}, \phi) \right] \end{aligned}$$

- ▶ RealNVP with coupling layers.
- ▶ Autoregressive flow (fast $f(\mathbf{z}, \lambda)$, slow $g(\mathbf{z}^*, \lambda)$).

Recall of previous lecture

ELBO decomposition

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \mathcal{L}(q, \boldsymbol{\theta}) + KL(q(\mathbf{z}|\mathbf{x}, \phi) || p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})).$$

- ▶ E-step of EM-algorithm: $KL(q(\mathbf{z}|\mathbf{x}, \phi) || p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})) = 0$.
(In this case the lower bound is tight $\log p(\mathbf{x}|\boldsymbol{\theta}) = \mathcal{L}(q, \boldsymbol{\theta})$).
- ▶ $q(\mathbf{z}|\mathbf{x}, \phi) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_\phi(\mathbf{x}), \boldsymbol{\sigma}_\phi^2(\mathbf{x}))$ is a unimodal distribution (not expressive enough).
- ▶ NF convert a simple distribution to a complex one. Let use NF in VAE posterior.

Apply a sequence of transformations to the random variable

$$\mathbf{z} \sim q(\mathbf{z}|\mathbf{x}, \phi) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_\phi(\mathbf{x}), \boldsymbol{\sigma}_\phi^2(\mathbf{x})).$$

Let $q(\mathbf{z}|\mathbf{x}, \phi)$ (VAE encoder) be a base distribution for a flow model.

Outline

1. Flow-based VAE posterior vs flow-based VAE prior
2. Likelihood-free learning
3. Generative adversarial networks (GAN)

Outline

1. Flow-based VAE posterior vs flow-based VAE prior
2. Likelihood-free learning
3. Generative adversarial networks (GAN)

Flows in VAE posterior

- ▶ Encoder outputs base distribution $q(\mathbf{z}|\mathbf{x}, \phi)$.
- ▶ Flow model $\mathbf{z}^* = f(\mathbf{z}, \lambda)$ transforms the base distribution $q(\mathbf{z}|\mathbf{x}, \phi)$ to the distribution $q(\mathbf{z}^*|\mathbf{x}, \phi, \lambda)$.
- ▶ Distribution $q(\mathbf{z}^*|\mathbf{x}, \phi, \lambda)$ is used as a variational distribution for ELBO maximization.
- ▶ Here ϕ – encoder parameters, λ – flow parameters.

Flow model in latent space

$$\log q(\mathbf{z}^*|\mathbf{x}, \phi, \lambda) = \log q(\mathbf{z}|\mathbf{x}, \phi) + \log \left| \det \left(\frac{d\mathbf{z}}{d\mathbf{z}^*} \right) \right|$$

$$\mathbf{z}^* = f(\mathbf{z}, \lambda) = g^{-1}(\mathbf{z}, \lambda)$$

ELBO with flow-based VAE posterior

$$\mathcal{L}(\phi, \theta, \lambda) = \mathbb{E}_{q(\mathbf{z}^*|\mathbf{x}, \phi, \lambda)} \log p(\mathbf{x}|\mathbf{z}^*, \theta) - \textcolor{violet}{KL}(q(\mathbf{z}^*|\mathbf{x}, \phi, \lambda) || p(\mathbf{z}^*)).$$

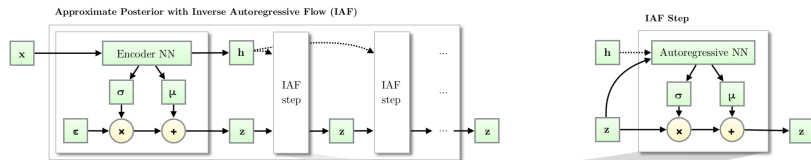
The second term in ELBO is **reverse** KL divergence with respect to $q(\mathbf{z}^*|\mathbf{x}, \phi, \lambda)$.

