Deep Generative Models

Fall 2025

Sharif University of Technology

Alireza Mirrokni - 401106617

Problem Set 2

Problem 1: Conditional Variational Autoencoder (CVAE)

1. To properly model the conditional distribution for the entire dataset, we define the distance criterion as the expected Kullback-Leibler (KL) divergence over the distribution of the conditioning variable y. That is, we want to minimize the average divergence between the data distribution $p_{\text{data}}(x|y)$ and the model distribution $p_\theta(x|y)$ weighted by how often each y appears:

$$\mathcal{L} = \mathbb{E}_{y\sim p_{\text{data}}(y)}\left[D_{\text{KL}}(p_{\text{data}}(x|y)\,||\,p_\theta(x|y))\right]$$

We can write this explicitly by expanding the expectation and the definition of the KL divergence:

$$\begin{aligned}\mathcal{L} &= \int p_{\text{data}}(y)\left(\int p_{\text{data}}(x|y)\log\left(\frac{p_{\text{data}}(x|y)}{p_\theta(x|y)}\right)dx\right)dy \\ &= \iint p_{\text{data}}(y)p_{\text{data}}(x|y)\log\left(\frac{p_{\text{data}}(x|y)}{p_\theta(x|y)}\right)dx\,dy\end{aligned}$$

Using the chain rule of probability, we know that the joint distribution is the product of the prior and the conditional: $p_{\text{data}}(x,y) = p_{\text{data}}(y)p_{\text{data}}(x|y)$. Substituting this into the integral, we obtain:

$$\mathcal{L} = \iint p_{\text{data}}(x,y)\log\left(\frac{p_{\text{data}}(x|y)}{p_\theta(x|y)}\right)dx\,dy$$

This integral is exactly the definition of the expectation over the joint distribution:

$$\mathcal{L} = \mathbb{E}_{(x,y)\sim p_{\text{data}}(x,y)}\left[\log\left(\frac{p_{\text{data}}(x|y)}{p_\theta(x|y)}\right)\right]$$

To derive the optimization objective, we expand the logarithmic term:

$$\mathcal{L} = \mathbb{E}_{(x,y)\sim p_{\text{data}}}[\log p_{\text{data}}(x|y)] - \mathbb{E}_{(x,y)\sim p_{\text{data}}}[\log p_\theta(x|y)]$$

The first term depends only on the true data distribution and is constant with respect to the model parameters θ (it is the conditional entropy of the data). Therefore, minimizing the distance $\mathcal{L}$ is equivalent to maximizing the second term, which is the expected log-likelihood.

Given a dataset of N pairs $\{(x_i,y_i)\}_{i=1}^N$ drawn from $p_{\text{data}}(x,y)$, we approximate this expectation using the Monte Carlo estimator (sample mean):

$$\text{Maximize } J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\log p_\theta(x_i|y_i)$$

2. We assume the data generation process involves a latent variable z. According to the laws of probability (specifically the sum rule or marginalization), we can express the marginal likelihood $p_\theta(x|y)$ by integrating the joint distribution $p_\theta(x, z|y)$ over all possible values of z. Using the chain rule of probability, the joint distribution decomposes into the likelihood (decoder) $p_\theta(x|y, z)$ and the prior $p_\theta(z|y)$. The integral representation is:

$$p_\theta(x|y) = \int p_\theta(x, z|y)\, dz = \int p_\theta(x|y, z)p_\theta(z|y)\, dz$$

This equation states that the probability of observing x given y is the weighted average of the probability of observing x given y and z, weighted by the probability of z given y.

3. We wish to introduce an arbitrary conditional distribution $q(z|x, y)$ into the integral. Assuming the support of $q(z|x, y)$ covers the support of the posterior, we can multiply and divide the integrand by $q(z|x, y)$:

$$p_\theta(x|y) = \int p_\theta(x, z|y)\frac{q(z|x, y)}{q(z|x, y)}\, dz$$

We can rearrange the terms to group the ratio:

$$p_\theta(x|y) = \int q(z|x, y)\left(\frac{p_\theta(x, z|y)}{q(z|x, y)}\right)\, dz$$

By definition, an integral of a function multiplied by a probability density function is the expectation of that function under that distribution. Thus:

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

Therefore, the expression for A inside the mathematical expectation is:

$$A = \frac{p_\theta(x, z|y)}{q(z|x, y)} = \frac{p_\theta(z|x, y)}{q(z|x, y)}p_\theta(x|y)$$

4. We start with the log-marginal likelihood and substitute the expectation derived in the previous step:

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

We apply Jensen's Inequality, which states that for a concave function ψ (such as the logarithm) and a random variable X, $\psi(\mathbb{E}[X]) \geq \mathbb{E}[\psi(X)]$. Applying this:

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

This lower bound is known as the Evidence Lower Bound (ELBO). We can further expand the term inside the expectation to separate the reconstruction loss and the regularization term:

$$\begin{aligned}
\text{ELBO}(x, y; \theta, q) &= \mathbb{E}_{z\sim q(z|x,y)}[\log p_\theta(x|y, z) + \log p_\theta(z|y) - \log q(z|x, y)] \\
&= \mathbb{E}_{z\sim q(z|x,y)}[\log p_\theta(x|y, z)] + \mathbb{E}_{z\sim q(z|x,y)}\left[\log\frac{p_\theta(z|y)}{q(z|x, y)}\right] \\
&= \mathbb{E}_{z\sim q(z|x,y)}[\log p_\theta(x|y, z)] - D_{\text{KL}}(q(z|x, y)\,||\,p_\theta(z|y))
\end{aligned}$$

Here, the first term represents the expected reconstruction quality, and the second term (the KL divergence) acts as a regularizer forcing the approximate posterior q to be close to the prior p (this would be important if the model is going to be used for generating new samples).

