A survey of traditional and state-of-the-art Generative Models

Richard Xu

January 1, 2026

1 Abstract

Variational Bayes has been one of the cornerstones of generative models in the deep learning era. They have been around for many years after the seminal VAE paper [?]. In this tutorial, I will try to explain variational models, and more importantly, I will show detailed derivations that not only make this tutorial self-contained, but also help machine learning beginners understand how these equations are derived. I'll also show some recent extensions to VAE, including Importance Weighted Autoencoders, adversarial variational bayes, Variational Autoencoders, Relation to VAE-GAN, Gaussian Mixture Model Variational Inference, Stick-breaking VAE and Normalized Flow, and of course, the last but not least, the noise diffusion model

2 Introduction

Variational Bayesian methods have been around for decades. Traditionally, it has been used to approximate complex distributions by the most common mean-field approximation.

2.1 Maximum Likelihood Estimation

Firstly, let's have a look at the good old maximum likelihood estimation:

$$\hat{\theta} = \arg\max_{\theta} \sum_{i=1}^{n} \log p_{\theta}(\mathbf{x}_i) \tag{1}$$

as many models are defined in terms of their latent variables z_i, then we must specify $p(x_i)$ as a marginal distribution:

$$\begin{aligned} \hat{\theta} &= \arg\max_{\theta} \sum_{i=1}^{n} \log \int_{z_i} p_{\theta}(\mathbf{x}_i, \mathbf{z}_i) \\ &= \arg\max_{\theta} \sum_{i=1}^{n} \log \int_{z_i} p_{\theta}(\mathbf{x}_i|\mathbf{z}_i) p(\mathbf{z}_i) \end{aligned} \tag{2}$$

the last term can be approximated by:

$$\approx \arg\max_{\theta} \sum_{i=1}^{n} \log \sum_{j=1}^{n_S} p_{\theta}(\mathbf{x}_i|\mathbf{z}_i^{(j)}) \qquad \mathbf{z}_i^{(j)} \sim p(\mathbf{z}) \tag{3}$$

However, the reason for not using it is because sampling from $p(\mathbf{z})$ may be inefficient, i.e. the corresponding $p(\mathbf{x}|\mathbf{z})$ may receive a very low probability. So ideally we want to use $p(\mathbf{z}|\mathbf{x})$ to generate samples. Of course, this is easier said than done. So the intuition here is for us to approximate this integal by sampling from a "simpler" distribution $q(\mathbf{z}|\mathbf{x})$ that approximates $p(\mathbf{z}|\mathbf{x})$.

3 let's talk about $q(\mathbf{z}|\mathbf{x}) \approx p(\mathbf{z}|\mathbf{x})$

dropping index i, we want to have a good estimator of $\log p(x|\theta)$, we know:

$$\begin{aligned}
\log p_\theta(x) &= \log \int_z p_\theta(x, z) \\
&= \log \int_z \frac{p_\theta(x, z|\theta)}{q_\phi(z|x)} q_\phi(z|x) \\
&= \log \left[\mathbb{E}_{z\sim q_\phi(z|x)} \left(\frac{p_\theta(x, z|\theta)}{q_\phi(z|x)} \right) \right]
\end{aligned} \tag{4}$$

in the above, $\log(\mathbb{E}[.])$ is not that useful, so we maximize its lower-bound, i.e., ELBO (Let's wait to see that the un-useful expression is actually the basis of IWAE)

$$\begin{aligned}
&\geq \mathbb{E}_{z\sim q_\phi(z|x)} \left[\log \left(\frac{p_\theta(x, z|\theta)}{q_\phi(z|x)} \right) \right] \qquad \text{by Jensen's inequality} \\
&= \mathbb{E}_{z\sim q_\phi(z|x)} \left[\log(p_\theta(x, z|\theta) \right] - \mathbb{E}_{z\sim q_\phi(z|x)} \left[\log(q_\phi(z|x) \right] \\
&= \text{ELBO}(\phi) \\
&= \text{ELBO}(\phi, \theta)
\end{aligned} \tag{5}$$

The advantage of ELBO is it has no "model conditional" $p(z|x) = \frac{p(z,x)}{\int_z p(x,z)}$ (it's hard to obtain). It can be approximated by monte-carlo, using integral of k samples, where samples are from "proposal conditional" $q_\phi(z|x)$

3.1 monte-carlo approximation

$$\begin{aligned}
\text{ELBO}(\phi) &= \mathbb{E}_{z\sim q_\phi(z|x)} \left[\log \left(\frac{p_\theta(x, z)}{q_\phi(z|x)} \right) \right] \\
\implies \text{ELBO}_k(\phi) &= \frac{1}{k} \sum_{j=1}^{k} \left[\log \left(\frac{p_\theta(x, z^j)}{q_\phi(z^j|x)} \right) \right] \\
&\text{where } z^j \sim q_\phi(z|x)
\end{aligned} \tag{6}$$

note that $\text{ELBO}_k(\phi)$ is a k samples approximation of Monte-Carlo expectation.
By LLN:

$$\lim_{k\to\infty} \text{ELBO}_k(\phi) = \text{ELBO}(\phi) \tag{7}$$

4 Evidence lower bound (ELBO)

4.1 Expression ELOB

knowing:

$$\begin{aligned}
\text{ELBO}(\phi) &= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})} \left[\log \left(\frac{p(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} p_\theta(\mathbf{x}|\mathbf{z}) \right) \right] \\
&= \int \log \left(\frac{p(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} p_\theta(\mathbf{x}|\mathbf{z}) \right) q_\phi(\mathbf{z}|\mathbf{x}) \mathbf{Dz}
\end{aligned} \tag{8}$$

there are two main ways of expressing ELBO in literature:

- split one

$$\begin{aligned}
&= \int \log p_\theta(\mathbf{x}|\mathbf{z}) q_\phi(\mathbf{z}|\mathbf{x})\mathbf{Dz} + \int \log\left(\frac{p(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right) q_\phi(\mathbf{z}|\mathbf{x})\mathbf{Dz} \\
&= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] - \int \log\left(\frac{q_\phi(\mathbf{z}|\mathbf{x})}{p(\mathbf{z})}\right) q_\phi(\mathbf{z}|\mathbf{x})\mathbf{Dz} \\
&= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] - \mathbb{KL}\left[q_\phi(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z})\right]
\end{aligned} \tag{9}$$

the advantage is that we can express it in terms of the KL. Let's look at split one, we can view the aim of $\text{ELBO}_{(\theta,\phi)}$ to be finding alignment between $q_\phi(\mathbf{z}|\mathbf{x})$ with the posterior $p_\theta(\mathbf{z}|\mathbf{x})$:

$$\text{ELBO}_{(\theta,\phi)} = \underbrace{\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right]}_{\text{alignment with likelihood} p_\theta(\mathbf{x}|\mathbf{z})} + \underbrace{-\mathbb{KL}\left[q_\phi(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z})\right]}_{\text{alignment with prior } p(z)} \tag{10}$$

Therefore, we can see that $q_\phi(\mathbf{z}|\mathbf{x})$ is the balance of the two alignments. This will be illustrated again the VAE-GAN

- split two

$$\begin{aligned}
&= \int \log p_\theta(\mathbf{x}|\mathbf{z}) p(\mathbf{z}) q_\phi(\mathbf{z}|\mathbf{x})\mathbf{Dz} + \int \log\left(\frac{1}{q_\phi(\mathbf{z}|\mathbf{x})}\right) q_\phi(\mathbf{z}|\mathbf{x})\mathbf{Dz} \\
&= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x},\mathbf{z})\right] - \int \log q_\phi(\mathbf{z}|\mathbf{x}) q_\phi(\mathbf{z}|\mathbf{x})\mathbf{Dz} \\
&= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x},\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})\right]
\end{aligned} \tag{11}$$

- split three

  There is no reason $q(\mathbf{z}|\mathbf{x})$ be used. $p(\mathbf{z}|\mathbf{x})$ may also be approximated by $q(\mathbf{z})$ as well. In this case, we have:

$$\text{ELBO}_{(\theta,\phi)} = \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z})}\left[\log p_\theta(\mathbf{x},\mathbf{z}) - \log q_\phi(\mathbf{z})\right] \tag{12}$$

We will document which split papers are using of the following literatures:

4.2 Purpose of Variational Bayes using ELBO

4.2.1 to approximate $p_\theta(z|x)$

We already stated that $p(z|x) = \frac{p(z,x)}{\int_z p(x,z)}$ is difficult to compute. Jensen's inequality did not explicitly stating what is actually missing between $\log p_\theta(x)$ and $\text{ELBO}(\phi)$, so the extract expression is:

$$\begin{aligned}
\log(p_\theta(x)) &= \log(p_\theta(x,z)) - \log(p_\theta(z|x)) \\
&= \log\left(\frac{p_\theta(x,z)}{q_\phi(z|x)}\right) - \log\left(\frac{p_\theta(z|x)}{q_\phi(z|x)}\right) \\
&= \underbrace{\int q_\phi(z|x)\log\left(\frac{p_\theta(x,z)}{q_\phi(z|x)}\right)\mathbf{D}z}_{\text{ELBO}(\phi)} + \underbrace{\left(-\int q_\phi(z|x)\log\left(\frac{p_\theta(z|x)}{q_\phi(z|x)}\right)\mathbf{D}z\right)}_{\mathbb{KL}(q_\phi(z|x))\|p_\theta(z|x)} \\
&= \text{ELBO}(\phi) + \mathbb{KL}(p_\theta(z|x)\|q_\phi(z|x))
\end{aligned} \tag{13}$$

Maximizing ELBO has the same effect as minimize KL, which means VB allow $q_\phi(z|x)$ to approximate $p_\theta(z|x)$.

4.2.2 perform Maximum Likelihood

to perform MLE:

$$
\begin{aligned}
\hat{\theta} &= \arg\max_{\theta} \sum_{i=1}^{n} \log p_\theta(x_i) \\
&\approx \arg\max_{\theta,\phi} \sum_{i=1}^{n} \mathrm{ELBO}(\phi) \quad \text{approximated by lower-bound} \\
&\approx \arg\max_{\theta,\phi} \sum_{i=1}^{n} \mathrm{ELBO}_k(\phi) \quad \text{further approximated by MC integral} \\
&= \arg\max_{\theta,\phi} \sum_{i=1}^{n} \frac{1}{k} \sum_{j=1}^{k} \left[\log \left(\frac{p_\theta(x, z^j)}{q_\phi(z^j|x)} \right) \right] \quad z^j \sim q_\phi(z^j|x) \\
&= \arg\max_{\theta,\phi} \sum_{i=1}^{n} \sum_{j=1}^{k} \left[\log \left(\frac{p_\theta(x, z^j)}{q_\phi(z^j|x)} \right) \right] \quad z^j \sim q_\phi(z^j|x)
\end{aligned}
\tag{14}
$$

5 Importance weighted auto-encoders

5.1 IWAE$_k$

this section is to explain [?]. Looking at Eq.(??), we know the following identity:

$$\log p_\theta(x) = \log \left[\mathbb{E}_{z \sim q_\phi(z|x)} \left(\frac{p_\theta(x, z|\theta)}{q_\phi(z|x)} \right) \right]$$

the goal is to approximate the above; however, let us first define an expression:

$$\widehat{\text{IWAE}}_k = \log \left[\frac{1}{k} \sum_{j=1}^{k} \frac{p_\theta(x|z^{(j)}) p(z^{(j)})}{q_\phi(z^{(j)}|x)} \right] \tag{15}$$

Note that although $\widehat{\text{IWAE}}_k$ looks like ELBO$_k(\phi)$, $\widehat{\text{IWAE}}_k$ was merely an expression inside the monte-carlo integral. Itself is a random variable, it's not an approximation to expectation. In fact, we need to "arm" it by putting this expression inside an Expectation, to make it functional:

$$\begin{aligned}
\text{IWAE}_k &= \mathbb{E}_{\left\{ z^{(j)} \sim q_\phi(z|x) \right\}_{j=1}^{k}} \left[\widehat{\text{IWAE}}_k \right] \\
&= \mathbb{E}_{\left\{ z^{(j)} \sim q_\phi(z|x) \right\}_{j=1}^{k}} \left[\log \left[\frac{1}{k} \sum_{j=1}^{k} \frac{p_\theta(x|z^{(j)}) p(z^{(j)})}{q_\phi(z^{(j)}|x)} \right] \right] \\
&= \int_{z^{(1)}} \cdots \int_{z^{(k)}} \log \left[\frac{1}{k} \sum_{j=1}^{k} \frac{p_\theta(x|z^{(j)}) p(z^{(j)})}{q_\phi(z^{(j)}|x)} \right] \prod_{j=1}^{k} q_\phi(z^{(j)}|x)
\end{aligned} \tag{16}$$

in summary, IWAE$_k$ itself is an expectation of the expression $\widehat{\text{IWAE}}_k$. So if one is to approximate IWAE$_k$, one must sample, sample-set $\{z^{(1)}, \ldots, z^{(k)}\}$ multiple say n times.

Now looking at what happens when we have $k = 1$ and $k = \infty$:

5.2 IWAE$_1$

what if we have $k = 1$, by looking Eq.(??), we have:

$$\begin{aligned}
\text{IWAE}_1 &= \mathbb{E}_{\mathbf{z}^{(1)} \sim q_\phi(z|x)} \left[\widehat{\text{IWAE}}_1 \right] \\
&= \mathbb{E}_{z^{(1)} \sim q_\phi(z|x)} \left[\log \left[\frac{p_\theta(x|z^{(1)}) p(z^{(1)})}{q_\phi(z^{(1)}|x)} \right] \right] \\
&= \mathbb{E}_{z \sim q_\phi(z|x)} \left[\log \left[\frac{p_\theta(x|z) p(z)}{q_\phi(z|x)} \right] \right] \quad \text{drop index} \\
&= \text{ELBO}(\phi)
\end{aligned} \tag{17}$$

5.3 IWAE$_\infty$

in fact, there is no need to explicitly proving IWAE$_\infty$, we can use the fact that $\forall k$:

$$\begin{aligned}
\mathrm{IWAE}_k &= \mathbb{E}_{\left\{z^{(j)} \sim q_\phi(z|x)\right\}_{j=1}^k}\left[\log\left[\left(\frac{1}{k}\sum_{j=1}^k \frac{p_\theta(x|z^{(j)})p(z^{(j)})}{q_\phi(z^{(j)}|x)}\right)\right]\right] \\
&\leq \log\left(\mathbb{E}_{\left\{z^{(j)} \sim q_\phi(z|x)\right\}_{j=1}^k}\left[\left(\frac{1}{k}\sum_{j=1}^k \frac{p_\theta(x|z^{(j)})p(z^{(j)})}{q_\phi(z^{(j)}|x)}\right)\right]\right) \\
&= \log\frac{1}{k}\int_{z^{(2)}}\cdots\int_{z^{(k)}}\left(\sum_{j=2}^k \frac{p_\theta(x|z^{(j)})p(z^{(j)})}{q_\phi(z^{(j)}|x)} + \underbrace{\int_{z^{(1)}} \frac{p_\theta(x|z^{(1)})p(z^{(1)})}{q_\phi(z^{(1)}|x)} q_\phi(z^{(1)}|x)}_{=p_\theta(x)}\right)\prod_{j=2}^k q_\phi(z^{(j)}|x) \\
&= \log\frac{kp_\theta(x)}{k} \qquad \because q_\phi(z^{(1)}|x) \quad \text{cancels out in numerator and denominator} \\
&= \log p_\theta(x)
\end{aligned} \tag{18}$$

since the upper-bound of $\mathrm{IWAE}_k = p_\theta(x)\ \forall k$, then, by proving section(??), we can deduce:

$$\mathrm{IWAE}_\infty = p_\theta(x) \tag{19}$$

5.4 Tighter bound

it can be proven that:

$$\mathrm{ELBO} = \mathrm{IWAE}_1 \leq \mathrm{IWAE}_2 \leq \cdots \leq \mathrm{IWAE}_\infty = \log p_\theta(x) \tag{20}$$

5.4.1 proof of why $k \geq m \implies \mathrm{IWAE}_k \geq \mathrm{IWAE}_m$

First, intuitively, the following is true:

$$\mathbb{E}_{I=\{j_1,\ldots,j_m\}}\left[\frac{w_{j_1}+\cdots+w_{j_m}}{m}\right] = \frac{w_1+\cdots+w_k}{k} \tag{21}$$

In words, the “average of a uniformly generated sub-set equal the average of a full-set”.

