Single-Cell Variational Auto Encoders

Paul Caucheteux, Marc Chevrière, Marion Hubler
Master MVA, ENS Paris Saclay

December 2024

1 Introduction

1.1 Overview and biological background

Single cell RNA sequencing (scRNA-seq) is a technology that allows the measurement of gene expressions at individual cell level. Many biological analyses can be performed on such datasets, namely: imputation, visualization and clustering in a smaller representation space, as well as differential expression analysis. However, before performing biological analyses on these datasets, it is essential to develop methods that address the main challenges associated with them, namely, correcting for technical variability (batch effect and difference in cell depth), managing the computational demands of these usually very large datasets and properly handling their sparsity.

1.2 Presentation of the research papers and the dataset

Article [2] introduced the idea of addressing these challenges, using Variational Autoencoders (VAEs). They argue that VAEs are well-suited for the previously mentioned challenges, as they can be efficiently trained on large datasets and are more likely to ensure the consistency of downstream tasks, by using a single probabilistic model for all of them. Article [2] focuses on the performances of a more refined scVAE framework: SCVI. Article [1] introduced a new scVAE architecture, with a Gaussian Mixture prior on the latent space, to try and improve the model on the specific task of clustering in the latent space.

1.3 Objectives

- Check if the GMVAE truly achieves better clustering performances than the simple VAE.
- Check if the new GMVAE architecture of [1] truly refines the model's understanding of the data, and achieves better performance on other tasks.
- Compare how different reconstruction likelihoods perform on these tasks.

1.4 Notations

The scRNA-seq data is represented by a count matrix $\mathbf{X} \in \mathbb{N}^{N \times G}$, where each vector $\mathbf{x}_i \in \mathbb{N}^G$ corresponds to the counts of G genes for a given cell i. In our case, we use the Cortex dataset available in the scvi-tools library, developed by Lopez et al. [2]. The dataset consists of N = 3,005 cells and G = 19,972 genes. We chose not to apply any preprocessing, such as selecting the most variable genes, and instead work directly with the raw dataset.

2 Methods

2.1 Variational auto-encoder models

2.1.1 Simple VAE

Variational Autoencoders are a well-established class of generative models that learn a latent representation of high-dimensional data. Given a count vector $\mathbf{x} \in \mathbb{N}^G$ representing gene expression for G genes, VAEs model $\mathbf{x}$ as sampled from $p_\theta(\mathbf{x}|\mathbf{z})$, where $\mathbf{z} \in \mathbb{R}^d$

is a latent variable drawn from a prior $p(\mathbf{z})$. The Evidence Lower Bound (ELBO) is maximized to train the model Appendix C.1:

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

where $q_\phi(\mathbf{z}|\mathbf{x})$ is the encoder approximating the posterior, and $p_\theta(\mathbf{x}|\mathbf{z})$ is the decoder reconstructing $\mathbf{x}$.

$$\begin{aligned} \mathbf{z} &\sim \mathcal{N}(0, I) \\ q_\phi(\mathbf{z}|\mathbf{x}) &\sim \mathcal{N}\left(\boldsymbol{\mu}_\phi(\mathbf{x}), \boldsymbol{\sigma}^2_\phi(\mathbf{x})\right) \\ p_\theta(\mathbf{x}|\mathbf{z}) &\sim \text{Likelihood}\left(f_\theta(\mathbf{z})\right) \end{aligned}$$

where:

$$\boldsymbol{\mu}_\phi(\mathbf{x}) = \text{MLP}^{\mu}_\phi(\mathbf{x}), \quad \boldsymbol{\sigma}^2_\phi(\mathbf{x}) = \text{diag}\left(\text{MLP}^{\sigma^2}_\phi(\mathbf{x})\right), \quad f_\theta(\mathbf{z}) = \text{MLP}^{f}_\theta(\mathbf{z}),$$

with $\boldsymbol{\mu}_\phi(\mathbf{x})$ being a vector of means and $\boldsymbol{\sigma}^2_\phi(\mathbf{x})$ representing a diagonal covariance matrix with different variances along the diagonal. Here, MLP_θ represents multi-layer perceptrons incorporating ReLU activation, batch normalization, and dropout at each layer.