5. Jensen's inequality becomes an equality if and only if the variable inside the expectation is constant with respect to the distribution being averaged over. In our case, this requires $\frac{p_\theta(x,z|y)}{q(z|x,y)} = c$ for some constant c. This implies that $q(z|x,y)$ must be proportional to the joint distribution $p_\theta(x,z|y)$. The distribution that satisfies this proportionality and sums to 1 is the true posterior distribution:

$$q^*(z|x,y) = p_\theta(z|x,y) = \frac{p_\theta(x|y,z)p_\theta(z|y)}{p_\theta(x|y)}$$

When $q(z|x,y)$ is the true posterior, the bound is tight. The measure of closeness is defined by the difference between the true log-likelihood and the ELBO. We can derive this difference analytically:

$$\begin{aligned}
\log p_\theta(x|y) - \text{ELBO} &= \log p_\theta(x|y) - \mathbb{E}_q\left[\log \frac{p_\theta(x,z|y)}{q(z|x,y)}\right] \\
&= \mathbb{E}_q[\log p_\theta(x|y)] - \mathbb{E}_q\left[\log \frac{p_\theta(z|x,y)p_\theta(x|y)}{q(z|x,y)}\right] \\
&= \mathbb{E}_q\left[\log p_\theta(x|y) - \log p_\theta(x|y) - \log \frac{p_\theta(z|x,y)}{q(z|x,y)}\right] \\
&= \mathbb{E}_q\left[\log \frac{q(z|x,y)}{p_\theta(z|x,y)}\right] \\
&= D_{\text{KL}}(q(z|x,y) \,||\, p_\theta(z|x,y))
\end{aligned}$$

Thus, the measure of closeness is the KL divergence between the approximate posterior q and the true posterior.

6. The optimization problem for finding the best variational distribution q for a specific data point (x,y) is:

$$\min_q D_{\text{KL}}(q(z|x,y) \,||\, p_\theta(z|x,y))$$

There are two main challenges in solving this directly:

1. Intractability of the True Posterior: The fundamental challenge is that the target distribution, the true posterior $p_\theta(z|x,y)$, contains the marginal likelihood $p_\theta(x|y)$ in its denominator:

$$p_\theta(z|x,y) = \frac{p_\theta(x|y,z)p_\theta(z|y)}{\int p_\theta(x|y,z)p_\theta(z|y)\,dz}$$

Computing this integral (the evidence) is analytically intractable and computationally expensive (exponential complexity) for complex, high-dimensional models like neural networks. This makes it impossible to evaluate the KL divergence directly because we cannot compute the target distribution.

2. Reformulation and Scalability: To bypass the intractability of the evidence, we look at the relationship derived in the previous parts:

$$\log p_\theta(x|y) = \text{ELBO}(x,y;\theta,q) + D_{\text{KL}}(q(z|x,y) \,||\, p_\theta(z|x,y))$$

Crucially, for a fixed data point (x, y) and fixed model parameters θ, the marginal likelihood $\log p_\theta(x|y)$ is a constant. Therefore, minimizing the KL divergence with respect to q is mathematically equivalent to maximizing the ELBO. Since the ELBO does not require computing the intractable evidence integral, this optimization is feasible.

However, a new challenge arises regarding how to parameterize q. We could technically assign a separate set of variational parameters λ_i for each data point (x_i, y_i) in our dataset, denoted as $q(z|x_i, y_i; \lambda_i)$. We would then optimize λ_i specifically for that sample.

This approach is not scalable for two reasons:

- The number of parameters to optimize grows linearly with the dataset size N (if N is millions, we have millions of parameter sets λ_i).
- Inference for a new, unseen data point would require running a full iterative optimization loop from scratch to find its optimal λ, which is too slow for real-time applications.

To solve these challenges, Variational Autoencoders use Amortized Inference. Instead of optimizing a separate variational distribution parameter for every single data point, we learn a single global function (a neural network) parameterized by ϕ, called the inference network or encoder:

$$z \sim q_\phi(z|x, y)$$

This network takes x and y as input and outputs the distribution parameters (e.g., mean μ and variance σ^2) for q. We then optimize the ELBO jointly over θ (generative parameters) and ϕ (variational parameters) using Stochastic Gradient Descent. This "amortizes" the cost of inference across the entire dataset, allowing efficient computation for both training data and new samples.

7. When a new data pair $(x_{\text{new}}, y_{\text{new}})$ is provided, we compute the ELBO using the trained encoder network and the decoder network. The process is as follows:

   (a) Feed $(x_{\text{new}}, y_{\text{new}})$ into the encoder network q_ϕ to obtain the distribution parameters (e.g., $\mu_{\text{new}}, \sigma_{\text{new}}$).

   (b) Sample L latent variables $z^{(l)}$ from this distribution using the reparameterization trick: $z^{(l)} = \mu_{\text{new}} + \sigma_{\text{new}} \odot \epsilon^{(l)}$, where $\epsilon^{(l)} \sim \mathcal{N}(0, I)$.

   (c) Compute the empirical estimate of the ELBO:

$$\text{ELBO} \approx \frac{1}{L} \sum_{l=1}^{L} \left(\log p_\theta(x_{\text{new}}|y_{\text{new}}, z^{(l)}) \right) - D_{\text{KL}}(q_\phi(z|x_{\text{new}}, y_{\text{new}}) \,||\, p_\theta(z|y_{\text{new}}))$$

   If we strictly want to approximate the marginal likelihood $p_\theta(x_{\text{new}}|y_{\text{new}})$ (and not just its lower bound), the ELBO is often an underestimate. A more accurate method for likelihood approximation in this context is Importance Sampling using the variational posterior as the proposal distribution. The Importance Weighted estimator is:

$$p_\theta(x_{\text{new}}|y_{\text{new}}) \approx \frac{1}{K} \sum_{k=1}^{K} \frac{p_\theta(x_{\text{new}}|y_{\text{new}}, z^{(k)}) p_\theta(z^{(k)}|y_{\text{new}})}{q_\phi(z^{(k)}|x_{\text{new}}, y_{\text{new}})}$$

   where $z^{(k)}$ are samples drawn from $q_\phi(z|x_{\text{new}}, y_{\text{new}})$.