More formally, it means is that given $m \leq k$, you are selecting uniformly a subset of m elements from k available data. Then, instead of perform true average on k-element data, you are performing an average on the m-element subset.

In Eq.(??), it says the expectation of the “average of uniformly-drawn sub-set”, equal the value of true average. Note the above should not work when $m > k$. Also note that the original set $\{w_1, \ldots w_k\}$ does not need to be stochastic.

Now we apply the above lemma to IWAE_k equation:

$$\begin{aligned}
\mathrm{IWAE}_k &= \mathbb{E}_{\left\{z^{(j)} \sim q_\phi(z|x)\right\}_{j=1}^k}\left[\log\left[\underbrace{\frac{1}{k}\sum_{j=1}^k \frac{p_\theta(x|z^{(j)})p(z^{(j)})}{q_\phi(z^{(j)}|x)}}_{\text{true average}}\right]\right] \\
&= \mathbb{E}_{\left\{z^{(j)} \sim q_\phi(z|x)\right\}_{j=1}^k}\left[\log\left[\underbrace{\mathbb{E}_{I=\{j_1,\ldots,j_m\}}\left[\frac{1}{m}\sum_{t=1}^m \frac{p_\theta(x|z^{(j_t)})p(z^{(j_t)})}{q_\phi(z^{(j_t)}|x)}\right]}_{\text{expectation of “average of uniformly-drawn sub-set”}}\right]\right] \quad \text{apply Eq.(??)} \\
&\geq \mathbb{E}_{\left\{z^{(j)} \sim q_\phi(z|x)\right\}_{j=1}^k}\left[\mathbb{E}_{I=\{j_1,\ldots,j_m\}}\left[\log\left[\frac{1}{m}\sum_{t=1}^m \frac{p_\theta(x|z^{(j_t)})p(z^{(j_t)})}{q_\phi(z^{(j_t)}|x)}\right]\right]\right] \quad \text{by Jensen's inequality}
\end{aligned} \tag{22}$$

Now looking at $\mathbb{E}_{\left\{z^{(j)}\sim q_\phi(z|x)\right\}_{j=1}^{k}}\left[\mathbb{E}_{I=\{j_1,\dots,j_m\}}[.]\right]$, these two nested expectation is computed over the probability, by first selecting k i.i.d samples from $q_\phi(z|x)$, and then select m subset from it. (However, the above may possibly result duplicating values of $z^{(j)}$)

So the two integral can combine together:

$$\begin{aligned}
&= \mathbb{E}_{\left\{z^{(j_t)}\sim q_\phi(z|x)\right\}_{t=1}^{m}}\left[\log\left[\frac{1}{m}\sum_{t=1}^{m}\frac{p_\theta(x|z^{(j_t)})p(z^{(j_t)})}{q_\phi(z^{(j_t)}|x)}\right]\right] \\
&= \mathbb{E}_{\left\{z^{(j)}\sim q_\phi(z|x)\right\}_{j=1}^{m}}\left[\log\left[\frac{1}{m}\sum_{j=1}^{m}\frac{p_\theta(x|z^{(j)})p(z^{(j)})}{q_\phi(z^{(j)}|x)}\right]\right] \quad \text{drop index of } t \\
&= \mathrm{IWAE}_m
\end{aligned} \tag{23}$$

we have proved $k \geq m \implies \mathrm{IWAE}_k \geq \mathrm{IWAE}_m$

6 Variational Auto Encoder

it uses the split one of ELBO derivation:

$$\text{ELBO}_{(\theta,\phi)} = \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\big[\log p_\theta(\mathbf{x}|\mathbf{z})\big] - \mathbb{KL}\big[q_\phi(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z})\big] \tag{24}$$

note that if we use split one:

$$\text{ELBO}_{(\theta,\phi)} = \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\big[\log p_\theta(\mathbf{x},\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})\big] \tag{25}$$

although it's the same thing, but we cannot have a nice KL interpretation.

6.1 VAE algorithm

during each iteration of gradient descend, the gradient is computed as:

$$\begin{aligned}
&\text{get mini-batch } \{\mathbf{x}\}\\
&\mathbf{z} \sim q_\phi(\cdot|\mathbf{x})\\
&\qquad \text{re-parameterization:}\\
&\qquad\quad \epsilon \sim \mathcal{N}(0, \mathbf{I})\\
&\qquad\quad \mathbf{z} = \text{Encoder}_\phi(\mathbf{x}, \epsilon)\\
&\qquad\qquad = \mu_\phi(\mathbf{x}) + \Sigma_\phi(\mathbf{x}) \times \epsilon\\
&\triangle\theta \propto -\nabla_\theta \text{ELBO}_{(\theta,\phi)}(\mathbf{x}, \mathbf{z})\\
&\triangle\phi = -\nabla_\phi \text{ELBO}_{(\theta,\phi)}(\mathbf{x}, \mathbf{z})
\end{aligned} \tag{26}$$

if we have just one z_i to correspond to a x_i, then we basically draw a single sample from each distribution $q_\phi(z_i|x_i)$ to approximate ELBO including the KL part

6.1.1 evaluating $\log p_\theta(\mathbf{x}|\mathbf{z})$ through reconstruction loss

under traditional variational inference $\log p_\theta(\mathbf{x}|\mathbf{z})$ is evaluable.

However, in the typical settings of VAE, for example where $\mathbf{x}$ is images, $\log p_\theta(\mathbf{x}|\mathbf{z})$ can not be evaluated. This is of course where the backward decoder becomes helpful to evaluate it, i.e:

$$\hat{\mathbf{x}} = \text{Decoder}_\theta(\mathbf{z}) \tag{27}$$

therefore:

$$\begin{aligned}
p_\theta(\mathbf{x}|\mathbf{z}) &\equiv p\big(\mathbf{x} \mid \text{Decoder}_\theta(\mathbf{z})\big) \qquad \text{by VAE}\\
&\propto \exp\big(-d(\mathbf{x},\ \hat{\mathbf{x}} = \text{Decoder}_\theta(\mathbf{z}))\big)\\
&= \exp\big(-d(\mathbf{x},\ \hat{\mathbf{x}})\big)\\
\implies \log p_\theta(\mathbf{x}|\mathbf{z}) &= -d(\mathbf{x},\ \hat{\mathbf{x}})
\end{aligned} \tag{28}$$

making the first term just the average reconstruction loss, we may rewrite ELOB again for VAE:

$$\begin{aligned}
\text{ELBO}_{(\theta,\phi)} &= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\big[\log p_\theta(\mathbf{x}|\mathbf{z})\big] - \mathbb{KL}\big[q_\phi(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z})\big]\\
&= \mathbb{E}_{\mathbf{z}\sim \text{Encoder}_\phi(\mathbf{x})}\big[-d\big(\mathbf{x}, \text{Decoder}_\theta(\mathbf{z})\big)\big] - \mathbb{KL}\big[\text{Encoder}_\phi(\mathbf{x}) \| p(\mathbf{z})\big]
\end{aligned} \tag{29}$$

6.2 some points to note

- $\text{Encoder}_\phi(\mathbf{x})$ is actually a re-parameterized probability density function $q_\phi(\mathbf{z}|\mathbf{x})$, whereas the $\text{Decoder}_\theta(\mathbf{z})$ is only part of the probability of $p_\theta(\mathbf{x}|\mathbf{z})$
- $p(\mathbf{z})$ are to evaluate $\mathbb{KL}\big[q_\phi(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})\big]$, it is not used for sampling. Therefore, in theory, one may use very complex $p(\mathbf{z})$ form, as long as it's evaluatable
- $\text{Encoder}_\phi(\mathbf{x}, \epsilon)$ is a single inference network

6.3 relationship with VAE-GAN

6.3.1 why VAE generate blur image

Due to the claim that VAE's decoder (used for reconstruction) may not be as effective as GAN's generator (Gen^{GAN}). A popular explanation of why VAE may generate blur image: one explanation is that if reconstruction loss was $d\big(\mathbf{x}, \text{Decoder}_\theta(\mathbf{z})\big)$, and imagine $\text{Decoder}_\theta(\mathbf{z})$ is a blur version of $\mathbf{x}$, then, their VAE-reconstruction loss is in fact small (They can be "content-wise" similar, but "style-wise" different - think about an image and its Gaussian smooth version can have small L_2 loss). GAN on the other hand has no individual reconstructions. Therefore, it is looking for global distribution similarity (style loss)

6.3.2 VAE-GAN loss

Therefore, we can do the following, and we also change the objective to minimization instead of maximization.

By letting $\text{Des}_l^{\text{GAN}}$ to be the l^{th} layer of Discriminator (therefore, $\text{Des}_l^{\text{GAN}} \in \mathbb{R}^{m_l}$ whereas $\text{Des}^{\text{GAN}} \in (0, \ldots 1)$). Of course the GAN objective will be able to train Gen^{GAN}, and Des^{GAN}

$$\begin{aligned}
-\text{ELBO}_{(\theta,\phi)} + \text{GAN} &= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\big[-\log p_\theta(\mathbf{x}|\mathbf{z})\big] + \mathbb{KL}\big[\text{Encoder}_\phi(\mathbf{x})\|p(\mathbf{z})\big] + \text{GAN} \\
&= \underbrace{\mathbb{E}_{\mathbf{z}\sim \text{Encoder}_\phi(\mathbf{x})}\big[-d\big(\mathbf{x}, \text{Decoder}_\theta(\mathbf{z})\big)\big]}_{\text{replace}} + \underbrace{\mathbb{KL}\big[q_\phi(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})\big]}_{\text{keep alignment with prior}} + \text{GAN} \\
&= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\Big[-\log p_\theta\Big(\text{Des}_l^{\text{GAN}}(\mathbf{x}) \;\Big|\; \text{Des}_l^{\text{GAN}}(\text{Decoder}_\theta(\mathbf{z}))\Big)\Big] + \mathbb{KL}\big[\text{Encoder}_\phi(\mathbf{x})\|p(\mathbf{z})\big] + \text{GAN} \\
&= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\Big[-\log \mathcal{N}\Big(\text{Des}_l^{\text{GAN}}(\mathbf{x}) \;;\; \text{Des}_l^{\text{GAN}}(\text{Decoder}_\theta(\mathbf{z}))\Big)\Big] + \mathbb{KL}\big[\text{Encoder}_\phi(\mathbf{x})\|p(\mathbf{z})\big] + \text{GAN}
\end{aligned} \tag{30}$$

Whereas in VAE-GAN-reconstruction loss $d\Big(\text{Des}_l^{\text{GAN}}(\mathbf{x}) \;, \text{Des}_l^{\text{GAN}}(\text{Decoder}_\theta(\mathbf{z}))\Big)$ needs to ensure they are "style-wise" similar features. Putting both, real data $\mathbf{x}$ and $\tilde{x}$ from decoder $\tilde{x} = \text{Decoder}_\theta(\mathbf{z})$ into Discriminator.

6.3.3 notes on VAE-GAN

there could be many different implementation to the above. for example:

- if one were to replace
  $\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\Big[-\log \mathcal{N}\Big(\text{Des}_l^{\text{GAN}}(\mathbf{x}) \;;\; \text{Des}_l^{\text{GAN}}(\text{Decoder}_\theta(\mathbf{z}))\Big)\Big]$ to also update Des^{GAN}
  with

  $$\mathcal{N}\Big(\text{Des}_l^{\text{GAN}}(\mathbf{x}) \;;\; \text{Des}_l^{\text{GAN}}(\text{Decoder}_\theta(\mathbf{z}))\Big) \rightarrow \mathcal{N}\Big(\text{Des}_l^{\text{GAN}}(\mathbf{x}) \;;\; \text{Des}_l^{\text{GAN}}(\text{Gen}^{\text{GAN}}(\mathbf{z}))\Big) \tag{31}$$

  it makes no sense, as we are not learning decoder parameter.