Flow-based VAE posterior

ELBO objective

$$\begin{aligned}\mathcal{L}(\phi, \theta, \lambda) &= \mathbb{E}_{q(\mathbf{z}^*|\mathbf{x}, \phi, \lambda)} [\log p(\mathbf{x}|\mathbf{z}^*, \theta) + \log p(\mathbf{z}^*) - \log q(\mathbf{z}^*|\mathbf{x}, \phi, \lambda)] = \\ &= \mathbb{E}_{q(\mathbf{z}^*|\mathbf{x}, \phi, \lambda)} \left[\log p(\mathbf{x}|\mathbf{z}^*, \theta) + \log p(\mathbf{z}^*) - \right. \\ &\quad \left. - \left(\log q(g(\mathbf{z}^*, \lambda)|\mathbf{x}, \phi) + \log |\det(\mathbf{J}_g)| \right) \right].\end{aligned}$$

- ▶ RealNVP with coupling layers.
- ▶ Inverse autoregressive flow (slow $f(\mathbf{z}, \lambda)$, fast $g(\mathbf{z}^*, \lambda)$).
- ▶ Is it OK to use AF for VAE posterior?



Flows-based VAE prior vs posterior

Theorem

VAE with the flow-based prior $p(\mathbf{z}|\boldsymbol{\lambda})$ for latent code \mathbf{z}^* is equivalent to VAE with flow-based posterior $q(\mathbf{z}^*|\mathbf{x}, \phi, \boldsymbol{\lambda})$ for latent code \mathbf{z} .

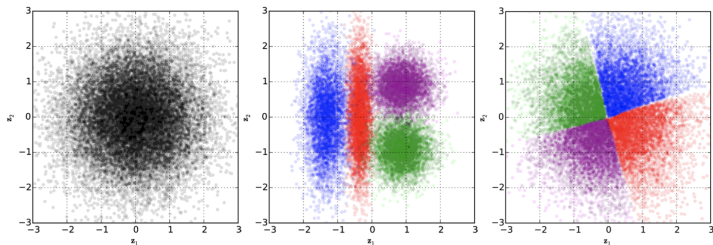
Proof

$$\begin{aligned}\mathcal{L}(\phi, \boldsymbol{\theta}, \boldsymbol{\lambda}) &= \mathbb{E}_{q(\mathbf{z}|\mathbf{x}, \phi)} \log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) - \underbrace{KL(q(\mathbf{z}|\mathbf{x}, \phi) || p(\mathbf{z}|\boldsymbol{\lambda}))}_{\text{flow-based prior}} \\ &= \mathbb{E}_{q(\mathbf{z}^*|\mathbf{x}, \phi, \boldsymbol{\lambda})} \log p(\mathbf{x}|f(\mathbf{z}^*, \boldsymbol{\lambda}), \boldsymbol{\theta}) - \underbrace{KL(q(\mathbf{z}^*|\mathbf{x}, \phi, \boldsymbol{\lambda}) || p(\mathbf{z}^*))}_{\text{flow-based posterior}}\end{aligned}$$

(Here we use Flow KL duality theorem from Lecture 5 and LOTUS)

- ▶ IAF posterior decoder path: $\mathbf{z} \sim p(\mathbf{z})$, $\mathbf{x} \sim p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$.
- ▶ AF prior decoder path: $\mathbf{z}^* \sim p(\mathbf{z}^*)$, $\mathbf{z} = f(\mathbf{z}^*, \boldsymbol{\lambda})$, $\mathbf{x} \sim p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$.