8. To generate a sample image corresponding to a new text description y_{new}, we utilize the generative part of the CVAE (the decoder) and the prior, independent of the encoder. The process is:

   (a) Sample a latent vector z from the prior distribution conditioned on y. In many standard CVAE implementations, the prior is assumed to be a standard normal distribution independent of the input, i.e., $z \sim \mathcal{N}(0, I)$. If the model uses a conditional prior $p_\theta(z|y)$, we sample from that distribution instead.

   (b) Concatenate or combine the sampled z with the condition y_{new}.

   (c) Pass this combination into the decoder network $p_\theta(x|y, z)$.

   (d) The decoder outputs the parameters of the distribution for x (e.g., pixel probabilities for Bernoulli, or means for Gaussian).

   (e) To visualize the result, we either take the mean of this output distribution or draw a sample from it to obtain the generated image x_{gen}.

Problem 2: Cauchy–Schwarz Divergence

1. Let $p(x) = \mathcal{N}(x; \mu_1, \sigma_1^2)$ and $q(x) = \mathcal{N}(x; \mu_2, \sigma_2^2)$. The Cauchy–Schwarz divergence is defined as:

$$D_{CS}(p\|q) = -\log \frac{\int p(x)q(x)\,dx}{\sqrt{\int p(x)^2\,dx \int q(x)^2\,dx}}$$

   To compute this, we first establish a key property regarding the integral of the product of two Gaussian probability density functions. We want to prove that:

$$\int \mathcal{N}(x; \mu_1, \sigma_1^2)\mathcal{N}(x; \mu_2, \sigma_2^2)\,dx = \mathcal{N}(\mu_1; \mu_2, \sigma_1^2 + \sigma_2^2)$$

   Let

$$I = \int \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(x-\mu_1)^2}{2\sigma_1^2}} \cdot \frac{1}{\sqrt{2\pi}\sigma_2} e^{-\frac{(x-\mu_2)^2}{2\sigma_2^2}}\,dx.$$

   We combine the exponents into a single term E:

$$E = -\frac{1}{2}\left[\frac{(x-\mu_1)^2}{\sigma_1^2} + \frac{(x-\mu_2)^2}{\sigma_2^2}\right]$$

   To solve the integral, we complete the square for x. We use the identity for the sum of two quadratic forms:

$$\frac{(x-\mu_1)^2}{\sigma_1^2} + \frac{(x-\mu_2)^2}{\sigma_2^2} = \frac{(x-\bar{\mu})^2}{\bar{\sigma}^2} + \frac{(\mu_1-\mu_2)^2}{\sigma_1^2+\sigma_2^2}$$

   where the combined variance $\bar{\sigma}^2$ and the combined mean $\bar{\mu}$ are defined as:

$$\bar{\sigma}^2 = \left(\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}\right)^{-1} = \frac{\sigma_1^2\sigma_2^2}{\sigma_1^2+\sigma_2^2}, \quad \bar{\mu} = \bar{\sigma}^2\left(\frac{\mu_1}{\sigma_1^2} + \frac{\mu_2}{\sigma_2^2}\right) = \frac{\mu_1\sigma_2^2 + \mu_2\sigma_1^2}{\sigma_1^2+\sigma_2^2}$$

Substituting this back into the exponent E:

$$E = -\frac{(x-\bar{\mu})^2}{2\bar{\sigma}^2} - \frac{(\mu_1-\mu_2)^2}{2(\sigma_1^2+\sigma_2^2)}$$

Now we substitute E back into the integral I:

$$\begin{aligned} I &= \int \frac{1}{2\pi\sigma_1\sigma_2} \exp\left(-\frac{(x-\bar{\mu})^2}{2\bar{\sigma}^2} - \frac{(\mu_1-\mu_2)^2}{2(\sigma_1^2+\sigma_2^2)}\right) dx \\ &= \frac{1}{2\pi\sigma_1\sigma_2} \exp\left(-\frac{(\mu_1-\mu_2)^2}{2(\sigma_1^2+\sigma_2^2)}\right) \underbrace{\int \exp\left(-\frac{(x-\bar{\mu})^2}{2\bar{\sigma}^2}\right) dx}_{\text{Gaussian integral kernel}} \end{aligned}$$

The integral is of an unnormalized Gaussian, which evaluates to $\sqrt{2\pi}\bar{\sigma} = \sqrt{2\pi}\frac{\sigma_1\sigma_2}{\sqrt{\sigma_1^2+\sigma_2^2}}$. Substituting this back:

$$\begin{aligned} I &= \frac{1}{2\pi\sigma_1\sigma_2} \exp\left(-\frac{(\mu_1-\mu_2)^2}{2(\sigma_1^2+\sigma_2^2)}\right) \cdot \sqrt{2\pi}\frac{\sigma_1\sigma_2}{\sqrt{\sigma_1^2+\sigma_2^2}} \\ &= \frac{1}{\sqrt{2\pi(\sigma_1^2+\sigma_2^2)}} \exp\left(-\frac{(\mu_1-\mu_2)^2}{2(\sigma_1^2+\sigma_2^2)}\right) \end{aligned}$$

This is exactly the definition of $\mathcal{N}(\mu_1; \mu_2, \sigma_1^2+\sigma_2^2)$.