6.4 KL between two Gaussian distributions

Last piece of puzzle is that, VAE objective function requires to compute KL between two Gaussians, let's have a look at their forms:

6.4.1 generallized for to compute $\mathbb{KL}\big(\mathcal{N}(\mu_1,\Sigma_1)\|\mathcal{N}(\mu_2,\Sigma_2)\big)$

$$\begin{aligned}
\mathbb{KL} &= \int_x \left[\frac{1}{2}\log\frac{|\Sigma_2|}{|\Sigma_1|} - \frac{1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1) + \frac{1}{2}(x-\mu_2)^T\Sigma_2^{-1}(x-\mu_2)\right] \times p(x)\mathbf{D}x \\
&= \frac{1}{2}\log\frac{|\Sigma_2|}{|\Sigma_1|} - \frac{1}{2}\text{tr}\ \left\{\mathbb{E}[(x-\mu_1)(x-\mu_1)^T]\ \Sigma_1^{-1}\right\} + \frac{1}{2}\mathbb{E}[(x-\mu_2)^T\Sigma_2^{-1}(x-\mu_2)] \\
&= \frac{1}{2}\log\frac{|\Sigma_2|}{|\Sigma_1|} - \frac{1}{2}\text{tr}\ \{I_d\} + \frac{1}{2}(\mu_1-\mu_2)^T\Sigma_2^{-1}(\mu_1-\mu_2) + \frac{1}{2}\text{tr}\{\Sigma_2^{-1}\Sigma_1\} \\
&= \frac{1}{2}\left[\log\frac{|\Sigma_2|}{|\Sigma_1|} - d + \text{tr}\{\Sigma_2^{-1}\Sigma_1\} + (\mu_2-\mu_1)^T\Sigma_2^{-1}(\mu_2-\mu_1)\right]
\end{aligned} \tag{32}$$

substitute $\bar{\mu}_1 = [\mu_1,\ldots,\mu_K]^\top$ and $\Sigma_1 = \text{diag}(\sigma_1,\ldots,\sigma_K)$, $\qquad \mu_2 = \mathbf{0}$ and $\Sigma_2 = \mathbf{I}$:

$$\begin{aligned}
\text{KL} &= \frac{1}{2}\ \Big(\text{tr}(\Sigma_1) + \bar{\mu}_1^T\bar{\mu}_1 - K - \log\ \det(\Sigma_1)\Big) \\
&= \frac{1}{2}\ \left(\sum_k \sigma_k^2 + \sum_k \mu_k^2 - \sum_k 1 - \log\prod_k \sigma_k^2\right) \\
&= \frac{1}{2}\sum_k \left(\sigma_k^2 + \mu_k^2 - 1 - \log\ \sigma_k^2\right)
\end{aligned} \tag{33}$$

6.4.2 when $p(x_1,x_2) = p(x_1)p(x_2)$ and $q(x_1,x_2) = q(x_1)q(x_2)$ (1)

$$\begin{aligned}
&\mathbb{KL}(p,q) = -\left(\int p(x_1)\log q(x_1)\mathbf{D}x_1 - \int p(x_1)\log p(x_1)\mathbf{D}x_1\right) \\
&\implies \mathbb{KL}(p(x_1)p(x_2)\|q(x_1)q(x_2)) \\
&= -\left(\int_{x_1}\int_{x_2} p(x_1)p(x_2)\big[\log q(x_1) + \log q(x_2)\big]\mathbf{D}x_1 - p(x_1)p(x_2)\big[\log p(x_1) + \log p(x_2)\big]\mathbf{D}x_1\right) \\
&= -\left(\int_{x_1}\int_{x_2} \big[p(x_1)p(x_2)\log q(x_1) + p(x_1)p(x_2)\log q(x_2) - p(x_1)p(x_2)\log p(x_1) - p(x_1)p(x_2)\log p(x_2)\big]\mathbf{D}x_1\right) \\
&= -\left(\int_{x_1}\int_{x_2} p(x_1)p(x_2)\log q(x_1) + \int_{x_1}\int_{x_2} p(x_1)p(x_2)\log q(x_2) - \int_{x_1}\int_{x_2} p(x_1)p(x_2)\log p(x_1) - \int_{x_1}\int_{x_2} p(x_1)p(x_2)\log p(x_2)\mathbf{D}x_1\right. \\
&= -\left(\int_{x_1} p(x_1)\log q(x_1)\int_{x_2} p(x_2) + \int_{x_1} p(x_1)\int_{x_2} p(x_2)\log q(x_2) - \int_{x_1} p(x_1)\log p(x_1)\int_{x_2} p(x_2) - \int_{x_1} p(x_1)\int_{x_2} p(x_2)\log p(x_2)\right) \\
&= -\left(\int_{x_1} p(x_1)\log q(x_1) + \int_{x_2} p(x_2)\log q(x_2) - \int_{x_1} p(x_1)\log p(x_1) - \int_{x_2} p(x_2)\log p(x_2)\right) \\
&= -\left(\int_{x_1} p(x_1)\log q(x_1) - \int_{x_1} p(x_1)\log p(x_1)\right) - \left(\int_{x_2} p(x_2)\log q(x_2) - \int_{x_2} p(x_2)\log p(x_2)\right) \\
&= \mathbb{KL}(p(x_1)\|q(x_1)) + \mathbb{KL}(p(x_2)\|q(x_2))
\end{aligned} \tag{34}$$

therefore,

$$\begin{aligned} &\mathbb{KL}(p(x_1)p(x_2)\|q(x_1)q(x_2)) = \mathbb{KL}(p(x_1)\|q(x_1)) + \mathbb{KL}(p(x_2)\|q(x_2)) \\ \implies &\mathbb{KL}\Big(\prod_k p(x_k)\|\prod_k q(x_k)\Big) = \sum_{i=1}^{k} \mathbb{KL}(p(x_i)\|q(x_i)) \end{aligned} \tag{35}$$

6.4.3 when $p(x_1, x_2) = p(x_1)p(x_2)$ and $q(x_1, x_2) = q(x_1)q(x_2)$ (2)

let $p(x) = \mathcal{N}(\mu_p, \sigma_p)$ and $q(x) = \mathcal{N}(\mu_q, \sigma_q)$:

$$\begin{aligned} \mathbb{KL}(p, q) &= -\int p(x)\log q(x)\mathbf{D}x + \int p(x)\log p(x)\mathbf{D}x \\ &= \frac{1}{2}\log(2\pi\sigma_q^2) + \frac{\sigma_p^2 + (\mu_p - \mu_q)^2}{2\sigma_q^2} - \frac{1}{2}(1 + \log 2\pi\sigma_p^2) \\ &= \log\frac{\sigma_q}{\sigma_p} + \frac{\sigma_p^2 + (\mu_p - \mu_q)^2}{2\sigma_q^2} - \frac{1}{2} \\ &= \log\sigma_q - \log\sigma_p + \frac{\sigma_p^2}{2\sigma_q^2} + \frac{(\mu_p - \mu_q)^2}{2\sigma_q^2} - \frac{1}{2} \end{aligned} \tag{36}$$

let $p(x) = \mathcal{N}(\mu, \sigma)$ and $q(x) = \mathcal{N}(0, 1)$:

$$\begin{aligned} \mathbb{KL}(p, q) &= \frac{\sigma^2}{2} + \frac{\mu^2}{2} - \frac{1}{2} - \log\sigma \\ &= \frac{1}{2}\Big[\frac{\sigma^2}{2} + \frac{\mu^2}{2} - \frac{1}{2} - \log\sigma^2\Big] \end{aligned} \tag{37}$$

moving into k dimensions, and apply $\mathbb{KL}\Big(\prod_k p(x_k)\|\prod_k q(x_k)\Big) = \sum_{i=1}^{k}\mathbb{KL}(p(x_i)\|q(x_i))$:

$$\mathbb{KL}\Big(\prod_k p(x_k)\|\prod_k q(x_k)\Big) = \frac{1}{2}\sum_k\Big[\frac{\sigma^2}{2} + \frac{\mu^2}{2} - \frac{1}{2} - \log\sigma^2\Big] \tag{38}$$

7 Gaussian Mixture Model variational inference

7.1 model

Refer to[1] for more details.

The model was presented from the paper [?]:

$$
\begin{aligned}
\mathbf{w} &\sim \mathcal{N}(0, \mathbf{I}) \\
\mathbf{z} &\sim \text{Mult}(\boldsymbol{\pi}) \\
\mathbf{x}|\mathbf{z}, \mathbf{w} &\sim \prod_{k=1} \mathcal{N}\big(\boldsymbol{\mu}_{z_k}(\mathbf{w}; \beta), \text{diag}\big(\boldsymbol{\sigma}^2_{z_k}(\mathbf{w}; \beta)\big)\big)^{z_k} \qquad \text{where } z_k \in \{0, 1\} \\
\mathbf{y}|\mathbf{x} &\sim \mathcal{N}\big(\boldsymbol{\mu}(\mathbf{x}; \theta), \text{diag}\big(\boldsymbol{\sigma}^2(\mathbf{w}; \theta)\big)\big)
\end{aligned}
\tag{39}
$$

Looking at the graphical model, it may also work if $\mathbf{w}$ is set to a fixed hyper-parameter. Basically, $\mathbf{x}$ is the $\mathbf{z}$ in the conventional VAE. The following is the relationship between the conventional and new representation:

$$
p(\mathbf{z}) \longrightarrow p(\mathbf{w})p(\mathbf{z})p_\beta(\mathbf{x}|\mathbf{w}, \mathbf{z}) \tag{40}
$$

note that in conventional VAE, $p(\mathbf{z})$ has no parameters. The decoder part is the similar:

$$
p(\mathbf{x}|\mathbf{z}) \longrightarrow p_\theta(\mathbf{y}|\mathbf{x}) \tag{41}
$$

7.2 choose appropriate $q(\cdot)$

the conventional $q(\mathbf{z}|\mathbf{x})$ becomes:

$$
q(\mathbf{x}, \mathbf{w}, \mathbf{z}|\mathbf{y}) = \prod_i q_{\phi_x}(\mathbf{x}_i|\mathbf{y}_i) q_{\phi_w}(\mathbf{w}_i|\mathbf{y}_i) p_\beta(\mathbf{z}_i|\mathbf{x}_i, \mathbf{w}_i) \tag{42}
$$

the key to the paper is that the prior $p_\beta(\mathbf{z}_i|\mathbf{x}_i, \mathbf{w}_i)$ also becomes part of $q(\cdot)$:

$$
\begin{aligned}
\text{ELBO} &= \mathbb{E}_q\left[\frac{p_{\beta,\theta}(\mathbf{y}, \mathbf{x}, \mathbf{w}, \mathbf{z})}{q(\mathbf{x}, \mathbf{w}, \mathbf{z}|\mathbf{y})}\right] \\
&= \mathbb{E}_q\left[\log\left(\frac{p_\theta(\mathbf{y}|\mathbf{x})p(\mathbf{w})p(\mathbf{z})p_\beta(\mathbf{x}|\mathbf{w}, \mathbf{z})}{q_{\phi_x}(\mathbf{x}|\mathbf{y})q_{\phi_w}(\mathbf{w}|\mathbf{y})p_\beta(\mathbf{z}|\mathbf{x}, \mathbf{w})}\right)\right] \\
&= \mathbb{E}_q\left[\log\left(p_\theta(\mathbf{y}|\mathbf{x})\right)\right] + \mathbb{E}_q\left[\log\left(\frac{p_\beta(\mathbf{x}|\mathbf{w}, \mathbf{z})}{q_{\phi_x}(\mathbf{x}|\mathbf{y})}\right)\right] + \mathbb{E}_q\left[\log\left(\frac{p(\mathbf{w})}{q_{\phi_w}(\mathbf{w}|\mathbf{y})}\right)\right] + \mathbb{E}_q\left[\log\left(\frac{p(\mathbf{z})}{p_\beta(\mathbf{z}|\mathbf{x}, \mathbf{w})}\right)\right]
\end{aligned}
\tag{43}
$$

Note that the $q(\cdot)$ used in the expectation is $q(\mathbf{x}, \mathbf{w}, \mathbf{z}|\mathbf{y}) = q_{\phi_x}(\mathbf{x}|\mathbf{y})q_{\phi_w}(\mathbf{w}|\mathbf{y})p_\beta(\mathbf{z}|\mathbf{x}, \mathbf{w})$, therefore we can omit terms contain variables do not appear inside the expectation. Also we rewrite the expectation into separate terms that participate towards KL:

$$
\begin{aligned}
\text{ELBO} = &\ \mathbb{E}_{q(\mathbf{x}|\mathbf{y})}\left[\log\left(p_\theta(\mathbf{y}|\mathbf{x})\right)\right] + \mathbb{E}_{q_{\phi_w}(\mathbf{w}|\mathbf{y})p_\beta(\mathbf{z}|\mathbf{x},\mathbf{w})}\mathbb{E}_{q_{\phi_x}(\mathbf{x}|\mathbf{y})}\left[\log\left(\frac{p_\beta(\mathbf{x}|\mathbf{w}, \mathbf{z})}{q_{\phi_x}(\mathbf{x}|\mathbf{y})}\right)\right] \\
&+ \mathbb{E}_{q_{\phi_w}(\mathbf{w}|\mathbf{y})}\left[\log\left(\frac{p(\mathbf{w})}{q_{\phi_w}(\mathbf{w}|\mathbf{y})}\right)\right] + \mathbb{E}_{q_{\phi_x}(\mathbf{x}|\mathbf{y})q_{\phi_w}(\mathbf{w}|\mathbf{y})}\mathbb{E}_{p_\beta(\mathbf{z}|\mathbf{x},\mathbf{w})}\left[\log\left(\frac{p(\mathbf{z})}{p_\beta(\mathbf{z}|\mathbf{x}, \mathbf{w})}\right)\right] \\
= &\ \mathbb{E}_{q(\mathbf{x}|\mathbf{y})}\left[\log\left(p_\theta(\mathbf{y}|\mathbf{x})\right)\right] - \mathbb{E}_{q_{\phi_w}(\mathbf{w}|\mathbf{y})p_\beta(\mathbf{z}|\mathbf{x},\mathbf{w})}\left[\mathbb{KL}(q_{\phi_x}(\mathbf{x}|\mathbf{y})\|p_\beta(\mathbf{x}|\mathbf{w}, \mathbf{z}))\right] \\
&- \mathbb{KL}[q_{\phi_w}(\mathbf{w}|\mathbf{y})\|p(\mathbf{w})] - \mathbb{E}_{q_{\phi_x}(\mathbf{x}|\mathbf{y})q_{\phi_w}(\mathbf{w}|\mathbf{y})}\left[\mathbb{KL}(p_\beta(\mathbf{z}|\mathbf{x}, \mathbf{w})\|p(\mathbf{z}))\right]
\end{aligned}
\tag{44}
$$

[1] https://arxiv.org/pdf/1611.02648.pdf

$$\begin{aligned} p_\beta(z_{i,j} = 1|\mathbf{x}, \mathbf{w}) &= \frac{p(z_{i,j} = 1)p(\mathbf{x}|z_{i,j} = 1, \mathbf{w})}{\sum_{k=1}^{K} p(z_{i,k} = 1)p(\mathbf{x}|z_{i,k} = 1, \mathbf{w})} \\ &= \frac{\boldsymbol{\pi}_i \mathcal{N} p(\mathbf{x}|z_j = 1, \mathbf{w})}{\sum_{k=1}^{K} p(z_j = k)p(\mathbf{x}|z_j = k, \mathbf{w})} \end{aligned} \tag{45}$$