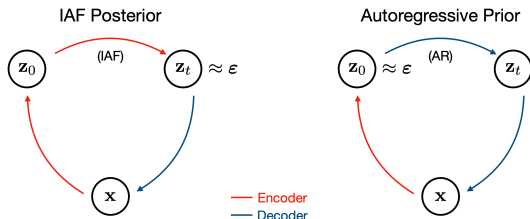
Flows-based VAE prior vs posterior



(a) Prior distribution

(b) Posteriors in standard VAE

(c) Posteriors in VAE with IAF



Kingma D. P. et al. *Improving Variational Inference with Inverse Autoregressive Flow*, 2016

image credit: <https://courses.cs.washington.edu/courses/cse599i/20au>

VAE limitations

- ▶ Poor generative distribution (decoder)

$$p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z}), \boldsymbol{\sigma}_{\boldsymbol{\theta}}^2(\mathbf{z})) \quad \text{or} \quad = \text{Softmax}(\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{z})).$$

- ▶ Loose lower bound

$$\log p(\mathbf{x}|\boldsymbol{\theta}) - \mathcal{L}(q, \boldsymbol{\theta}) = (?).$$

- ▶ Poor prior distribution

$$p(\mathbf{z}) = \mathcal{N}(0, \mathbf{I}).$$

- ▶ Poor variational posterior distribution (encoder)

$$q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi}) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}), \boldsymbol{\sigma}_{\boldsymbol{\phi}}^2(\mathbf{x})).$$

Outline

1. Flow-based VAE posterior vs flow-based VAE prior
2. Likelihood-free learning
3. Generative adversarial networks (GAN)

Likelihood based models

Is likelihood a good measure of model quality?

Poor likelihood
Great samples

$$p_1(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n \mathcal{N}(\mathbf{x}|\mathbf{x}_i, \epsilon \mathbf{I})$$

For small ϵ this model will generate samples with great quality, but likelihood will be very poor.

Great likelihood
Poor samples

$$p_2(\mathbf{x}) = 0.01p(\mathbf{x}) + 0.99p_{\text{noise}}(\mathbf{x})$$

$$\begin{aligned} \log [0.01p(\mathbf{x}) + 0.99p_{\text{noise}}(\mathbf{x})] &\geq \\ &\geq \log [0.01p(\mathbf{x})] = \log p(\mathbf{x}) - \log 100 \end{aligned}$$

Noisy irrelevant samples, but for high dimensions $\log p(\mathbf{x})$ becomes proportional to m .

Likelihood-free learning

- ▶ Likelihood is not a perfect quality measure for generative model.
- ▶ Likelihood could be intractable.

Where did we start

We would like to approximate true data distribution $\pi(\mathbf{x})$. Instead of searching true $\pi(\mathbf{x})$ over all probability distributions, learn function approximation $p(\mathbf{x}|\boldsymbol{\theta}) \approx \pi(\mathbf{x})$.

Imagine we have two sets of samples

- ▶ $\mathcal{S}_1 = \{\mathbf{x}_i\}_{i=1}^{n_1} \sim \pi(\mathbf{x})$ – real samples;
- ▶ $\mathcal{S}_2 = \{\mathbf{x}_i\}_{i=1}^{n_2} \sim p(\mathbf{x}|\boldsymbol{\theta})$ – generated (or fake) samples.

Two sample test

$$H_0 : \pi(\mathbf{x}) = p(\mathbf{x}|\boldsymbol{\theta}), \quad H_1 : \pi(\mathbf{x}) \neq p(\mathbf{x}|\boldsymbol{\theta})$$

Define test statistic $T(\mathcal{S}_1, \mathcal{S}_2)$. The test statistic is likelihood free. If $T(\mathcal{S}_1, \mathcal{S}_2) < \alpha$, then accept H_0 , else reject it.

Likelihood-free learning

Two sample test

$$H_0 : \pi(\mathbf{x}) = p(\mathbf{x}|\theta), \quad H_1 : \pi(\mathbf{x}) \neq p(\mathbf{x}|\theta)$$

Desired behaviour

- ▶ $p(\mathbf{x}|\theta)$ minimizes the value of test statistic $T(\mathcal{S}_1, \mathcal{S}_2)$.
- ▶ It is hard to find an appropriate test statistic in high dimensions. $T(\mathcal{S}_1, \mathcal{S}_2)$ could be learnable.