With this result proven, we can now compute the terms for D_{CS}. The numerator term is simply the result we just derived:

$$\int p(x)q(x)\,dx = \frac{1}{\sqrt{2\pi(\sigma_1^2+\sigma_2^2)}} \exp\left(-\frac{(\mu_1-\mu_2)^2}{2(\sigma_1^2+\sigma_2^2)}\right)$$

For the denominator, we compute $\int p(x)^2\,dx$ by setting $\mu_2 = \mu_1$ and $\sigma_2 = \sigma_1$ in our formula. The exponential term becomes $e^0 = 1$:

$$\int p(x)^2\,dx = \frac{1}{\sqrt{2\pi(2\sigma_1^2)}} = \frac{1}{2\sigma_1\sqrt{\pi}}$$

Similarly, for $q(x)$:

$$\int q(x)^2\,dx = \frac{1}{2\sigma_2\sqrt{\pi}}$$

The product inside the square root of the denominator is:

$$\int p(x)^2\,dx \int q(x)^2\,dx = \frac{1}{4\pi\sigma_1\sigma_2}$$

Taking the square root:

$$\sqrt{\int p(x)^2\,dx \int q(x)^2\,dx} = \frac{1}{2\sqrt{\pi\sigma_1\sigma_2}}$$

We now form the ratio inside the logarithm for D_{CS}:

$$\frac{\int pq}{\sqrt{\int p^2 \int q^2}} = \frac{2\sqrt{\pi\sigma_1\sigma_2}}{\sqrt{2\pi(\sigma_1^2+\sigma_2^2)}} \exp\left(-\frac{(\mu_1-\mu_2)^2}{2(\sigma_1^2+\sigma_2^2)}\right)$$
$$= \sqrt{\frac{2\sigma_1\sigma_2}{\sigma_1^2+\sigma_2^2}} \exp\left(-\frac{(\mu_1-\mu_2)^2}{2(\sigma_1^2+\sigma_2^2)}\right)$$

Finally, taking the negative logarithm gives the divergence:

$$D_{CS}(p\|q) = -\log\left(\sqrt{\frac{2\sigma_1\sigma_2}{\sigma_1^2+\sigma_2^2}}\right) - \left(-\frac{(\mu_1-\mu_2)^2}{2(\sigma_1^2+\sigma_2^2)}\right)$$
$$= \frac{1}{2}\log\left(\frac{\sigma_1^2+\sigma_2^2}{2\sigma_1\sigma_2}\right) + \frac{(\mu_1-\mu_2)^2}{2(\sigma_1^2+\sigma_2^2)}$$

Now we wish to show that $D_{CS}(p\|q) \leq \min\{D_{KL}(p\|q), D_{KL}(q\|p)\}$. First, observe the derived closed-form expression for $D_{CS}(p\|q)$:

$$D_{CS}(p\|q) = \frac{1}{2}\log\left(\frac{\sigma_1^2+\sigma_2^2}{2\sigma_1\sigma_2}\right) + \frac{(\mu_1-\mu_2)^2}{2(\sigma_1^2+\sigma_2^2)}$$

Notice that this expression is invariant under the exchange of parameters (μ_1, σ_1) and (μ_2, σ_2). The term $(\mu_1-\mu_2)^2$ is symmetric, and the ratio $\frac{\sigma_1^2+\sigma_2^2}{2\sigma_1\sigma_2}$ is symmetric. Therefore:

$$D_{CS}(p\|q) = D_{CS}(q\|p)$$

To prove the main inequality $D_{CS}(p\|q) \leq \min\{D_{KL}(p\|q), D_{KL}(q\|p)\}$, it suffices to show that $D_{CS}(p\|q) \leq D_{KL}(p\|q)$. If this holds, then by symmetry, $D_{CS}(q\|p) \leq D_{KL}(q\|p)$ also holds, implying D_{CS} is less than or equal to both.

The KL divergence is given by:

$$D_{KL}(p\|q) = \log\frac{\sigma_2}{\sigma_1} + \frac{\sigma_1^2+(\mu_1-\mu_2)^2}{2\sigma_2^2} - \frac{1}{2}$$

Let $\Delta = D_{KL} - D_{CS}$. We can split this difference into a mean component Δ_μ and a variance component Δ_σ.

$$\Delta = \underbrace{\left(\frac{(\mu_1-\mu_2)^2}{2\sigma_2^2} - \frac{(\mu_1-\mu_2)^2}{2(\sigma_1^2+\sigma_2^2)}\right)}_{\Delta_\mu} + \underbrace{\left(\log\frac{\sigma_2}{\sigma_1} + \frac{\sigma_1^2}{2\sigma_2^2} - \frac{1}{2} - \frac{1}{2}\log\frac{\sigma_1^2+\sigma_2^2}{2\sigma_1\sigma_2}\right)}_{\Delta_\sigma}$$

Note that we can rewrite Δ_μ as

$$\Delta_\mu = \frac{(\mu_1-\mu_2)^2}{2}\left(\frac{1}{\sigma_2^2} - \frac{1}{\sigma_1^2+\sigma_2^2}\right)$$

Since variances are positive, $\sigma_1^2+\sigma_2^2 > \sigma_2^2$, which implies $\frac{1}{\sigma_2^2} > \frac{1}{\sigma_1^2+\sigma_2^2}$. Thus, for any μ_1, μ_2; $\Delta_\mu \geq 0$. The remaining terms are:

$$\Delta_\sigma = \left(\log\frac{\sigma_2}{\sigma_1} + \frac{\sigma_1^2}{2\sigma_2^2} - \frac{1}{2}\right) - \left(\frac{1}{2}\log\frac{\sigma_1^2+\sigma_2^2}{2\sigma_1\sigma_2}\right)$$

Let $t = \frac{\sigma_1}{\sigma_2}$. Then $\frac{\sigma_2}{\sigma_1} = \frac{1}{t}$. We rewrite the expression in terms of t:

$$\begin{aligned}\Delta_\sigma(t) &= -\log t + \frac{t^2}{2} - \frac{1}{2} - \frac{1}{2}\log\left(\frac{t^2\sigma_2^2 + \sigma_2^2}{2t\sigma_2^2}\right) \\ &= \frac{t^2-1}{2} - \log t - \frac{1}{2}\log\left(\frac{t^2+1}{2t}\right) \\ &= \frac{t^2-1}{2} - \log t - \frac{1}{2}\log(t^2+1) + \frac{1}{2}\log 2 + \frac{1}{2}\log t \\ &= \frac{t^2-1}{2} - \frac{1}{2}\log t - \frac{1}{2}\log(t^2+1) + \frac{1}{2}\log 2\end{aligned}$$