8 Traditional Variational Bayes for DPMM

This is from [?] and let $\{\eta_1^*, \dots, \eta_{|c|}^*\}$ denote distinct values of $\{\eta_1, \dots, \eta_n\}$, let $\mathbf{c} = \{c_1, \dots, c_n\}$. Looking at ELBO, letting posterior to be $p(\mathbf{V}, \boldsymbol{\eta}, \mathbf{Z}|\mathbf{x})$, and we use split three:

$$\begin{aligned}
\text{ELBO}_{(\theta,\phi)} &= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z})}\big[\log p_\theta(\mathbf{x},\mathbf{z}) - \log q_\phi(\mathbf{z})\big] \\
&= \mathbb{E}_{q_\phi(\mathbf{V})}\big[\log p_\theta(\mathbf{V}|\lambda)\big] + \mathbb{E}_{q_\phi(\boldsymbol{\eta}^*)}\big[\log p_\theta(\boldsymbol{\eta}^*|\lambda)\big] \\
&+ \sum_{n=1}^{N}\Big(\mathbb{E}_{q_\phi(Z_n,\mathbf{V})}\big[\log p(Z_n|\mathbf{V})\big] + \underbrace{\mathbb{E}_{q_\phi(Z_n)}\big[\log p(\mathbf{x}_n|Z_n)\big]}_{\text{should be } \mathbb{E}_{q_\phi(Z_n,\boldsymbol{\eta}^*)}\big[\log p(\mathbf{x}_n|Z_n,\boldsymbol{\eta}^*)\big]}\Big) \\
&- \mathbb{E}_{q_\phi(\mathbf{V},\boldsymbol{\eta}^*,\mathbf{Z})}\big[q_\phi(\mathbf{V},\boldsymbol{\eta}^*,\mathbf{Z})\big]
\end{aligned} \tag{46}$$

the most important part is to realize that:

$$\mathbb{E}_{q(Z_n,\mathbf{V})}\big[\log p(Z_n|\mathbf{V})\big] = \mathbb{E}_{q(Z_n,\mathbf{V})}\Big[\log\big(\prod_{i=1}^{\infty}(1-V_i)^{\mathbf{1}_{Z_n>i}}\, V_i^{\mathbf{1}_{Z_n=i}}\big)\Big] \tag{47}$$

For example $Z_n = 5$, which means it does not belong to stick $1, 2, 3, 4$, but to stick 5. The reason the rest of the sticks 6 and beyond are not involved in the calculation is because the sticks comes in this stick-breaking order. They do not need to be part of the calculation. Then we have:

$$\begin{aligned}
&= \mathbb{E}_{q(Z_n,\mathbf{V})}\Big[\sum_{i=1}^{\infty}\log\big((1-V_i)^{\mathbf{1}_{Z_n>i}}\, V_i^{\mathbf{1}_{Z_n=i}}\big)\Big] \\
&= \sum_{i=1}^{\infty}\mathbb{E}_{q(Z_n,V_i)}\log\big((1-V_i)^{\mathbf{1}_{Z_n>i}}\, V_i^{\mathbf{1}_{Z_n=i}}\big) \\
&= \sum_{i=1}^{T}\mathbb{E}_{q(Z_n,V_i)}\log\big((1-V_i)^{\mathbf{1}_{Z_n>i}}\, V_i^{\mathbf{1}_{Z_n=i}}\big) \\
&= \sum_{i=1}^{T}\mathbb{E}_{q(Z_n,V_i)}\log(1-V_i)^{\mathbf{1}_{Z_n>i}} + \sum_{i=1}^{T}\mathbb{E}_{q(Z_n,V_i)}\log V_i^{\mathbf{1}_{Z_n=i}} \\
&= \sum_{i=1}^{T} q(Z_n > i)\mathbb{E}_{q(V_i)}[\log(1-V_i)] + \sum_{i=1}^{T} q(Z_n = i)\mathbb{E}_{q(V_i)}[\log V_i]
\end{aligned} \tag{48}$$

the last line because the alternative case is $\mathbf{1}_{Z_n>i} = 0 \implies \log(1-V_i)^{\mathbf{1}_{Z_n>i}} = 0$, then we have:

$$\begin{aligned}
q(Z_n = i) &= \phi_{n,i} \\
q(Z_n > i) &= \sum_{j=i+1}^{T}\phi_{n,j}
\end{aligned} \tag{49}$$

9 Stick-breaking VAE

this comes from the paper [?]. The main improvemnts is the use of Kumaraswamy distribution which generate non-central re-parameterization.

$$\begin{aligned} v_k &\sim \text{Beta}(1, \alpha) \\ \pi_k &= v_k \prod_{l=1}^{k-1} (1 - v_l) \\ \theta_k &\sim H \\ G_0 &= \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k} \end{aligned} \tag{50}$$

9.1 re-parameterization

unlike the VAE algorithm from Eq.(??) where one can re-parameterize:

$$\begin{aligned} &\text{re-parameterization:} \\ &\quad \epsilon \sim \mathcal{N}(0, \mathbf{I}) \\ &\quad \mathbf{z} = \text{Encoder}_\phi(\mathbf{x}, \epsilon) \\ &\quad\quad = \mu_\phi(\mathbf{x}) + \Sigma_\phi(\mathbf{x}) \times \epsilon \end{aligned} \tag{51}$$

one may not do the same if $q_\phi(v_k|\mathbf{x})$ has a beta distribution, i.e., beta distribution does not generate non-central re-parameterization. Therefore we need to have:

$$q(v) \equiv \text{Kumaraswamy}(v; a, b) = abv^{a-1}(1 - v^a)^{b-1} \tag{52}$$

since one can re-parameterize it through the inverse of CDF:

$$v = (1 - u^{\frac{1}{b}})^{\frac{1}{a}} \qquad u \sim \text{Uniform}(0, 1) \tag{53}$$

10 Adversarial Variational Bayes

This section is to explain [?]

it uses split one of ELBO:

$$\begin{aligned} \text{ELBO}_{(\theta,\phi)} &= \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \Big[\log p_\theta(\mathbf{x}|\mathbf{z}) \Big] - \mathbb{KL} \Big[q_\phi(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z}) \Big] \\ &= \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \left[\log p_\theta(\mathbf{x}|\mathbf{z}) - \log \frac{q_\phi(z|x)}{p(z)} \right] \\ &= \max_\psi \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \Big[\log p_\theta(\mathbf{x}|\mathbf{z}) - T_\psi(\mathbf{x}, \mathbf{z}) \Big] \\ &= \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \Big[\log p_\theta(\mathbf{x}|\mathbf{z}) - T^*_\psi(\mathbf{x}, \mathbf{z}) \Big] \end{aligned} \tag{54}$$

the paper ignores structure of $\log \frac{q_\phi(z|x)}{p(z)}$ and train to obtain $T^*_\psi(\mathbf{x}, \mathbf{z})$ complete separate network.

in VAE, one needs to assume how to evaluate $q_\phi(\mathbf{z}|\mathbf{x})$ to be some distribution, in AVB, we treat it as black-box inference model, we only need to know how to sample from $q_\phi(\mathbf{z}|\mathbf{x})$

10.1 how do you obtain $T^*_\psi(\mathbf{x}, \mathbf{z})$

we use the following objective function:

$$T^*_\psi(\mathbf{x}, \mathbf{z}) = \max_\psi \mathbb{E}_{p(\mathbf{x})} \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \Big[\log \sigma(T_\psi(\mathbf{x}, \mathbf{z})) \Big] + \mathbb{E}_{p(\mathbf{x})} \mathbb{E}_{p(\mathbf{z})} \Big[\log(1 - \sigma(T_\psi(\mathbf{x}, \mathbf{z}))) \Big] \tag{55}$$

this expression looks like a logistic regression to differentiate $(\mathbf{x}, \mathbf{z})$ between $\underbrace{p(\mathbf{x}) q_\phi(\mathbf{z}|\mathbf{x})}_{\text{real}}$ and $\underbrace{p(\mathbf{x}) p(\mathbf{z})}_{\text{fake}}$

note that we didn't use $p(\mathbf{x}, \mathbf{z})$ but instead $p(\mathbf{x})$ and $p(\mathbf{z})$

10.1.1 why does this objective work?

we must prove the following lemma:

Lemma 1 by defining $T^*_\psi(\mathbf{x}, \mathbf{z})$ to be:

$$T^*_\psi(\mathbf{x}, \mathbf{z}) = \max_\psi \mathbb{E}_{p(\mathbf{x})} \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \Big[\log \sigma(T(\mathbf{x}, \mathbf{z})) \Big] + \mathbb{E}_{p(\mathbf{x})} \mathbb{E}_{p(\mathbf{z})} \Big[\log(1 - \sigma(T(\mathbf{x}, \mathbf{z}))) \Big] \tag{56}$$

we then have:

$$\mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \Big[\log p_\theta(\mathbf{x}|\mathbf{z}) - T^*_\psi(\mathbf{x}, \mathbf{z}) \Big] = \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \Big[\log p_\theta(\mathbf{x}|\mathbf{z}) \Big] - \mathbb{KL} \Big[q_\phi(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z}) \Big] \tag{57}$$

i.e., after $\max_\psi$, we get our original ELBO back. Consequentially, we have the following overall objective:

10.1.2 overall objective

$$\begin{aligned}
&\max_\theta \max_\phi \mathrm{ELBO}_{(\theta,\phi)} \\
&= \max_\theta \max_\phi \big[\mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \big[\log p_\theta(\mathbf{x}|\mathbf{z}) - T^*_\psi(\mathbf{x}, \mathbf{z}) \big] \big] \\
&= \max_\theta \max_\phi \Big[\mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \Big[\log p_\theta(\mathbf{x}|\mathbf{z}) - \max_\psi \big[\mathbb{E}_{p(\mathbf{x})} \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \big[\log \sigma(T_\psi(\mathbf{x}, \mathbf{z})) \big] + \mathbb{E}_{p(\mathbf{x})} \mathbb{E}_{p(\mathbf{z})} \big[\log(1 - \sigma(T_\psi(\mathbf{x}, \mathbf{z}))) \big] \big] \Big] \Big] \\
&= \max_\theta \max_\phi \min_\psi \Big[\mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \big[\log p_\theta(\mathbf{x}|\mathbf{z}) \big] - \mathbb{E}_{p(\mathbf{x})} \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \big[\log \sigma(T_\psi(\mathbf{x}, \mathbf{z})) \big] + \mathbb{E}_{p(\mathbf{x})} \mathbb{E}_{p(\mathbf{z})} \big[\log(1 - \sigma(T_\psi(\mathbf{x}, \mathbf{z}))) \big] \Big]
\end{aligned} \tag{58}$$

10.1.3 proof is similarity to GAN's optimum $D^*(\mathbf{x})$

look at GAN after fix G and optimize D: (see my GAN notes):

$$\begin{aligned}
&\max_D \mathbb{E}_{\mathbf{x} \sim p_{\mathrm{r}}(\mathbf{x})} [\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{x} \sim p^\theta_{\mathrm{g}}(\mathbf{x})} [\log(1 - D(\mathbf{x})] \\
\Longrightarrow \; &D^*(x) = \frac{p_{\mathrm{r}}(x)}{p_{\mathrm{r}}(x) + p^\theta_{\mathrm{g}}(x)}
\end{aligned} \tag{59}$$

compare it with Eq.(??) and to look at pattern, the best $\sigma(T^*(\mathbf{x}, \mathbf{z}))$ should occur when:

$$\begin{aligned}
\sigma(T^*(\mathbf{x}, \mathbf{z})) &= \frac{p(\mathbf{x}) q_\phi(\mathbf{z}|\mathbf{x})}{p(\mathbf{x}) q_\phi(\mathbf{z}|\mathbf{x}) + p(\mathbf{x}) p(\mathbf{z})} \\
&= \frac{q_\phi(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x}) + p(\mathbf{z})} \\
&= \frac{q}{q + p} \quad \text{for simple notation}
\end{aligned} \tag{60}$$

$$
\begin{aligned}
\implies \frac{1}{1+\exp(-T^{*})} &= \frac{q}{q+p} \qquad \text{definition of } \sigma \\
\implies q+p &= q(1+\exp(-T^{*}) \\
\implies p &= q\exp(-T^{*}) \\
\implies \log\frac{p}{q} &= -T^{*} \\
\implies T^{*}_{\psi} &= \log(q_{\phi}(\mathbf{z}|\mathbf{x})) - \log p(\mathbf{z})
\end{aligned}
\tag{61}
$$

in summary, by calculating:

$$
\begin{aligned}
T^{*}_{\psi}(\mathbf{x},\mathbf{z}) &= \max_{\psi} \mathbb{E}_{p(\mathbf{x})}\mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})}\Big[\log\sigma(T(\mathbf{x},\mathbf{z}))\Big] + \mathbb{E}_{p(\mathbf{x})}\mathbb{E}_{p(\mathbf{z})}\Big[\log(1-\sigma(T(\mathbf{x},\mathbf{z})))\Big] \\
\implies &\mathbb{E}_{\mathbf{z}\sim q_{\phi}(\mathbf{z}|\mathbf{x})}\Big[\log p_{\theta}(\mathbf{x}|\mathbf{z}) - T^{*}_{\psi}(\mathbf{x},\mathbf{z})\Big] \\
&= \mathbb{E}_{\mathbf{z}\sim q_{\phi}(\mathbf{z}|\mathbf{x})}\Big[\log p_{\theta}(\mathbf{x}|\mathbf{z}) - \log\frac{q_{\phi}(\mathbf{z}|\mathbf{x})}{p(\mathbf{z})}\Big] \\
&= \mathbb{E}_{\mathbf{z}\sim q_{\phi}(\mathbf{z}|\mathbf{x})}\big[\log p_{\theta}(\mathbf{x}|\mathbf{z})\big] - \mathbb{KL}(q_{\phi}(\mathbf{z}|\mathbf{x})\|p(\mathbf{z}))
\end{aligned}
\tag{62}
$$

11 Normalized Flow

The first paper of VB on Normalized Flow can be found at [?]

11.1 Revision on Change of Variable

looking at example with 1-D:

Let's consider a simple 1-D example where $y = 2x$. This means $f(x) = 2x$ and $f^{-1}(y) = \frac{y}{2}$.

For a function $g(x)$, we want to perform a change of variable to express it in terms of y:

$$\begin{aligned} \int_0^1 g(x)\,dx &= \int_0^2 g(f^{-1}(y)) \left| \frac{d}{dy} f^{-1}(y) \right| dy \\ &= \int_0^2 g\left(\frac{y}{2}\right) \left| \frac{d}{dy} \left(\frac{y}{2}\right) \right| dy \\ &= \int_0^2 g\left(\frac{y}{2}\right) \cdot \frac{1}{2}\, dy \end{aligned} \tag{63}$$

the reason why there is absolute value is because we want to make sure that the integral is still valid even if the function is not monotonic.

Here, $\left| \frac{d}{dy} f^{-1}(y) \right| = \frac{1}{2}$ is the absolute value of the derivative of the inverse function, which corresponds to the determinant of the Jacobian in higher dimensions.