2.1.2 Gaussian mixture VAE

The Gaussian-Mixture Variational Autoencoder (GMVAE) [1] extends the VAE by introducing a discrete latent variable y to model cluster memberships, directly integrating clustering into the latent space. This approach, which reflects the assumption that cells belong to specific types, influencing their gene expression, better captures the data's structure compared to post-hoc clustering methods like K-means.

$$\begin{aligned} &\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}_\theta(y), \sigma^2_\theta(y)\mathbf{I}), \quad q_\phi(\mathbf{z}|\mathbf{x}, y) \sim \mathcal{N}\left(\boldsymbol{\mu}_\phi(\mathbf{x}, y), \boldsymbol{\sigma}^2_\phi(\mathbf{x}, y)\right), \\ &q_\phi(y|\mathbf{x}) \sim \text{Categorical}\left(\boldsymbol{\pi}_\phi(\mathbf{x})\right), \quad p_\theta(\mathbf{x}|\mathbf{z}) \sim \text{Likelihood}\left(f_\theta(\mathbf{z})\right) \end{aligned}$$

where:

$$\boldsymbol{\mu}_\phi(\mathbf{x}, y) = \text{MLP}^{\mu}_\phi(\mathbf{x}, y), \quad \boldsymbol{\sigma}^2_\phi(\mathbf{x}, y) = \text{diag}\left(\text{MLP}^{\sigma^2}_\phi(\mathbf{x}, y)\right), \boldsymbol{\pi}_\phi(\mathbf{x}) = \text{MLP}^{\pi}_\phi(\boldsymbol{x}),$$

$$\boldsymbol{\mu}_\theta(y) = \text{MLP}^{\mu}_\theta(y), \quad \boldsymbol{\sigma^2}_\theta(y) = \text{diag}\left(\text{MLP}^{\sigma^2}_\theta(y)\right), \quad f_\theta(\mathbf{z}) = \text{MLP}^{f}_\theta(\mathbf{z}),$$

with $\boldsymbol{\mu}_\phi(\mathbf{x}, y)$ and $\boldsymbol{\sigma}^2_\phi(\mathbf{x}, y)$ as the posterior parameters learned by the encoder, $\boldsymbol{\mu}_\theta(y)$ and $\boldsymbol{\sigma}^2_\theta(y)$ as the prior parameters for cluster y, $p_\theta(y) \sim \text{Uniform}(1/K)$ as the uniform prior over K clusters, $\boldsymbol{\pi}_\phi(\mathbf{x})$ as the cluster responsibilities, and MLP_ϕ, MLP_θ as multi-layer perceptrons with ReLU activation functions, normalization, and dropout.

Generative Process The GMVAE models the joint probability distribution of the observed data $\mathbf{x}$, the latent variable $\mathbf{z}$, and the cluster variable y (Figure 1b):

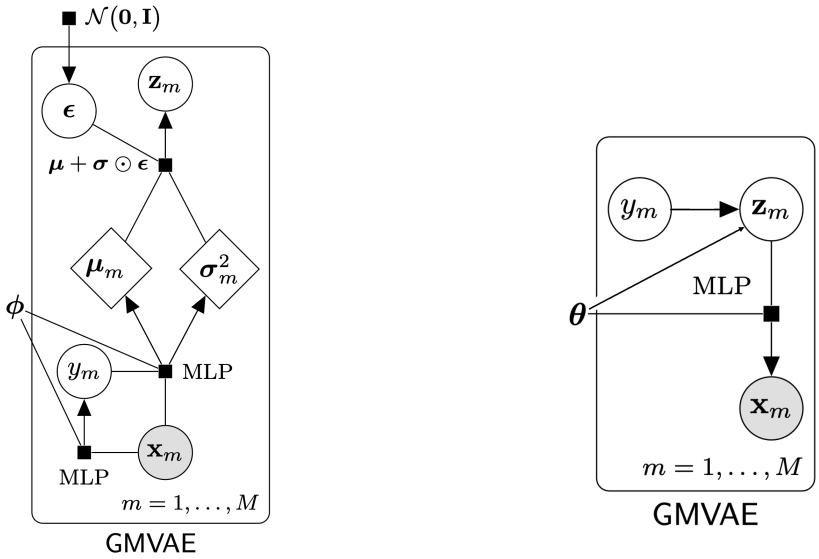
$$p_\theta(\mathbf{x}, y, \mathbf{z}) = p_\theta(\mathbf{x}|\mathbf{z}) p_\theta(\mathbf{z}|y) p_\theta(y),$$

where $p_\theta(\mathbf{x}|\mathbf{z})$ is the likelihood function modeling the observed data, $p_\theta(\mathbf{z}|y)$ is a Gaussian prior conditioned on the cluster y, and $p_\theta(y)$ is a categorical uniform distribution over K clusters.

Inference Process The GMVAE employs an approximate posterior distribution $q_\phi(\mathbf{z}, y|\mathbf{x})$, factorized as (Figure 1a):

$$q_\phi(\mathbf{z}, y|\mathbf{x}) = q_\phi(\mathbf{z}|\mathbf{x}, y)q_\phi(y|\mathbf{x}),$$

where $q_\phi(\mathbf{z}|\mathbf{x}, y) \sim \mathcal{N}\left(\boldsymbol{\mu}_\phi(\mathbf{x}, y), \boldsymbol{\sigma}^2_\phi(\mathbf{x}, y)\right)$ and $q_\phi(y|\mathbf{x})$ is a categorical distribution with probabilities $\boldsymbol{\pi}_\phi(\mathbf{x})$.