To find the minimum, we take the derivative with respect to t:

$$\frac{d\Delta_\sigma}{dt} = t - \frac{1}{2t} - \frac{1}{2}\cdot\frac{2t}{t^2+1} = t - \frac{1}{2t} - \frac{t}{t^2+1}$$

Combine terms over a common denominator $2t(t^2+1)$:

$$\frac{d\Delta_\sigma}{dt} = \frac{2t^2(t^2+1) - (t^2+1) - 2t^2}{2t(t^2+1)} = \frac{2t^4 + 2t^2 - t^2 - 1 - 2t^2}{2t(t^2+1)} = \frac{2t^4 - t^2 - 1}{2t(t^2+1)}$$

Setting the derivative to zero: $2t^4 - t^2 - 1 = 0$. Let $u = t^2$ $(u > 0)$.

$$2u^2 - u - 1 = (2u+1)(u-1) = 0$$

The roots are $u = 1$ and $u = -0.5$. Since t is a ratio of variances $t^2, t > 0$, so the only solution is $u = 1 \implies t = 1$. Checking the value at $t = 1$:

$$\Delta_\sigma(1) = \frac{1-1}{2} - 0 - \frac{1}{2}\log(2) + \frac{1}{2}\log 2 = 0$$

Since the derivative changes sign from negative $(t < 1)$ to positive $(t > 1)$, $t = 1$ is a global minimum. Therefore $\Delta_\sigma \geq 0$. Since both $\Delta_\mu \geq 0$ and $\Delta_\sigma \geq 0$, the total difference is non-negative:

$$D_{KL}(p\|q) - D_{CS}(p\|q) \geq 0 \implies D_{CS}(p\|q) \leq D_{KL}(p\|q)$$

which gives the desired inequality.

2. We are given the modified objective:

$$\mathcal{L}_{CS}(x;\theta,\phi) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - \lambda D_{CS}(q_\phi(z|x)\|p_\theta(z))$$

To understand this objective in relation to the standard VAE optimization, we utilize the fundamental identity of the Variational Autoencoder. We know that the log-marginal likelihood (evidence) can be decomposed as:

$$\log p_\theta(x) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x)\|p_\theta(z)) + D_{KL}(q_\phi(z|x)\|p_\theta(z|x))$$

From this, we can isolate the reconstruction term (expected log-likelihood):

$$\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] = \log p_\theta(x) + D_{KL}(q_\phi(z|x)\|p_\theta(z)) - D_{KL}(q_\phi(z|x)\|p_\theta(z|x))$$

Substituting this expression into the definition of $\mathcal{L}_{CS}$:

$$\begin{aligned}\mathcal{L}_{CS} &= [\log p_\theta(x) + D_{KL}(q_\phi(z|x)\|p_\theta(z)) - D_{KL}(q_\phi(z|x)\|p_\theta(z|x))] - \lambda D_{CS}(q_\phi(z|x)\|p_\theta(z)) \\ &= \log p_\theta(x) - \underbrace{D_{KL}(q_\phi(z|x)\|p_\theta(z|x))}_{\text{Variational Gap}} + \underbrace{D_{KL}(q_\phi(z|x)\|p_\theta(z)) - \lambda D_{CS}(q_\phi(z|x)\|p_\theta(z))}_{\text{Regularization Difference}}\end{aligned}$$

Interpretation of Terms:

- $\log p_\theta(x)$: The actual evidence we wish to maximize.
- $D_{KL}(q_\phi(z|x)\|p_\theta(z|x))$: The approximation error of the true posterior by the variational posterior. Minimizing the loss minimizes this gap.
- $D_{KL}(q\|p) - \lambda D_{CS}(q\|p)$: This term represents a modified regularization constraint. In a standard VAE, we simply subtract $D_{KL}(q\|p)$. Here, we are adding back the difference between the KL and the scaled CS divergence.

Impact of λ: The coefficient λ controls the strength of the prior matching constraint using the Cauchy-Schwarz metric.

- Since $D_{CS}(q\|p) \leq D_{KL}(q\|p)$, the gradient signals provided by D_{CS} are generally bounded and smoother than D_{KL}.
- Increasing λ emphasizes the Cauchy-Schwarz metric as the primary regularizer. D_{CS} is derived from the Renyi divergence of order 2, which penalizes the overlap between distributions differently than KL (which is based on logarithmic ratios).
- Qualitatively, the KL divergence forces the posterior q to cover the support of p (mode-covering behavior) but punishes q heavily if it has mass where p is zero. D_{CS} is more tolerant. A higher λ combined with the use of D_{CS} allows the latent representation to be less strictly compacted into the prior, potentially preventing the posterior collapse problem and allowing for more expressive latent structures, as the penalty for deviating from the prior is less aggressive than the standard KL term.

Problem 3: Posterior Collapse in Variational Autoencoders

1. Posterior collapse (also known as KL vanishing) is a phenomenon in Variational Autoencoders where the model learns to ignore the latent variable z. Intuitively, this occurs when the decoder $p_\theta(x|z)$ is powerful enough to model the data distribution $p_{\text{data}}(x)$ (or a sufficiently good approximation of it) without relying on the information provided by the latent code z. Consequently, the encoder $q_\phi(z|x)$ fails to encode any meaningful information about the input x and instead falls back to the prior distribution $p(z)$ to minimize the KL divergence regularization term.

   Mathematically, posterior collapse is characterized by the variational posterior becoming independent of the input x and equal to the prior. The formal expression characterizing this state is:

   $$\forall x \sim p_{\text{data}}(x), \quad q_\phi(z|x) \approx p_\theta(z)$$

   Alternatively, this can be expressed using the Kullback-Leibler divergence or Mutual In-

formation:

$$\mathbb{E}_{p_{\text{data}}(x)}[D_{\text{KL}}(q_\phi(z|x)\,||\,p_\theta(z))] \approx 0 \quad \text{or} \quad I_q(x;z) \approx 0$$

where $I_q(x;z)$ denotes the mutual information between the observation and the latent variable under the joint distribution induced by the encoder.