This example illustrates how the change of variable affects both the argument of the function and introduces a scaling factor (in this case, $\frac{1}{2}$) to maintain the equality of the integrals.

take Integration by substitution problem to higher dimensions, and let:

$$\mathbf{y} = f(\mathbf{x}) \quad \Longrightarrow \quad \mathbf{x} = f^{-1}(\mathbf{y}) \tag{64}$$

let $\mathbf{x} \in \mathbf{R}^n$ and $\mathbf{y} \in \mathbf{R}^n$

So how are $\mathrm{d}\mathbf{x}$ and $\mathrm{d}\mathbf{y}$ related? in turns out that:

$$\underbrace{\mathrm{d}x_1 \cdots \mathrm{d}x_n}_{\text{corresponding infinitesimal base volume in } \mathrm{d}\mathbf{x}} = \underbrace{\left| \det\left(\frac{\partial \mathbf{x}}{\partial \mathbf{y}}\right) \right|}_{\text{volume of change ratios}} \underbrace{\mathrm{d}y_1 \cdots \mathrm{d}y_n}_{\text{reference infinitesimal base volume in } \mathrm{d}\mathbf{y}} \tag{65}$$

using $\mathbf{Dy}$ as the infinitesimal "reference" base volume, then the corresponding $\mathbf{Dx}$ (or $\mathbf{D}f^{-1}(\mathbf{y})$) must be:

$$\mathrm{d}\mathbf{y} \times \underbrace{\text{volume of } \underbrace{\text{instantaneous changes ratio between } \mathbf{x} \text{ and } \mathbf{y}}_{\textcircled{1}}}_{\textcircled{2}} \tag{66}$$

there are two parts to the above equation, $\textcircled{1}$ "instantaneous changes ratio between $\mathbf{x}$ and $\mathbf{y}$" can be described by:

$$\frac{\partial \mathbf{x}}{\partial \mathbf{y}} = \frac{\partial f^{-1}(\mathbf{y})}{\partial \mathbf{y}} \tag{67}$$

which is the Jacobian matrix w.r.t. to $\mathbf{y}$, and then, the $\textcircled{2}$ is the volume of the parallelotope spanned by the columns of this Jacobian. This is the determinant!

$$\left|\det\left(\frac{\partial f^{-1}(\mathbf{y})}{\partial \mathbf{y}}\right)\right| \tag{68}$$

one can visualize it as following: given a reference rectangle volume (or area if it's 2D): $\mathrm{d}\mathbf{y}$, through mapping function $f^{-1}(\mathbf{y})$, its corresponding parallelogram $\mathrm{d}\mathbf{x}$'s volume is determined by $\left|\det\left(\frac{\partial f^{-1}(\mathbf{y})}{\partial \mathbf{y}}\right)\right| \mathrm{d}\mathbf{y}$. formally:

$$\begin{aligned} \mathrm{d}x_1 \cdots \mathrm{d}x_n &= \left|\det\left(\frac{\partial f^{-1}(\mathbf{y})}{\partial \mathbf{y}}\right)\right| \mathrm{d}y_1 \cdots \mathrm{d}y_n, \text{ or,} \\ \mathrm{d}\mathbf{x} &= \left|\det\left(\frac{\partial f^{-1}(\mathbf{y})}{\partial \mathbf{y}}\right)\right| \mathrm{d}\mathbf{y} \end{aligned} \tag{69}$$

11.1.1 apply change of variable to probability

$$\begin{aligned} \Pr\left(Y \in \mathbf{S}\right) &= \int_{\mathbf{S}} \underbrace{p_Y(\mathbf{y})}\, \mathrm{d}\mathbf{y} \\ &= \int_{f^{-1}(\mathbf{S})} p_X(\mathbf{x})\, \mathrm{d}\mathbf{x} \qquad = \Pr\left(X \in f^{-1}(\mathbf{S})\right) \\ &= \int_{f^{-1}(\mathbf{S})} p_X(\mathbf{x}) \left|\det\left(\frac{\partial f^{-1}(\mathbf{y})}{\partial \mathbf{y}}\right)\right| \mathrm{d}\mathbf{y} \qquad \text{substitute change of variable} \\ &= \int_{\mathbf{S}} \underbrace{p_X(f^{-1}(\mathbf{y})) \left|\det\left(\frac{\partial f^{-1}(\mathbf{y})}{\partial \mathbf{y}}\right)\right|}\, \mathrm{d}\mathbf{y} \\ \implies p_Y(\mathbf{y}) &= p_X(f^{-1}(\mathbf{y})) \left|\det\left(\frac{\partial f^{-1}(\mathbf{y})}{\partial \mathbf{y}}\right)\right| \qquad \text{things inside the integral} \\ &= p_X(f^{-1}(\mathbf{y})) \left|\det\left(\frac{\partial \mathbf{y}}{\partial f^{-1}(\mathbf{y})}\right)\right|^{-1} \qquad \text{property of } \det(.) \\ &= p_X(\mathbf{x}) \left|\det\left(\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}\right)\right|^{-1} \qquad \text{that's the familiar expression} \end{aligned} \tag{70}$$

11.1.2 one reason to have $|\det(\cdot)|$

$$\begin{aligned} \Pr(b \le Y \le a) &= \int_a^b p_X(f^{-1}(\mathbf{y})) \left(\det \frac{\partial f^{-1}(\mathbf{y})}{\partial \mathbf{y}}\right) \mathbf{Dy} \\ &= \int_b^a p_X(f^{-1}(\mathbf{y})) \left(-\det \frac{\partial f^{-1}(\mathbf{y})}{\partial \mathbf{y}}\right) \mathbf{Dy} \\ &= \int_a^b p_X(f^{-1}(\mathbf{y})) \left|\det \frac{\partial f^{-1}(\mathbf{y})}{\partial \mathbf{y}}\right| \mathbf{Dy} \end{aligned} \tag{71}$$

11.2 apply to Normalized Flow

re-writing $\mathbf{x} \to \mathbf{z}$, and $\mathbf{y} \to \mathbf{z}'$:

$$p(\mathbf{z}') = p(\mathbf{z}) \left| \det \frac{\partial f(\mathbf{z})}{\partial \mathbf{z}} \right|^{-1} \tag{72}$$

we let:

$$\mathbf{z}_K = f_K \circ \cdots \circ f_2 \circ f_1(\mathbf{z}_0) \tag{73}$$

starting backwards, and let $\underbrace{\mathbf{z}_K}_{\mathbf{z}'} = f_K(\underbrace{\mathbf{z}_{K-1}}_{\mathbf{z}})$:

$$\begin{aligned}
p(\mathbf{z}_K) &= \underbrace{p(\mathbf{z}_{K-1})} \left| \det \frac{\partial f_K(\mathbf{z}_{K-1})}{\partial \mathbf{z}_{K-1}} \right|^{-1} \\
&= \underbrace{p(\mathbf{z}_{K-2}) \left| \det \frac{\partial f_{K-1}(\mathbf{z}_{K-2})}{\partial \mathbf{z}_{K-2}} \right|^{-1}} \times \left| \det \frac{\partial f_K(\mathbf{z}_{K-1})}{\partial \mathbf{z}_{K-1}} \right|^{-1} \\
&= \vdots \\
&= p_0(\mathbf{z}_0) \left| \det \frac{\partial f_1(\mathbf{z}_0)}{\partial \mathbf{z}_0} \right|^{-1} \times \cdots \times \left| \det \frac{\partial f_{K-1}(\mathbf{z}_{K-2})}{\partial \mathbf{z}_{K-2}} \right|^{-1} \times \left| \det \frac{\partial f_K(\mathbf{z}_{K-1})}{\partial \mathbf{z}_{K-1}} \right|^{-1} \\
\implies \log(p(\mathbf{z}_K) &= \log\big(p_0(\mathbf{z}_0)\big) + \sum_{k=1}^{K} \log \left| \det \frac{\partial f_k(\mathbf{z}_{k-1})}{\partial \mathbf{z}_{k-1}} \right|^{-1} \\
&= \log\big(p_0(\mathbf{z}_0)\big) - \sum_{k=1}^{K} \log \left| \det \frac{\partial f_k(\mathbf{z}_{k-1})}{\partial \mathbf{z}_{k-1}} \right|^{-1}
\end{aligned} \tag{74}$$

11.2.1 Expectation

using the final equation form:

$$\log(p(\mathbf{z}_K) = \log\big(p(\mathbf{z}_0)\big) - \sum_{k=1}^{K} \log \left| \det \frac{\partial f_k(\mathbf{z}_{k-1})}{\partial \mathbf{z}_{k-1}} \right|^{-1} \tag{75}$$

substitute it to derive expectation:

$$\begin{aligned}
\mathbb{E}_{p_K}[h(\mathbf{z})] \equiv \mathbb{E}_{p(\mathbf{z}_K)}[h(\mathbf{z}_K)] \\
&= \int_{\mathbf{z}_K} h(\mathbf{z}_K) p(\mathbf{z}_K) \mathbf{D}\mathbf{z}_K \\
&= \int_{\mathbf{z}_0} h\big(f_K \circ \cdots \circ f_2 \circ f_1(\mathbf{z}_0)\big) p(\mathbf{z}_0) \mathbf{D}\mathbf{z}_0 \\
&= \mathbb{E}_{p(\mathbf{z}_0)}\big[h\big(f_K \circ \cdots \circ f_2 \circ f_1(\mathbf{z}_0)\big)\big]
\end{aligned} \tag{76}$$

11.3 variational learning of Normalized Flow

Obviously, Normalized Flow is used in a varieties of settings. However, when it is used in ELBO, it is used in $q_\phi(\mathbf{z})$ we replace all previous representation from $p \to q$, and also not explicitly writing out q_ϕ for clarity:

let $q_\phi(\mathbf{z}|\mathbf{x}) \equiv q_K(\mathbf{z}_K)$, and substitute:

$$\log(q_\phi(\mathbf{z}_K) = \log\big(q_0(\mathbf{z}_0)\big) - \sum_{k=1}^{K} \log \left| \det \frac{\partial f_k(\mathbf{z}_{k-1})}{\partial \mathbf{z}_{k-1}} \right|^{-1} \tag{77}$$

It us using split two of the ELBO:

$$\begin{aligned}
\text{ELBO}_{(\theta,\phi)} &= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x},\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})\right] \\
&= \mathbb{E}_{\mathbf{z}_K\sim q_K(\mathbf{z}_K)}\left[\log p_\theta(\mathbf{x},\mathbf{z}_K) - \log q_K(\mathbf{z}_K)\right] \qquad \text{using } q_K(\mathbf{z}_K) \equiv q_\phi(\mathbf{z}|\mathbf{x}) \\
&= \mathbb{E}_{\mathbf{z}_0\sim q_0(\mathbf{z}_0|\mathbf{x})}\left[\log p_\theta(\mathbf{x},\mathbf{z}_K) - \log q_K(\mathbf{z}_K)\right] \qquad q_0(\mathbf{z}_0) = q_K(\mathbf{z}_K) \text{ by NF construction} \\
&= \mathbb{E}_{\mathbf{z}_0\sim q_0(\mathbf{z}_0|\mathbf{x})}\left[\log p_\theta(\mathbf{x},\mathbf{z}_K) - \log\left(q_0(\mathbf{z}_0|\mathbf{x})\right) + \sum_{k=1}^{K}\log\left|\det\frac{\partial f_k(\mathbf{z}_{k-1})}{\partial \mathbf{z}_{k-1}}\right|^{-1}\right]
\end{aligned} \tag{78}$$

$\mathbf{z}_K$ replaces what was $\hat{\mathbf{x}}$ (reconstructed $\mathbf{x}$) in traditional VAE.

11.3.1 NF variational algorithm

now by keeping det $\frac{\partial f_k(\mathbf{z}_{k-1})}{\partial \mathbf{z}_{k-1}} = 1$, which is not necessary, but convenient to make:

$$\sum_{k=1}^{K}\log\left|\det\frac{\partial f_k(\mathbf{z}_{k-1})}{\partial \mathbf{z}_{k-1}}\right|^{-1} = 0 \tag{79}$$

in each iteration:

$$\begin{aligned}
&\text{get mini-batch } \{\mathbf{x}\} \\
&\mathbf{z}_0 \sim q_0(\cdot|\mathbf{x}) \\
&\qquad \text{re-parameterization it as:} \\
&\qquad\quad \epsilon \sim \mathcal{N}(0,\mathbf{I}) \\
&\qquad\quad \mathbf{z}_0|\mathbf{x} \equiv \mathbf{z}_{0\phi}(\mathbf{x},\epsilon) \\
&\qquad\qquad = \mu_\phi(\mathbf{x}) + \Sigma_\phi(\mathbf{x}) \times \epsilon \\
&\mathbf{z}_K \leftarrow f_K^\theta \circ \cdots \circ f_2^\theta \circ f_1^\theta(\mathbf{z}_0) \qquad \text{decoder step} \\
&\triangle\theta \propto -\nabla_\theta \text{ELBO}_{\theta,\phi}(\mathbf{x}, z_K) \\
&\triangle\phi = -\nabla_\phi \text{ELBO}_{\theta,\phi}(\mathbf{x}, z_K)
\end{aligned} \tag{80}$$

11.4 How to construct such function

remaining task is to choose appropriate $f_k(\mathbf{z}_{k-1})$ such that it is easy to find:

1. inverse of forward function
2. determinant of Jacobian

11.4.1 Planner

Forward mapping $\mathbf{z}_{k-1} \to \mathbf{z}_k$:

$$\mathbf{z}_k = f_\theta(\mathbf{z}_{k-1}) = \mathbf{z}_{k-1} + \mathbf{u}\mathbf{h}(\mathbf{w}^\top \mathbf{z}_{k-1} + \mathbf{b}) \tag{81}$$

1. inverse

   $f_\theta^{-1}(\mathbf{z}_k)$ can be difficult, need careful choice

2. determinant

   $$\left|\det\left(\frac{\partial \mathbf{z}_k}{\partial \mathbf{z}_{k-1}}\right)\right| = \left|1 + h'(\mathbf{w}^\top \mathbf{z}_{k-1} + \mathbf{b})\mathbf{u}^\top \mathbf{w}\right| \tag{82}$$

   need to make sure $h'(\cdot)$ is bounded.