Generative adversarial network (GAN) objective

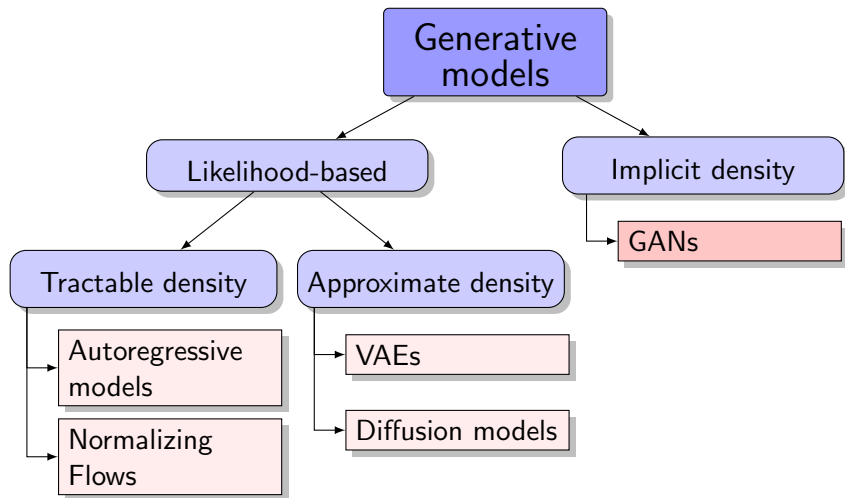
- ▶ **Generator:** generative model $\mathbf{x} = G(\mathbf{z})$, which makes generated sample more realistic. Here $\mathbf{z} \sim p(\mathbf{z})$, $\mathbf{x} \sim p(\mathbf{x}|\theta)$.
- ▶ **Discriminator:** a classifier $D(\mathbf{x}) \in [0, 1]$, which distinguishes real samples from generated samples.

$$\min_G \max_D [\mathbb{E}_{\pi(\mathbf{x})} \log D(\mathbf{x}) + \mathbb{E}_{p(\mathbf{x}|\theta)} \log(1 - D(\mathbf{x}))]$$

Outline

1. Flow-based VAE posterior vs flow-based VAE prior
2. Likelihood-free learning
3. Generative adversarial networks (GAN)

Generative models zoo



GAN optimality

Theorem

The minimax game

$$\min_G \max_D \underbrace{\left[\mathbb{E}_{\pi(\mathbf{x})} \log D(\mathbf{x}) + \mathbb{E}_{p(\mathbf{z})} \log(1 - D(G(\mathbf{z}))) \right]}_{V(G,D)}$$

has the global optimum $\pi(\mathbf{x}) = p(\mathbf{x}|\theta)$, in this case $D^*(\mathbf{x}) = 0.5$.

Proof (fixed G)

$$\begin{aligned} V(G, D) &= \mathbb{E}_{\pi(\mathbf{x})} \log D(\mathbf{x}) + \mathbb{E}_{p(\mathbf{x}|\theta)} \log(1 - D(\mathbf{x})) \\ &= \int \underbrace{[\pi(\mathbf{x}) \log D(\mathbf{x}) + p(\mathbf{x}|\theta) \log(1 - D(\mathbf{x}))]}_{y(D)} d\mathbf{x} \end{aligned}$$

$$\frac{dy(D)}{dD} = \frac{\pi(\mathbf{x})}{D(\mathbf{x})} - \frac{p(\mathbf{x}|\theta)}{1 - D(\mathbf{x})} = 0 \quad \Rightarrow \quad D^*(\mathbf{x}) = \frac{\pi(\mathbf{x})}{\pi(\mathbf{x}) + p(\mathbf{x}|\theta)}$$