Source: Bowman, S. R., et al. (2016). "Generating Sentences from a Continuous Space". Proceedings of the 20th CoNLL.

2. We wish to find the optimal encoder $q_\phi(z|x)$ that maximizes the ELBO under the assumption that the decoder is powerful and independent of z.

   The Evidence Lower Bound (ELBO) for a single data point x is given by:

   $$\text{ELBO}(x;\theta,\phi) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - D_{\text{KL}}(q_\phi(z|x)\,||\,p(z))$$

   We are given the condition that the decoder is capable of modeling the data perfectly without z: $\exists \tilde{p}(x)$ such that $p_\theta(x|z) = \tilde{p}(x) \approx p_{\text{data}}(x)$. Substituting this into the reconstruction term:

   $$\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] = \mathbb{E}_{q_\phi(z|x)}[\log \tilde{p}(x)]$$

   Since $\log \tilde{p}(x)$ does not depend on z, it can be pulled out of the expectation. The expectation of a constant is the constant itself (since $\int q_\phi(z|x)dz = 1$):

   $$\mathbb{E}_{q_\phi(z|x)}[\log \tilde{p}(x)] = \log \tilde{p}(x)$$

   Now, the ELBO becomes:

   $$\text{ELBO}(x;\theta,\phi) = \log \tilde{p}(x) - D_{\text{KL}}(q_\phi(z|x)\,||\,p(z))$$

   To find the optimal encoder q_ϕ, we maximize this expression with respect to ϕ. The term $\log \tilde{p}(x)$ depends only on the decoder and the data, so it is constant with respect to the encoder parameters ϕ. Therefore, maximizing the ELBO is equivalent to minimizing the KL divergence term:

   $$\max_\phi \text{ELBO} \iff \min_\phi D_{\text{KL}}(q_\phi(z|x)\,||\,p(z))$$

   By Gibbs' inequality, the KL divergence is non-negative and is minimized (equal to zero) if and only if the two distributions are identical almost everywhere. Thus, the optimal encoder satisfies:

   $$q_\phi(z|x) = p(z)$$

   This confirms that if the decoder ignores z, the optimal strategy for the encoder is to effectively ignore x and output the prior, leading to posterior collapse.

3. We start with the standard definition of the ELBO averaged over the dataset:

   $$\mathcal{L} = \mathbb{E}_{p_{\text{data}}(x)}\left[\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - D_{\text{KL}}(q_\phi(z|x)\,||\,p(z))\right]$$

   We focus on decomposing the expectation of the KL term: $\mathbb{E}_{p_{\text{data}}(x)}[D_{\text{KL}}(q_\phi(z|x)\,||\,p(z))]$. Let $q(x,z) = p_{\text{data}}(x)q_\phi(z|x)$ be the joint distribution, and $q_\phi(z) = \mathbb{E}_{p_{\text{data}}(x)}[q_\phi(z|x)]$ be

the aggregated posterior (marginal inference distribution).

$$\begin{aligned}
\mathbb{E}_{p_{\text{data}}(x)}[D_{\text{KL}}(q_\phi(z|x)\,||\,p(z))] &= \int p_{\text{data}}(x) \int q_\phi(z|x) \log \frac{q_\phi(z|x)}{p(z)}\, dz\, dx \\
&= \iint q(x,z) \log \left(\frac{q_\phi(z|x)}{p(z)} \cdot \frac{q_\phi(z)}{q_\phi(z)} \right) dz\, dx \\
&= \iint q(x,z) \left(\log \frac{q_\phi(z|x)}{q_\phi(z)} + \log \frac{q_\phi(z)}{p(z)} \right) dz\, dx \\
&= \iint q(x,z) \log \frac{q_\phi(z|x)}{q_\phi(z)}\, dz\, dx + \iint q(x,z) \log \frac{q_\phi(z)}{p(z)}\, dz\, dx
\end{aligned}$$

The first term is exactly the definition of the Mutual Information $I_q(x;z)$:

$$\iint q(x,z) \log \frac{q(x,z)}{p_{\text{data}}(x) q_\phi(z)}\, dz\, dx = I_q(x;z)$$

For the second term, we marginalize out x:

$$\int \left(\int p_{\text{data}}(x) q_\phi(z|x)\, dx \right) \log \frac{q_\phi(z)}{p(z)}\, dz = \int q_\phi(z) \log \frac{q_\phi(z)}{p(z)}\, dz = D_{\text{KL}}(q_\phi(z)\,||\,p(z))$$

Substituting this back into the ELBO expression:

$$\text{ELBO}(\theta,\phi) = \mathbb{E}_{p_{\text{data}}(x)} \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - D_{\text{KL}}(q_\phi(z)\,||\,p(z)) - I_q(x;z)$$

In this decomposition, posterior collapse corresponds to the case where the Mutual Information term vanishes:

$$I_q(x;z) \approx 0$$

This implies that x and z are independent under the joint distribution learned by the model. Furthermore, the term $D_{\text{KL}}(q_\phi(z)\,||\,p(z))$ measures how much the aggregated posterior deviates from the prior. In a total collapse scenario, this term also tends to 0.

4. Three practical factors that can lead to posterior collapse are:

   - **Powerful Autoregressive Decoders:** If the decoder is a strong autoregressive model (e.g., PixelCNN for images or LSTM for text), it can model the local dependencies in x very well based solely on previous ground-truth tokens $x_{<t}$. The decoder effectively ignores the global context provided by z because the reconstruction signal from local correlations is stronger and easier to learn than the signal from the noisy latent variable z.
   - **Aggressive KL Regularization:** The optimization objective involves a trade-off between reconstruction and the KL divergence. At the start of training, the approximate posterior $q_\phi(z|x)$ often carries little information and is far from the prior. The KL term exerts a strong gradient penalty pushing q_ϕ toward $p(z)$. If this signal is too strong relative to the reconstruction signal (which is initially weak), the optimizer may set $q_\phi(z|x) = p(z)$ (a local minimum) to zero out the KL cost immediately, causing the model to get stuck in a collapsed state.