11.4.2 Nonlinear Independent Components Estimation (NICE)

In [?], we have the following construction:
Forward mapping $\mathbf{z}_{k-1} \rightarrow \mathbf{z}_k$
each $\mathbf{z}_k$ are separated into two portions $\{\mathbf{z}_k^{(1)}, \mathbf{z}_k^{(2)}\}$:

$$\begin{aligned} \mathbf{z}_k^{(1)} &= \mathbf{z}_{k-1}^{(1)} \\ \mathbf{z}_k^{(2)} &= \mathbf{z}_{k-1}^{(2)} + f_{\theta^{(1)}}(\mathbf{z}_{k-1}^{(1)}) \end{aligned} \tag{83}$$

one can also rotate the role of $\{\mathbf{z}_k^{(1)}, \mathbf{z}_k^{(2)}\}$, such that:

$$\begin{aligned} \mathbf{z}_{k'}^{(2)} &= \mathbf{z}_{k'-1}^{(2)} \\ \mathbf{z}_{k'}^{(1)} &= \mathbf{z}_{k'-1}^{(1)} + f_{\theta^{(2)}}(\mathbf{z}_{k'-1}^{(2)}) \end{aligned} \tag{84}$$

1. inverse mapping

   Inverse mapping $\mathbf{z}_k \rightarrow \mathbf{z}_{k-1}$:

$$\begin{aligned} \mathbf{z}_{k-1}^{(1)} &= \mathbf{z}_k^{(1)} \\ \mathbf{z}_{k-1}^{(2)} &= \mathbf{z}_k^{(2)} - f_{\theta^{(1)}}(\mathbf{z}_k^{(1)}) \end{aligned} \tag{85}$$

2. determinant of Jacobian

   Either Eq.(??) gives:

$$\begin{bmatrix} \frac{\partial \mathbf{z}_k^{(1)}}{\partial \mathbf{z}_{k-1}^{(1)}} & \frac{\partial \mathbf{z}_k^{(1)}}{\partial \mathbf{z}_{k-1}^{(2)}} \\ \frac{\partial \mathbf{z}_k^{(2)}}{\partial \mathbf{z}_{k-1}^{(1)}} & \frac{\partial \mathbf{z}_k^{(2)}}{\partial \mathbf{z}_{k-1}^{(2)}} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{d_1} & \mathbf{0} \\ f'_{\theta^{(1)}}(\mathbf{z}_{k-1}^{(1)}) & \mathbf{I}_{d_2} \end{bmatrix} \tag{86}$$

   and Eq.(??) gives (change $k' \rightarrow k$):

$$\begin{bmatrix} \frac{\partial \mathbf{z}_k^{(1)}}{\partial \mathbf{z}_{k-1}^{(1)}} & \frac{\partial \mathbf{z}_k^{(1)}}{\partial \mathbf{z}_{k-1}^{(2)}} \\ \frac{\partial \mathbf{z}_k^{(2)}}{\partial \mathbf{z}_{k-1}^{(1)}} & \frac{\partial \mathbf{z}_k^{(2)}}{\partial \mathbf{z}_{k-1}^{(2)}} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{d_1} & f'_{\theta^{(2)}}(\mathbf{z}_{k-1}^{(2)}) \\ \mathbf{0} & \mathbf{I}_{d_2} \end{bmatrix} \tag{87}$$

   the determinant is one. the moral of the story is that this setup works for any “complex” $f_\theta(\cdot)$

11.4.3 RealNVP

Volume preserving may be too restrictive, so some NVP model is proposed, for example in [?]:

$$\begin{aligned} \mathbf{z}_k^{(2)} &= \exp^{s_\theta\left(\mathbf{z}_{k-1}^{(1)}\right)} \odot \mathbf{z}_{k-1}^{(2)} + f_\theta(\mathbf{z}_{k-1}^{(1)}) \\ \frac{\partial \mathbf{z}_k^{(2)}}{\partial \mathbf{z}_{k-1}^{(1)}} &= \begin{bmatrix} \exp^{s_\theta\left(\mathbf{z}_{k-1,1}^{(1)}\right)} & \dots & 0 \\ 0 & \ddots & 0 \\ 0 & \dots & \exp^{s_\theta\left(\mathbf{z}_{k-1,d_1}^{(1)}\right)} \end{bmatrix} \qquad \text{assume } \mathbf{z}^{(1)} \in \mathbb{R}^{d_1} \end{aligned} \tag{88}$$

11.5 Continuous time Normalize Flow

technically, it concerns most with ODE, but I leave it here for completeness. These are found in [?].

$$\begin{aligned}\log p_X(\mathbf{x}) &= \log p_Z(\mathbf{z}) + \int_{t=0}^{1} \frac{\partial \log p(\mathbf{z}(t))}{\partial t}\,\mathrm{d}t \\ &= \log p_Z(\mathbf{z}) + \int_{t=0}^{1} \mathrm{Tr}\Big(\frac{\partial f(\mathbf{z}(t),t)}{\partial \mathbf{z}(t)}\Big)\mathrm{d}t \qquad \text{see neuralODE paper} \\ \mathbf{z} &= \mathbf{x} + \int_{t=0}^{1} f_\theta(\mathbf{x}(t))\mathrm{d}t\end{aligned} \tag{89}$$

where $\mathbf{x}(0) = \mathbf{x}$ and $\mathbf{x}(1) = \mathbf{z}$, i.e, the encoder

note, we replaced the "discrete" Jacobian $\frac{\partial \mathbf{z}_k}{\partial \mathbf{z}_{k-1}}$ by the "continuous" Jacobian $\frac{\partial f(\mathbf{z}(t),t)}{\partial \mathbf{z}(t)}$. But essentially, it's just the $\frac{\partial \text{output}}{\partial \text{input}}$

11.5.1 Plannar CNF

$$\mathbf{z}_k = f_\theta(\mathbf{z}_{k-1}) = \mathbf{z}_{k-1} + \mathbf{u}\mathbf{h}(\mathbf{w}^\top \mathbf{z}_{k-1} + \mathbf{b}) \tag{90}$$

the continuous version becomes:

$$\begin{aligned} f_\theta(\mathbf{z}(t)) &= \mathbf{u}\mathbf{h}(\mathbf{w}^\top \mathbf{z}(t) + \mathbf{b}) \\ \implies \frac{\partial \log p(\mathbf{z}(t))}{\partial t} &= -\mathrm{Tr}\Big(\mathbf{u}\frac{\partial h}{\partial \mathbf{z}}^\top\Big) \\ &= -\mathbf{u}^\top \frac{\partial h}{\partial \mathbf{z}} \end{aligned} \tag{91}$$

11.5.2 Hamiltonian CNF

$$\begin{aligned} &\begin{cases} \mathbf{z}_k^{(1)} &= \mathbf{z}_{k-1}^{(1)} \\ \mathbf{z}_k^{(2)} &= \mathbf{z}_{k-1}^{(2)} + f_{\theta^{(1)}}(\mathbf{z}_{k-1}^{(1)}) \end{cases} \\ &\qquad\qquad \text{in alternative steps:} \\ &\begin{cases} \mathbf{z}_{k'}^{(2)} &= \mathbf{z}_{k'-1}^{(2)} \\ \mathbf{z}_{k'}^{(1)} &= \mathbf{z}_{k'-1}^{(1)} + f_{\theta^{(2)}}(\mathbf{z}_{k'-1}^{(2)}) \end{cases} \end{aligned} \tag{92}$$

the continuous version becomes: (also for clarity, we have index:

$$\begin{cases} D_1 &= 1 : d_1 \\ D_2 &= d_1 + 1 : d_1 + d_2 \end{cases} \tag{93}$$

$$\begin{bmatrix} \frac{\mathrm{d}\mathbf{z}_{D_1}}{\mathrm{d}t} \\ \frac{\mathrm{d}\mathbf{z}_{D_2}}{\mathrm{d}t} \end{bmatrix} = \begin{bmatrix} f_{\theta^2}(\mathbf{z}_{D_2}) \\ f_{\theta^1}(\mathbf{z}_{D_1}) \end{bmatrix} \qquad \begin{cases} \mathbf{z}_{k'}^{(1)} &= \mathbf{z}_{k'-1}^{(1)} + f_{\theta^{(2)}}(\mathbf{z}_{k'-1}^{(2)}) \\ \mathbf{z}_k^{(2)} &= \mathbf{z}_{k-1}^{(2)} + f_{\theta^{(1)}}(\mathbf{z}_{k-1}^{(1)}) \end{cases} \tag{94}$$

therefore:

$$\begin{aligned} \frac{\partial f_{\theta^1}(\mathbf{z}_{D_1})}{\partial \mathbf{z}_{D_1}} &= \mathbf{0} \\ \frac{\partial f_{\theta^2}(\mathbf{z}_{D_2})}{\partial \mathbf{z}_{D_2}} &= \mathbf{0} \end{aligned} \tag{95}$$

Jacobian are zeros in diagonal, so trace is zero. Since it's volume preserving, therefore $\frac{\partial \log p(\mathbf{z}(t))}{\partial t} = 0$ no surprise!

12 Denoise Diffusion model

12.1 data distribution $q(\mathbf{x}_0)$

There is a lot of focus in [?] around denoising diffusion models that claim to replace generative adversarial networks.

Basically, the data distribution $q(\mathbf{x}_0)$ is the only real distribution of data. Note that the notation of p and q are swapped compared with what was in the other variational inference literature. I keep it consistent with the original paper.

Then the trick is although we have just the marginal distribution of real data, but we can extend its distribution to higher dimensional space, where $q(\mathbf{x}_{1:T}|\mathbf{x}_0)$ can be of any arbitrary distributions.

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \tag{96}$$

In here, any conditional density $q(\mathbf{x}_t|\mathbf{x}_{t-1})$ is going to work. But what is an appropriate distribution to extend them to? We have chosen it to be:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}\big(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}\big) \tag{97}$$

12.1.1 why $q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}\big(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}\big)$ works?

we can proof that:

$$\begin{aligned} q(\mathbf{x}_1|\mathbf{x}_0) &= \mathcal{N}\big(\mathbf{x}_1; \sqrt{1-\beta_1}\mathbf{x}_0, \beta_1\mathbf{I}\big) \\ q(\mathbf{x}_2|\mathbf{x}_1) &= \mathcal{N}\big(\mathbf{x}_2; \sqrt{1-\beta_2}\mathbf{x}_1, \beta_2\mathbf{I}\big) \end{aligned} \tag{98}$$

then:

$$\mathbf{x}_2|\mathbf{x}_0 = \sqrt{1-\beta_2}(\mathbf{x}_1|\mathbf{x}_0) + \mathbf{w}_t \qquad \mathbf{w}_t \sim (0, \beta_2\mathbf{I}) \tag{99}$$

using similar derivations from Kalman Filter's prediction (using moment matching):

$$\begin{aligned} \mathbf{x}_2|\mathbf{x}_0 \sim \mathcal{N}\big(&\underbrace{\sqrt{1-\beta_2}}_{\mathbf{A}}\underbrace{\sqrt{1-\beta_1}\mathbf{x}_0}_{\hat{\mu}_{t-1}}\ ,\ \underbrace{\sqrt{1-\beta_2}}_{\mathbf{A}}\ \underbrace{\beta_1\mathbf{I}}_{\hat{\Sigma}_{t-1}}\ \underbrace{\sqrt{1-\beta_2}}_{\mathbf{A}^\top} + \underbrace{\beta_2\mathbf{I}}_{\mathbf{Q}_t}\ \big) \\ &= \mathcal{N}\big(\sqrt{(1-\beta_1)(1-\beta_2)}\ \mathbf{x}_0\ ,\ \beta_1(1-\beta_2)\mathbf{I} + \beta_2\mathbf{I}\big) \\ &= \mathcal{N}\big(\sqrt{(1-\beta_1)(1-\beta_2)}\ \mathbf{x}_0\ ,\ (\beta_1 - \beta_1\beta_2 + \beta_2)\mathbf{I}\big) \\ &= \mathcal{N}\big(\sqrt{(1-\beta_1)(1-\beta_2)}\ \mathbf{x}_0\ ,\ \big(1 - (1 - \beta_1 - \beta_2 + \beta_1\beta_2)\mathbf{I}\big) \\ &= \mathcal{N}\big(\sqrt{(1-\beta_1)(1-\beta_2)}\ \mathbf{x}_0\ ,\ \big(1 - (1-\beta_1)(1-\beta_2)\big)\mathbf{I}\big) \end{aligned} \tag{100}$$

the reason of rewriting the last line into this form is so that we can better analyse its variance

12.1.2 Kalman Filter's prediction $p(\mathbf{x}_t|\mathbf{y}_{1:t-1})$

In Kalman Filter, we have: $p(x_{t-1}|y_{1:t-1}) = \mathcal{N}(\hat{\mu}_{t-1}, \hat{\Sigma}_{t-1})$ from the previous time $t-1$. Then in order to compute:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \int_{x_{t-1}} p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) \tag{101}$$

However, this is instead computed using moment matching techniques, where we let random variable $\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}$ be:

$$\begin{aligned} \mathbf{x}_{t-1}|\mathbf{y}_{1:t-1} &= \mathbb{E}[\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}] + \epsilon_{t-1} \qquad \epsilon_{t-1} \sim \mathcal{N}(0, \hat{\Sigma}_{t-1}) \\ &= \hat{\mu}_{t-1} + \epsilon_{t-1} \end{aligned} \tag{102}$$

it's important to know that random variable $\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}$ and $\mathbf{x}_t|\mathbf{y}_{1:t-1}$ are different ones.
and we attempt to write both ϵ_t in terms of ϵ_{t-1}:

$$\begin{aligned} \mathbf{x}_t|\mathbf{y}_{1:t-1} &= \mathbf{A}(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) + \mathbf{w}_t \qquad \mathbf{w}_t \sim \mathcal{N}(0, \mathbf{Q}_t) \\ &= \mathbf{A}(\hat{\mu}_{t-1} + \epsilon_{t-1}) + \mathbf{w}_t \\ &= \mathbf{A}\hat{\mu}_{t-1} + \mathbf{A}\epsilon_{t-1} + \mathbf{w}_t \end{aligned} \tag{103}$$

mean: $\bar{\mu}_t = \mathbb{E}[\mathbf{x}_t|\mathbf{y}_{1:t-1}]$:

$$\begin{aligned} \bar{\mu}_t &= \mathbb{E}[\mathbf{A}\hat{\mu}_{t-1} + \mathbf{A}\epsilon_{t-1} + \mathbf{w}_t] \\ &= \mathbf{A}\hat{\mu}_{t-1} \end{aligned} \tag{104}$$

covariance: $\bar{\Sigma}_t = \mathbb{VAR}[\mathbf{x}_t|\mathbf{y}_{1:t-1}]$

$$\begin{aligned} \bar{\Sigma}_t &= \mathbb{E}\big[(\mathbf{A}\hat{\mu}_{t-1} + \mathbf{A}\epsilon_{t-1} + \mathbf{w}_t)(\mathbf{A}\hat{\mu}_{t-1} + \mathbf{A}\epsilon_{t-1} + \mathbf{w}_t)^\top\big] \\ &= \mathbb{E}\big[(\mathbf{A}\epsilon_{t-1} + \mathbf{w}_t)(\mathbf{A}\epsilon_{t-1} + \mathbf{w}_t)^\top\big] \\ &= \mathbb{E}\big[(\mathbf{A}\epsilon_{t-1} + \mathbf{w}_t)(\epsilon_{t-1}^\top\mathbf{A}^\top + \mathbf{w}_t^\top)\big] \\ &= \mathbb{E}\big[\mathbf{A}\epsilon_{t-1}\epsilon_{t-1}^\top\mathbf{A}^\top\big] + \mathbb{E}[\mathbf{w}_t\mathbf{w}_t^\top] \qquad \mathbb{E}[\epsilon_{t-1}\mathbf{w}_t^\top] = \mathbf{0} \\ &= \mathbf{A}\mathbb{E}\big[\epsilon_{t-1}\epsilon_{t-1}^\top\big]\mathbf{A}^\top + \mathbb{E}[\mathbf{w}_t\mathbf{w}_t^\top] \\ &= \mathbf{A}\hat{\Sigma}_{t-1}\mathbf{A}^\top + \mathbf{Q}_t \end{aligned} \tag{105}$$