GAN optimality

Proof continued (fixed $D = D^*$)

$$\begin{aligned} V(G, D^*) &= \mathbb{E}_{\pi(\mathbf{x})} \log \frac{\pi(\mathbf{x})}{\pi(\mathbf{x}) + p(\mathbf{x}|\theta)} + \mathbb{E}_{p(\mathbf{x}|\theta)} \log \frac{p(\mathbf{x}|\theta)}{\pi(\mathbf{x}) + p(\mathbf{x}|\theta)} \\ &= KL\left(\pi(\mathbf{x}) \parallel \frac{\pi(\mathbf{x}) + p(\mathbf{x}|\theta)}{2}\right) + KL\left(p(\mathbf{x}|\theta) \parallel \frac{\pi(\mathbf{x}) + p(\mathbf{x}|\theta)}{2}\right) - 2 \log 2 \\ &= 2JSD(\pi(\mathbf{x}) \parallel p(\mathbf{x}|\theta)) - 2 \log 2. \end{aligned}$$

Jensen-Shannon divergence (symmetric KL divergence)

$$JSD(\pi(\mathbf{x}) \parallel p(\mathbf{x}|\theta)) = \frac{1}{2} \left[KL\left(\pi(\mathbf{x}) \parallel \frac{\pi(\mathbf{x}) + p(\mathbf{x}|\theta)}{2}\right) + KL\left(p(\mathbf{x}|\theta) \parallel \frac{\pi(\mathbf{x}) + p(\mathbf{x}|\theta)}{2}\right) \right]$$

Could be used as a distance measure!

$$V(G^*, D^*) = -2 \log 2, \quad \pi(\mathbf{x}) = p(\mathbf{x}|\theta), \quad D^*(\mathbf{x}) = 0.5.$$

GAN optimality

Theorem

The minimax game

$$\min_G \max_D \underbrace{\left[\mathbb{E}_{\pi(\mathbf{x})} \log D(\mathbf{x}) + \mathbb{E}_{p(\mathbf{z})} \log(1 - D(G(\mathbf{z}))) \right]}_{V(G,D)}$$

has the global optimum $\pi(\mathbf{x}) = p(\mathbf{x}|\theta)$, in this case $D^*(\mathbf{x}) = 0.5$.

Expectations

If the generator could be **any** function and the discriminator is **optimal** at every step, then the generator is **guaranteed to converge** to the data distribution.

Reality

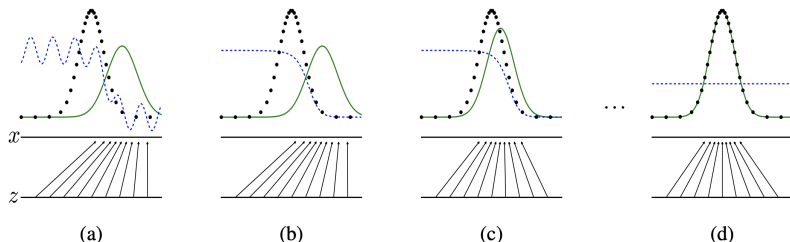
- ▶ Generator updates are made in parameter space, discriminator is not optimal at every step.
- ▶ Generator and discriminator loss keeps oscillating during GAN training.

GAN

Let further assume that generator and discriminator are parametric models: $D(\mathbf{x}, \phi)$ and $G(\mathbf{z}, \theta)$.

Objective

$$\min_{\theta} \max_{\phi} [\mathbb{E}_{\pi(\mathbf{x})} \log D(\mathbf{x}, \phi) + \mathbb{E}_{p(\mathbf{z})} \log(1 - D(G(\mathbf{z}, \theta), \phi))]$$



- ▶ $\mathbf{z} \sim p(\mathbf{z})$ is a latent variable.
- ▶ $p(\mathbf{x}|\mathbf{z}, \theta) = \delta(\mathbf{x} - G(\mathbf{z}, \theta))$ is deterministic decoder (like NF).
- ▶ We do not have encoder at all.

Summary

- ▶ It is possible to use flows in VAE prior and posterior. This is almost the same.
- ▶ Likelihood is not a perfect criteria to measure quality of generative model.
- ▶ Adversarial learning suggests to solve minimax problem to match the distributions.
- ▶ GAN tries to optimize Jensen-Shannon divergence (in theory).