- **Lagging Inference Networks:** During training, the encoder (inference network) and decoder (generative network) are optimized simultaneously. Often, the encoder requires more iterations or is harder to optimize than the decoder. If the encoder lags behind, the z it produces is uninformative (effectively noise). The decoder learns to ignore this noise to minimize reconstruction error. Once the decoder learns to ignore z, the gradients propagating back to the encoder through the latent variable vanish, preventing the encoder from ever learning meaningful representations.

5. Three strategies for preventing posterior collapse are:

   - **KL Annealing:** This technique involves re-weighting the KL term in the objective function with a variable coefficient β that starts at 0 and gradually increases to 1 over the course of training.

     $$\mathcal{L} \approx \text{Reconstruction} - \beta \cdot D_{\text{KL}}$$

     By setting $\beta \approx 0$ initially, the model acts like a standard autoencoder, allowing the encoder to learn to encode information into z solely to minimize reconstruction error without the penalty of the prior. Once z contains useful information, β is slowly introduced, forcing the distribution toward the prior while retaining the information already learned.
   - **Free Bits / KL Thresholding:** This method modifies the objective function to ensure the KL term is only penalized if it exceeds a certain threshold λ (target bits).

     $$\mathcal{L} = \mathbb{E}[\log p(x|z)] - \max(\lambda, D_{\text{KL}}(q(z|x)||p(z)))$$

     This effectively "reserves" λ units of information capacity (free bits) for the latent variable. If the KL divergence is very low (collapse), the gradient with respect to the KL term becomes zero (constant cost λ), removing the pressure to reduce the information content further. This allows the encoder to utilize at least λ nats/bits of information without penalty.
   - **Weakening the Decoder:** This involves limiting the capacity or the receptive field of the decoder (e.g., using Word Dropout or limiting the context window in autoregressive models). If the decoder $p_\theta(x|z)$ is made less powerful (e.g., it cannot predict x_t perfectly from $x_{<t}$), the term $\log p_\theta(x|z)$ will be high (poor reconstruction) unless it utilizes the extra information provided by z. This increases the gradient magnitude for the reconstruction term with respect to z, making the mutual information $I_q(x; z)$ valuable enough to outweigh the KL penalty.

Problem 4: Probabilistic Graph Forecasting with Autoregressive Decoders

1. We aim to maximize the log marginal likelihood $\log p_\theta(X_{1:T}, A_{1:T})$. The joint distribution is factorized as:

   $$p_\theta(X_{1:T}, A_{1:T}, Z_{0:T}) = p(Z_0) \prod_{t=0}^{T-1} p_\theta(X_{t+1}, A_{t+1}|Z_t) p_\theta(Z_{t+1}|Z_t)$$

   We assume an initial latent state Z_0 is given (fixed) or handled by a separate prior that

is not part of the optimization loop. The approximate posterior for the sequence $Z_{1:T}$ factorizes as $q_\phi(Z_{1:T}|X_{1:T}, A_{1:T}) = \prod_{t=0}^{T-1} q_\phi(Z_{t+1}|Z_t, X_{1:t+1}, A_{1:t+1})$ (assuming a recursive structure) or simply independent terms given history.

Using Jensen's inequality and the approximate posterior $q_\phi(Z_{1:T}|X_{1:T}, A_{1:T})$, we write the lower bound:

$$\log p_\theta(X_{1:T}, A_{1:T}) \geq \mathbb{E}_{q_\phi(Z_{1:T}|X_{1:T},A_{1:T})}\left[\log \frac{p(Z_0)\prod_{t=0}^{T-1} p_\theta(X_{t+1}, A_{t+1}|Z_t)p_\theta(Z_{t+1}|Z_t)}{\prod_{t=0}^{T-1} q_\phi(Z_{t+1}|Z_t, X_{1:t+1}, A_{1:t+1})}\right]$$

Assuming Z_0 is fixed (constant), the terms involving $p(Z_0)$ vanish from the optimization (or evaluate to 0 in log-probability). We can rearrange the product into a sum over time steps $t = 0$ to $T-1$.

The ELBO is derived as:

$$\mathcal{L} = \sum_{t=0}^{T-1} \mathbb{E}_{q_\phi(Z_t|X_{1:t},A_{1:t})}\left[\underbrace{\log p_\theta(X_{t+1}, A_{t+1}|Z_t)}_{\text{Reconstruction Term}} - \underbrace{D_{\mathrm{KL}}(q_\phi(Z_{t+1}|Z_t, X_{1:t+1}, A_{1:t+1}) \,||\, p_\theta(Z_{t+1}|Z_t))}_{\text{Temporal KL Divergence}}\right]$$

Note: For the base case $t = 0$, the expectation is trivial since Z_0 is fixed. The reconstruction term becomes $\log p_\theta(X_1, A_1|Z_0)$ and the KL term becomes $D_{\mathrm{KL}}(q_\phi(Z_1|Z_0, X_{1:1}, A_{1:1}) \,||\, p_\theta(Z_1|Z_0))$.

Interpretation of Indices:

- At $t = 0$: We reconstruct (X_1, A_1) using the initial state Z_0. We also compute the KL divergence for the first inferred latent Z_1 against the prior transition $p(Z_1|Z_0)$.
- At $t = T-1$: We reconstruct (X_T, A_T) using Z_{T-1} and infer Z_T.

2. Given the conditional likelihood $p_\theta(X_{t+1}, A_{t+1}|Z_t)$, we sum the log-probabilities for node features and edges.