12.1.3 generalization and let $t \to \infty$

now looking at the variance of

$$\mathbf{x}_2|\mathbf{x}_0 \sim \mathcal{N}\big(\sqrt{(1-\beta_1)(1-\beta_2)}\ \mathbf{x}_0\ ,\ \big(1 - (1-\beta_1)(1-\beta_2)\big)\mathbf{I}\big) \tag{106}$$

if we generalize, we should have:

$$\begin{aligned} \mathbf{x}_t|\mathbf{x}_0 &\sim \mathcal{N}\Big(\sqrt{\prod_{s=1}^{t}(1-\beta_s)}\ \mathbf{x}_0\ ,\ \Big(1 - \prod_{s=1}^{t}(1-\beta_s)\Big)\mathbf{I}\Big) \\ &\sim \mathcal{N}\big(\mathbf{x}_t; \sqrt{\alpha_t}\mathbf{x}_0\ ,\ (1-\alpha_t)\mathbf{I}\big) \qquad \text{letting } \alpha_t = \prod_{s=1}^{t}(1-\beta_s) \end{aligned} \tag{107}$$

do you notice that $\mathbf{x}_t|\mathbf{x}_0$ is a Gaussian distribution
therefore, one need $\alpha_t \to 0$:

$$\mathbf{x}_t|\mathbf{x}_0 \sim \mathcal{N}\big(\mathbf{x}_t; \mathbf{0},\ \mathbf{I}\big) \tag{108}$$

which basically makes it standard Gaussian noise.

12.2 proposal $p_\theta(\mathbf{x}_0)$

we have the proposal (but it's written in $p_\theta(\mathbf{x}_0)$) this time:

$$p_\theta(\mathbf{x}_0) = \int p_\theta(\mathbf{x}_{0:T}) \mathrm{d}\mathbf{x}_{1:T} \tag{109}$$

It should also be extended to higher dimension $p_\theta(\mathbf{x}_{0:T})$ to match $q_{1:T}$.

The loss function is a lot more direct, in which we want to minimize the cross entropy between $q(\mathbf{x}_0)$ and $p_\theta(\mathbf{x}_0)$. However, we also bring the other dimension into it as this is where the $p_\theta(\mathbf{x}_T)$ and $q(\mathbf{x}_0)$ "meet". this is important.

$$\begin{aligned}
\mathbb{E}_{q(\mathbf{x}_0)}\big[-\log p_\theta(\mathbf{x}_0)\big] &= \int_{\mathbf{x}_0} -\log p_\theta(\mathbf{x}_0) q(\mathbf{x}_0)\mathrm{d}\mathbf{x}_0 \\
&= \int_{\mathbf{x}_0} -\log\Big(\int_{\mathbf{x}_{1:T}} p_\theta(\mathbf{x}_{0:T})\mathrm{d}\mathbf{x}_{1:T}\Big) q(\mathbf{x}_0)\mathrm{d}\mathbf{x}_0 \\
&= \int_{\mathbf{x}_0} -\log\Big(\int_{\mathbf{x}_{1:T}} \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} q(\mathbf{x}_{1:T}|\mathbf{x}_0)\mathrm{d}\mathbf{x}_{1:T}\Big) q(\mathbf{x}_0)\mathrm{d}\mathbf{x}_0 \\
&= \int_{\mathbf{x}_0} -\log\Big(\mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\Big[\frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\Big]\Big) q(\mathbf{x}_0)\mathrm{d}\mathbf{x}_0 \\
&\geq \int_{\mathbf{x}_0} \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\Big[-\log\Big(\frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\Big)\Big] q(\mathbf{x}_0)\mathrm{d}\mathbf{x}_0 \\
&= \mathbb{E}_{q(\mathbf{x}_{0:T})}\Big[-\log\Big(\frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\Big)\Big] \\
&= \mathbb{E}_{q(\mathbf{x}_{0:T})}\big[-\log p_\theta(\mathbf{x}_{0:T}) + \log q(\mathbf{x}_{1:T}|\mathbf{x}_0)\big] \\
&= \mathcal{L}(\theta)
\end{aligned} \tag{110}$$

this is different to traditional VAE approaches, where the proposal distribution is used in Monte-Carlo sampling. In here, the data (and its extension) distribution is used instead.

12.3 expressing $\mathcal{L}(\theta)$ in terms of KL

$$\begin{aligned}
\mathcal{L}(\theta) &= \mathbb{E}_q\Big[-\log\frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\Big] \\
&= \mathbb{E}_q\Big[-\log\frac{p_\theta(\mathbf{x}_0|\mathbf{x}_1)\ p_\theta(\mathbf{x}_1|\mathbf{x}_2)\dots p_\theta(\mathbf{x}_{T-1}|\mathbf{x}_T)\ p_\theta(\mathbf{x}_T)}{q(\mathbf{x}_1|\mathbf{x}_0)\ q(\mathbf{x}_2|\mathbf{x}_1)\dots q(\mathbf{x}_T|\mathbf{x}_{T-1})}\Big]
\end{aligned} \tag{111}$$

where $p_\theta(\mathbf{x}_T) = \mathcal{N}(\mathbf{0},\mathbf{I})$. note that we use forward and backwards dependencies for $p_\theta(\cdot)$ and $q(\cdot)$:

$$\begin{aligned}
\mathcal{L}(\theta) &= \mathbb{E}_q\Big[-\log(p_\theta(\mathbf{x}_T)) - \log\frac{p_\theta(\mathbf{x}_1|\mathbf{x}_2)\dots p_\theta(\mathbf{x}_{T-1}|\mathbf{x}_T)}{q(\mathbf{x}_2|\mathbf{x}_1)\dots q(\mathbf{x}_T|\mathbf{x}_{T-1})} - \log\frac{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_1|\mathbf{x}_0)}\Big]\\
&= \mathbb{E}_q\Big[-\log(p_\theta(\mathbf{x}_T)) - \log\frac{p_\theta(\mathbf{x}_1|\mathbf{x}_2)\dots p_\theta(\mathbf{x}_{T-1}|\mathbf{x}_T)}{q(\mathbf{x}_2|\mathbf{x}_1,\mathbf{x}_0)\dots q(\mathbf{x}_T|\mathbf{x}_{T-1},\mathbf{x}_0)} - \log\frac{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_1|\mathbf{x}_0)}\Big] \qquad \text{markov forward assumption}\\
&= \mathbb{E}_q\Big[-\log(p_\theta(\mathbf{x}_T)) - \sum_{t>1}\log\frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{x}_0)} - \log\frac{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_1|\mathbf{x}_0)}\Big]
\end{aligned} \tag{112}$$

looking at the term: $\log\frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{x}_0)}$. The problem is that the numerator and denominator are not about the same random variable, i.e., x_{t-1}. Therefore, we change $q(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{x}_0)$ in terms of $q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)$:

$$\begin{aligned}
q(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{x}_0) &= \frac{q(\mathbf{x}_t,\mathbf{x}_{t-1},\mathbf{x}_0)}{q(\mathbf{x}_{t-1},\mathbf{x}_0)}\\
&= \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}
\end{aligned} \tag{113}$$

substitute the above into main equation:

$$\begin{aligned}
\mathcal{L}(\theta) &= \mathbb{E}_q\Big[-\log(p_\theta(\mathbf{x}_T)) - \sum_{t>1}\log\frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)q(\mathbf{x}_t|\mathbf{x}_0)} - \log\frac{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_1|\mathbf{x}_0)}\Big]\\
&= \mathbb{E}_q\Big[-\log(p_\theta(\mathbf{x}_T)) - \sum_{t>1}\log\frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)} - \sum_{t>1}\log\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)} - \log\frac{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_1|\mathbf{x}_0)}\Big]\\
&= \mathbb{E}_q\Big[-\log(p_\theta(\mathbf{x}_T)) - \sum_{t>1}\log\frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)} - \underbrace{\Big(\sum_{t>1}\log q(\mathbf{x}_{t-1}|\mathbf{x}_0) - \log q(\mathbf{x}_t|\mathbf{x}_0)\Big)}_{\text{telescope sum}} - \log\frac{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_1|\mathbf{x}_0)}\Big]\\
&= \mathbb{E}_q\Big[-\log(p_\theta(\mathbf{x}_T)) - \sum_{t>1}\log\frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)} + \log q(\mathbf{x}_T|\mathbf{x}_0) - \log q(\mathbf{x}_1|\mathbf{x}_0) - \log\frac{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_1|\mathbf{x}_0)}\Big]\\
&= \mathbb{E}_q\Big[-\log\frac{p_\theta(\mathbf{x}_T)}{q(\mathbf{x}_T|\mathbf{x}_0)} - \sum_{t>1}\log\frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)} - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1)\Big]\\
&= \mathbb{KL}(q(\mathbf{x}_T\|p(\mathbf{x}_T)) - \sum_{t>1}\mathbb{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)\|p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)) - \mathbb{E}_q\Big[\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)\Big]
\end{aligned} \tag{114}$$

since they are KL between Gaussian, which can be computed efficiently.

12.3.1 alternative expression

let's look at the alternative expression, starting from Eq.(??):

$$\begin{aligned}
\mathcal{L}(\theta) &= \mathbb{E}_q\Big[-\log(p_\theta(\mathbf{x}_T)) - \log\frac{p_\theta(\mathbf{x}_1|\mathbf{x}_2)\dots p_\theta(\mathbf{x}_{T-1}|\mathbf{x}_T)}{q(\mathbf{x}_2|\mathbf{x}_1)\dots q(\mathbf{x}_T|\mathbf{x}_{T-1})} - \log\frac{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_1|\mathbf{x}_0)}\Big]\\
&= \mathbb{E}_q\Big[-\log(p_\theta(\mathbf{x}_T)) - \sum_{t\geq 1}\log\frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})}\Big]\\
&= \mathbb{E}_q\Big[-\log(p_\theta(\mathbf{x}_T)) - \sum_{t\geq 1}\log\frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)q(\mathbf{x}_{t-1})}{q(\mathbf{x}_{t-1}|\mathbf{x}_t)q(\mathbf{x}_t)}\Big]
\end{aligned} \tag{115}$$

the last line was because we let:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \frac{q(\mathbf{x}_t, \mathbf{x}_{t-1})}{q(\mathbf{x}_{t-1})} = \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t)q(\mathbf{x}_t)}{q(\mathbf{x}_{t-1})} \tag{116}$$

we have:

$$\begin{aligned}
&= \mathbb{E}_q\Big[-\log(p_\theta(\mathbf{x}_T)) - \sum_{t\geq 1}\log\frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t)} - \sum_{t\geq 1}\big(\log q(\mathbf{x}_{t-1}) - \log q(\mathbf{x}_t)\big)\Big]\\
&= \mathbb{E}_q\Big[-\log(p_\theta(\mathbf{x}_T)) - \sum_{t\geq 1}\log\frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t)} - \big(\log q(\mathbf{x}_0) - \log q(\mathbf{x}_T)\big)\Big]\\
&= \mathbb{E}_q\Big[-\log\frac{p_\theta(\mathbf{x}_T)}{q(\mathbf{x}_T)} - \sum_{t\geq 1}\log\frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t)} - \log q(\mathbf{x}_0)\Big]\\
&= \mathbb{E}_q\Big[\mathbb{KL}\big(q(\mathbf{x}_T)\|p_\theta(\mathbf{x}_T)\big) - \sum_{t\geq 1}\mathbb{KL}\big(q(\mathbf{x}_{t-1}|\mathbf{x}_t)\|p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)\big) - \log q(\mathbf{x}_0)\Big]
\end{aligned} \tag{117}$$

since the first and last term contains a single r.v:

$$\mathcal{L}(\theta) = \mathbb{KL}\big(q(\mathbf{x}_T)\|p_\theta(\mathbf{x}_T)\big) - \sum_{t\geq 1}\mathbb{E}_q\Big[\mathbb{KL}\big(q(\mathbf{x}_{t-1}|\mathbf{x}_t)\|p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)\big)\Big] - H(\mathbf{x}_0) \tag{118}$$

13 DDPM as a Stochastic Differential Equation

From the Discrete Forward Process (DDPM), we have:

$$x_t = \sqrt{1-\beta_t}\,x_{t-1} + \sqrt{\beta_t}\epsilon_t, \quad \epsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{119}$$

We can rewrite the discrete update rule in terms of a change in x, and let $t \to t+\Delta t$ and $t-1 \to t$:

$$\begin{aligned}
x_{t+\Delta t} &= (\sqrt{1-\beta_{t+\Delta t}})x_t + \sqrt{\beta_{t+\Delta t}}\epsilon_t\\
x_{t+\Delta t} - x_t &= (\sqrt{1-\beta_{t+\Delta t}} - 1)x_t + \sqrt{\beta_{t+\Delta t}}z_t
\end{aligned} \tag{120}$$

1. Let us define a continuous noise function $\beta(t)$ such that the discrete:

$$\beta_{t+\Delta t} \approx \beta(t+\Delta t)\Delta t \tag{121}$$

2. Now, let's look at the term $\sqrt{1-\beta_{t+\Delta t}}$. Using the Taylor series expansion $\sqrt{1-x} \approx 1 - \frac{1}{2}x$ for small x:

$$\sqrt{1-\beta(t)\Delta t} \approx 1 - \frac{1}{2}\beta(t)\Delta t \tag{122}$$

Substitute this back into our difference equation:

$$\begin{aligned}
x_{t+\Delta t} - x_t &\approx \left((1 - \frac{1}{2}\beta(t)\Delta t) - 1\right)x_t + \sqrt{\beta(t)\Delta t}\epsilon_t\\
&= -\frac{1}{2}\beta(t)x_t\Delta t + \sqrt{\beta(t)}\sqrt{\Delta t}\epsilon_t
\end{aligned} \tag{123}$$

By definition of the Wiener process:

$$dw_t \sim \mathcal{N}(0, dt) \implies dw_t \approx \sqrt{\Delta t}\epsilon_t \text{ where } \epsilon_t \sim \mathcal{N}(0,1) \tag{124}$$

Substituting this into our equation:

$$dx_t = -\frac{1}{2}\beta(t)x_t dt + \sqrt{\beta(t)}dw_t \tag{125}$$

The equation above is the specific SDE for the "Variance Preserving" (VP) SDE used in DDPM. If we look at the general form requested:

$$\mathrm{d}\mathbf{x}_t = f(\mathbf{x}_t, t)\mathrm{d}t + g(t)\mathrm{d}w_t \tag{126}$$

or we may use the notation $\mathbf{v}(\mathbf{x}_t, t)$ instead of $f(\mathbf{x}_t, t)$ to indicate the velocity field:

$$\mathrm{d}\mathbf{x}_t = v(\mathbf{x}_t, t)\mathrm{d}t + g(t)\mathrm{d}w_t \tag{127}$$

By comparing terms, we identify:

- Drift Coefficient (f or v):

  $$f(x_t, t) = -\frac{1}{2}\beta(t)x_t \tag{128}$$

  This term pulls the data towards zero (mean reversion), since $\beta(t) > 0 \quad \forall t$.