   1. Node Feature Term (Gaussian):

$$\log p_\theta(X_{t+1}|Z_t) = \sum_{v=1}^{n}\left(-\frac{1}{2}(x_v^{t+1} - \mu_v^\theta(Z_t))^\top(\Sigma_v^\theta(Z_t))^{-1}(x_v^{t+1} - \mu_v^\theta(Z_t)) - \frac{1}{2}\log|2\pi\Sigma_v^\theta(Z_t)|\right)$$

   2. Adjacency Matrix Term (Bernoulli):

$$\log p_\theta(A_{t+1}|Z_t) = \sum_{u<v}\left[A_{uv}^{t+1}\log(\pi_{uv}^\theta(Z_t)) + (1 - A_{uv}^{t+1})\log(1 - \pi_{uv}^\theta(Z_t))\right]$$

   The total reconstruction term at step t is the sum of these two components.

3. If the decoder is deterministic:

   - **Effect on ELBO:** The variance $\Sigma \to 0$ causes the log-likelihood density to approach infinity for perfect reconstructions and negative infinity otherwise. This makes the ELBO ill-defined and effectively transforms the objective into a squared error minimization (ignoring the normalization constant), losing the probabilistic interpretation.

- **Multimodality:** A deterministic mapping $Z_t \mapsto (X_{t+1}, A_{t+1})$ implies a one-to-one correspondence. It cannot model multimodal futures (e.g., if the graph could evolve in two equally likely distinct ways given Z_t, a deterministic decoder will likely output the "blur" or average of these outcomes).
- **Stability:** Removing the variance component removes the model's ability to handle uncertainty in the data, often leading to mode collapse or unstable gradients where the model tries to fit noise exactly.

4. **Generation Process:** Given an observed history $(X_{1:T}, A_{1:T})$ and assuming we have inferred the final latent state Z_T (or the sequence ending in Z_T): To generate future graphs for H steps (starting from time T to generate $X_{T+1:T+H}$):

   Initialize with current latent $\hat{Z}_T = Z_T$. For $k = 0$ to $H - 1$:

   1. **Generate Graph:** Sample $(\hat{X}_{T+k+1}, \hat{A}_{T+k+1}) \sim p_\theta(X, A|\hat{Z}_{T+k})$. (Note the index lag: Z_t generates X_{t+1}).
   2. **Update Latent:** Sample next latent state $\hat{Z}_{T+k+1} \sim p_\theta(Z|\hat{Z}_{T+k})$.

   Convergence Conditions: The sequence of generated graphs converges to a stationary distribution if the latent transition kernel $p_\theta(Z_{t+1}|Z_t)$ forms an **ergodic Markov chain**. Specifically, if the chain is irreducible (possible to reach any region of the state space with non-zero probability) and aperiodic, the distribution of Z_t converges to a unique stationary distribution $\pi(Z)$ as $t \to \infty$. Mathematically, $\pi(Z)$ satisfies the integral equation:

$$\pi(Z') = \int p_\theta(Z'|Z)\pi(Z)\, dZ$$

   Consequently, the generated graphs will follow the marginal distribution induced by passing this stationary latent distribution through the decoder:

$$p_\infty(X, A) = \int p_\theta(X, A|Z)\pi(Z)\, dZ$$

5. **Amortization Gap:** The amortization gap is the divergence between the true posterior $p_\theta(Z_t|X_{1:t})$ (which is computationally intractable) and the approximate posterior $q_\phi(Z_t|X_{1:t})$ parameterized by the GNN. Mathematically, this gap is non-zero:

$$\Delta_{\text{gap}} = D_{\text{KL}}(q_\phi(Z_t|X_{1:t}) \,||\, p_\theta(Z_t|X_{1:t})) > 0$$

   This implies that even with optimal training, the samples $Z_t \sim q_\phi$ used to train the transition prior are not perfectly distributed according to the true posterior.

   Distribution Drift (Covariate Shift):

   - **Training Phase (Teacher Forcing):** The transition prior $p_\theta(Z_{t+1}|Z_t)$ is trained to maximize probability conditioned on correct latents $Z_t \sim q_\phi(\cdot|X_{1:t})$.
   - **Generation Phase (Autoregressive):** The model conditions on its own predicted samples $\hat{Z}_t \sim p_\theta(\cdot|\hat{Z}_{t-1})$.

If there is a small error ϵ at step t (due to the amortization gap or sampling noise), such that $\|\hat{Z}_t - Z_t\| \approx \epsilon$, this error propagates through the transition kernel. If the transition dynamics are Lipschitz continuous with constant $L > 1$ (typical in unstable dynamics), the error at the next step scales as:

$$\|\hat{Z}_{t+1} - Z_{t+1}\| \leq L\|\hat{Z}_t - Z_t\| + \text{noise} \approx L\epsilon$$

Over k steps, the error grows as $L^k\epsilon$. This exponential divergence causes the latent trajectory to drift off the manifold of valid states that the decoder $p_\theta(X|Z)$ was trained on, leading to degraded graph generation.

Scheduled Sampling: Scheduled Sampling bridges the gap between training and inference distributions. We introduce a coin flip variable $c_t \sim \text{Bernoulli}(\epsilon_i)$ at training step i. The input $\tilde{Z}_t$ passed to the transition model to predict Z_{t+1} is selected as:

$$\tilde{Z}_t = \begin{cases} Z_t \sim q_\phi(Z_t|X_{1:t}) & \text{with probability } \epsilon_i \text{ (Teacher Forcing)} \\ \hat{Z}_t \sim p_\theta(Z_t|\tilde{Z}_{t-1}) & \text{with probability } 1 - \epsilon_i \text{ (Autoregressive)} \end{cases}$$

The probability ϵ_i starts at 1 (pure teacher forcing) and decays to 0 (pure autoregressive) during training. This forces the model to optimize:

$$\max_\theta \mathbb{E}_{\tilde{Z}_t}[\log p_\theta(Z_{t+1}|\tilde{Z}_t)]$$

By training on its own noisy samples $\hat{Z}_t$, the model learns to map slightly perturbed latent states back towards the correct trajectory, effectively learning a corrective vector field that reduces drift.