  – Case 1: x is positive ($x > 0$): the velocity $f(x_t, t)$ is negative.
  – Case 2: x is negative ($x < 0$): the velocity $f(x_t, t)$ is positive.
  – Case 3: x is zero ($x = 0$): the velocity $f(x_t, t)$ is zero.

  therefore, counteracting the expansion of variance caused by the noise.

- Diffusion Coefficient (g):

  $$g(t) = \sqrt{\beta(t)} \tag{129}$$

  This term controls the intensity of the noise being injected.

14 Flow matching

The flow matching approach is to find a velocity field $v(x_t, t)$ that matches the data distribution. It can be simply formulating the Eq.(??) by removing the stochastic term:

$$\mathrm{d}\mathbf{x}_t = v(\mathbf{x}_t, t)\,\mathrm{d}t \tag{130}$$

Then, for inference (or decoding), we can solve the ODE:

$$\underbrace{\mathbf{x}_1}_{\text{clean image}} = \underbrace{\mathbf{x}_0}_{\text{noise}} + \int_0^1 v(\mathbf{x}_t, t)\,\mathrm{d}t \tag{131}$$

In order to make the two apporaches comparable, we standardize the notation to use $t = 1$ as the clean image and $t = 0$ as the noise. (this is mainly because of making sense for the flow matching inference $x_1 = x_0 + \int_0^1 v(x_t, t)dt$.)

Diffusion Models

Diffusion Equation

$$\text{SDE: } \mathrm{d}\mathbf{x}_t = v(\mathbf{x}_t, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}_t$$

TRAINING

- $\mathbf{x}_1 \sim \text{data}$
- $t \sim U(0, 1)$
- $\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_1 + \sqrt{1 - \alpha_t}\epsilon$
- $\hat{\epsilon}_\theta = v(\mathbf{x}_t, t)$
- $\mathcal{L} = ||\hat{\epsilon}_\theta - \epsilon||_2^2$

INFERENCE

- $x_0 \sim \mathcal{N}(0, 1)$
- in steps $t : 0 \to 1$, solve the reverse SDE

Flow-Matching

Diffusion Equation

$$\text{ODE: } \mathrm{d}\mathbf{x}_t = v(\mathbf{x}_t, t)\mathrm{d}t$$

TRAINING

- $\mathbf{x}_1 \sim \text{data}$
- $t \sim U(0, 1) \quad \mathbf{x}_0 \sim \mathcal{N}(0, 1)$
- $\mathbf{x}_t = (1 - t)\mathbf{x}_0 + t\mathbf{x}_1$
- $v = \mathbf{x}_1 - \mathbf{x}_0$
- $\hat{v}_\theta = v(\mathbf{x}_t, t)$
- $\mathcal{L} = ||\hat{v}_\theta - v||_2^2$

INFERENCE

- $\mathbf{x}_0 \sim \mathcal{N}(0, 1)$
- solve integral:

$$\mathbf{x}_1 = \mathbf{x}_0 + \int_0^1 v(\mathbf{x}_t, t)\mathrm{d}t$$

Here is the question then, what is the denosing step?

14.1 the reverse SDE

we also stated that the corresponding SDE of DPMM [?] is:

$$\mathrm{d}\mathbf{x} = \underbrace{-\frac{1}{2}\beta(t)\mathbf{x}\mathrm{d}t}_{\mathbf{f}(\mathbf{x},t)}\,\mathrm{d}t + \underbrace{\sqrt{\beta(t)}}_{g(t)}\,\mathrm{d}\mathbf{w}, \tag{132}$$

Importantly, any SDE has a corresponding reverse SDE whose closed form is given by (we will see later it requires $g(t)$ is independent from $\mathbf{x}$):

$$\mathrm{d}\mathbf{x} = [\mathbf{f}(\mathbf{x}, t) - g^2(t)\nabla_{\mathbf{x}} \log p_t(\mathbf{x})]\mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{w}}. \tag{133}$$

Here $\mathrm{d}t$ represents a negative infinitesimal time step, and $\bar{\mathbf{w}}$ represents a standard Wiener process. The SDE needs to be solved backwards in time (from $t = T$ to $t = 0$). In order to compute the reverse SDE, we need to estimate $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$, which is exactly the score function of $p_t(\mathbf{x})$. We can learn the score function which is parameterized by θ, such that $\mathbf{s}_\theta(\mathbf{x}, t) \approx \nabla_{\mathbf{x}} \log p_t(\mathbf{x})$. This needs to be done before solving the reverse SDE.

$$\mathrm{d}\mathbf{x} = [\mathbf{f}(\mathbf{x}, t) - g^2(t)\mathbf{s}_\theta(\mathbf{x}, t)]\mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{w}} \tag{134}$$

14.1.1 How to generate sample

The goal of generative models is to produce samples. We commence with x_T drawn from $p_T(\cdot)$, which serves as the initial prior distribution by design, allowing us to tailor it according to any chosen Stochastic Differential Equation (SDE). Also keep in mind that p_0 is the data distribution. We can then solve the reverse SDE to obtain a sample $\mathbf{x}_0$.

Of course, we can do so, only if the score function of each marginal distribution, $\nabla x \log p_t(\mathbf{x})$, is known for all t, i.e., we need to train $\mathbf{s}_\theta(\mathbf{x}, t)$ well.

14.2 Some mathematical foundation for Reverse SDE Eq.(??)

The reverse SDE is from the same paper using Eq.(??). A summary of workflow can be described as follows:

1. The conventional SDE is expressed in drift-diffusion paradim: $\mathrm{d}X_t = \mu(X_t, t)\mathrm{d}t + \sigma(X_t, t)\mathrm{d}W_t$, where W_t represents the standard Wiener process.

2. An Ito Lemma provides an additional drift-diffusion SDE when applied to a function f, i.e., for $f(X_t)$:

$$\mathrm{d}f(X_t) = \left(\mu_t \frac{\partial f}{\partial x} + \frac{1}{2}\sigma_t^2 \frac{\partial^2 f}{\partial x^2}\right)\mathrm{d}t + \sigma_t \frac{\partial f}{\partial x}\mathrm{d}W_t \tag{135}$$

   the above two items will be described in detail by the rest of the report from section ?? onwards.

3. Then, the Fokker-Planck equation describes the temporal evolution of the probability density function (PDF) for X_t (given the correponding μ_t and σ_t) which is analogous to the Kolmogorov forward equation.

$$\partial_t p(x_t) = -\,\partial_{x_t}\left[\mu(x_t)p(x_t)\right] + \frac{1}{2}\partial_{x_t}^2\left[\sigma^2(x_t)p(x_t)\right] \tag{136}$$

   The derivation of the Fokker-Planck equation is given at [2]

4. The Kolmogorov backward equation establishes the inverse relationship for the PDF. for $s \geq t$ is defined as

$$-\partial_t p(x_s|x_t) = \mu(x_t)\ \partial_{x_t} p(x_s|x_t) + \frac{1}{2}\ \sigma^2(x_t)\ \partial_{x_t}^2 p(x_s|x_t) \tag{137}$$

   the derivation of the Kolmogorov backward equation is given at [3], or from this YouTube video [4].

5. Using both the Fokker-Planck equation (forward) with the Kolmogorov backward (backward) equation, in the paper, "Reverse-time diffusion equation models", the authors derived an equation for the reverse SDE, which incorporates the score function into its formulation. This is the general form.

$$dX_\tau = \left(-\mu(x_{1-\tau}) + \frac{1}{p(x_{1-\tau})}\partial_{x_{1-\tau}}\left[\sigma^2(x_{1-\tau})\ p(x_{1-\tau})\right]\right)d\tau + \sigma(x_{1-\tau})dW_\tau \tag{138}$$

   By keeping the $\sigma(x_t)$ as constant and independent of x_t (but still dependant on t) and applying the log-derivative trick, the drift simplifies to:

[2] https://ludwigwinkler.github.io/blog/FokkerPlanck/

[3] https://ludwigwinkler.github.io/blog/Kramers/

[4] https://www.youtube.com/watch?v=wrvHHNCRl7I

$$\mathrm{d}X_\tau = \Big(-\mu(x_{1-\tau}) + \frac{1}{p(x_{1-\tau})}\partial_{x_{1-\tau}}\big[\overbrace{\sigma^2(x_{1-\tau})}^{\to\,\sigma^2(1-\tau)}\ p(x_{1-\tau})\big]\Big)d\tau + \overbrace{\sigma(x_{1-\tau})}^{\to\,\sigma(1-\tau)}\,\mathrm{d}W_\tau \tag{139}$$

$$= \left(-\mu(x_{1-\tau}) + \frac{\sigma^2(1-\tau)}{p(x_{1-\tau})}\partial_{x_{1-\tau}}\ p(x_{1-\tau})\right) d\tau + \sigma(1-\tau)\mathrm{d}W_\tau \tag{140}$$

$$= \Big(-\mu(x_{1-\tau}) + \sigma^2(1-\tau)\partial_{x_{1-\tau}}\ \log p(x_{1-\tau})\Big)\mathrm{d}\tau + \sigma(1-\tau)\mathrm{d}W_\tau \tag{141}$$

compare it with $\mathrm{d}\mathbf{x} = [\mathbf{f}(\mathbf{x},t) - g^2(t)\mathbf{s}_\theta(\mathbf{x},t)]\mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{w}}$, we see that we allow $\mathrm{d}\tau$ to be the positive infinitesimal time step, and $\mathrm{d}t$ to be the negative infinitesimal time step. We also express $\sigma(1-\tau) \equiv g(t)$

6. Then, As a result, many papers, [?] have developed a method to estimate the score function, which enables the determination of $X(0)$ by resolving the SDE from a given sample of $X(T)$.

15 discussion on contrastive-ness in p and q

the same tricks have been used in GAN, AVB, and NCE

- in GAN, since we already "operationalised":

$$\max_D \mathbb{E}_{\mathbf{x}\sim p_{\mathrm{r}}(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{x}\sim p_{\mathrm{g}}^{\theta}(\mathbf{x})}[\log(1-D(\mathbf{x})] \tag{142}$$

  the fact that $D^*(x) = \frac{p_{\mathrm{r}}(x)}{p_{\mathrm{r}}(x)+p_{\mathrm{g}}^{\theta}(x)}$ is mainly gives us an explanation of what $\max_D$ aims to obtain, in GAN however, such property is not practically important

- however, in AVB, by proving that the maximization of:

$$\max_T \mathbb{E}_{p(\mathbf{x})}\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\Big[\log\sigma(T(\mathbf{x},\mathbf{z}))\Big] + \mathbb{E}_{p(\mathbf{x})}\mathbb{E}_{p(\mathbf{z})}\Big[\log(1-\sigma(T(\mathbf{x},\mathbf{z})))\Big] \tag{143}$$

  then immediately, it is realized that, with some abuse of notation:

$$\begin{aligned}\sigma(T_\psi^*(\mathbf{x},\mathbf{z})) &= \frac{p(\mathbf{x})q_\phi(\mathbf{z}|\mathbf{x})}{p(\mathbf{x})q_\phi(\mathbf{z}|\mathbf{x})+p(\mathbf{x})p(\mathbf{z})} = \frac{q_\phi(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})+p(\mathbf{z})}\\ \implies T_\psi^*(\mathbf{x},\mathbf{z}) &= \log(q_\phi(\mathbf{z}|\mathbf{x})) - \log p(\mathbf{z})\end{aligned} \tag{144}$$

- the moral of the story is that instead of $\max_D f(D)$, we perform $\max_T f(D(T))$:

$$\begin{aligned}D(T^*(p,q)) &= \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}[\log D(T(p(\mathbf{x}),q(\mathbf{x})))] + \mathbb{E}_{\mathbf{x}\sim q(\mathbf{x})}\left[\log(1-D(T(p(\mathbf{x}),q(\mathbf{x}))))\right]\\ &= \frac{p(\mathbf{x})}{p(\mathbf{x})+q(\mathbf{x})}\end{aligned} \tag{145}$$

  then, we can choose some $D:\mathbf{R}\to[0,\ldots 1]$, for example, $\sigma(\cdot)$, and obtain get $T^*(p(\mathbf{x}),q(\mathbf{x}))$ into a desirable quantity

- the same applies to NCE, where the operationalised objective is:

$$\begin{aligned}\theta^* &= \arg\max_\theta \mathbb{E}_{\mathbf{w}\sim p(\mathbf{w})}\Big[\log\Big(\sigma\big(\mathbf{w}^\top\theta - \log\big[kq(\mathbf{w})\big]\big)\Big)\Big] + k\mathbb{E}_{\mathbf{w}\in q(\mathbf{w})}\log\Big[1-\sigma\big(\mathbf{w}^\top\theta - \log\big[kq(\mathbf{w})\big]\big)\Big]\\ \implies &\Big(\sigma\big(\mathbf{w}^\top\theta^* - \log\big[kq(\mathbf{w})\big]\big)\Big) = \frac{p}{p+kq}\\ \implies &\frac{\exp(\mathbf{w}^\top\theta)}{\exp(\mathbf{w}^\top\theta)+kq(\mathbf{w})} = \frac{p}{p+kq}\\ \implies &p(\mathbf{w}) = \exp(\mathbf{w}^\top\theta)\end{aligned} \tag{146}$$

  in here $D(T^*(p,q)) = \sigma\big(\mathbf{w}^\top\theta^* - \log\big[kq(\mathbf{w})\big]\big)$