(a) Encoder: Approximates $q_\phi(\mathbf{z}, y|\mathbf{x})$.

(b) Decoder: Models $p_\theta(\mathbf{x}|\mathbf{z}, y)$.

Figure 1: Graphical models for the encoder and decoder of the GMVAE.

Optimization Objective The Evidence Lower Bound for the GMVAE is given by Appendix C.2.:

$$\begin{aligned}\mathcal{L}(\theta, \phi; \mathbf{x}) = &\mathbb{E}_{q_\phi(y|\mathbf{x})}\Big[\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x},y)}\big[\log p_\theta(\mathbf{x}|\mathbf{z})\big] \\ &- \mathrm{KL}\big(q_\phi(\mathbf{z}|\mathbf{x}, y) \,\|\, p_\theta(\mathbf{z}|y)\big)\Big] \\ &- \mathrm{KL}\big(q_\phi(y|\mathbf{x}) \,\|\, p_\theta(y)\big).\end{aligned}$$

2.2 Choice of the Likelihood

In the VAE framework, the choice of likelihood is critical for modeling single-cell data, as shown in [1]. Single-cell observations are sparse count vectors with over-dispersion, requiring suitable discrete distributions.

The **Poisson** distribution, though simple, assumes its variance equals its mean, which is unrealistic for single-cell data. The **Negative Binomial** distribution addresses this limitation by capturing over-dispersion and extreme values effectively.

To model sparsity and both sparsity and over-dispersion, we use **Zero-Inflated Poisson (ZIP)** or **Zero-Inflated Negative Binomial (ZINB)** distributions. These include a binary process for excess zeros and a base likelihood for non-zero values, making them ideal for single-cell observations.

The final form of a Zero-Inflated distribution can be expressed as follows:

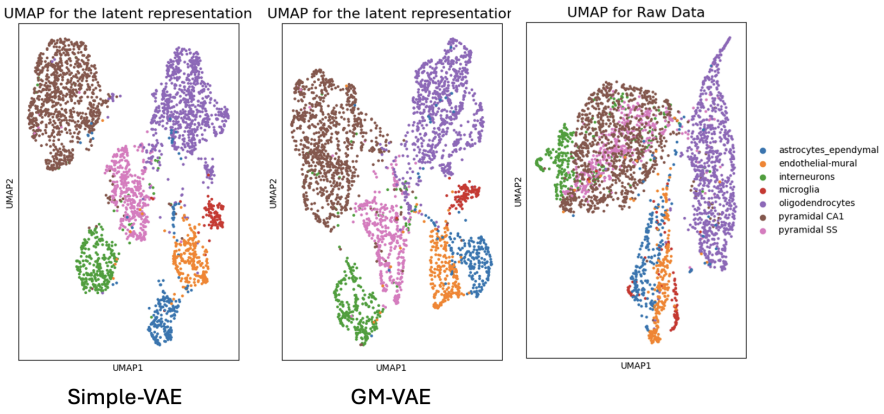


Figure 2: UMAP on the latent variables with each model compared to a UMAP after a PCA on raw data.

$$p_\theta(x_j) = \begin{cases} \rho_j + (1-\rho_j) f(x_j;\lambda) & \text{if } x_j = 0, \\ (1-\rho_j) f(x_j;\lambda) & \text{if } x_j > 0, \end{cases}$$

where x_j is the count of gene j in cell $\mathbf{x}$, $\rho_j \in [0,1]$ is the zero-inflation probability given by $\text{MLP}_\theta(\mathbf{z})$, and $f(x_j;\lambda)$ is the count distribution with parameter λ.

3 Experiments and Results

As shown in [2], VAEs applied to single-cell data enable tasks like batch effect removal and differential expression analysis. Here, we focus on clustering and imputation, two widely used and interpretable tasks.

3.1 Visualization

VAEs provide dimensionality reduction for large-scale single-cell data, enabling the identification of clusters and cell types through a latent representation. Although this latent space compresses high-dimensional data, it can still be too high-dimensional for direct visualization (dimension 10 in our case).

To address this, we use UMAP to project the latent space into 2D or 3D, facilitating the interpretation of its structure. As a baseline, we apply PCA to the raw input data to match the latent space dimension, followed by UMAP for visualization. This comparison helps assess the quality of the latent representations learned by the VAE models, as illustrated in Figure 2.

3.2 Clustering task

To quantitatively assess the clustering performance of our models, we use metrics such as Adjusted Rand Index (ARI), Normalized Mutual Information (NMI) and others detailed in Appendix D. These metrics require cluster assignments for each observation to compare to expert labels that come with the dataset. The GMVAE inherently provides cluster assignments by taking the `argmax` of the responsibilities $\boldsymbol{\pi}_\phi(\mathbf{x})$ to determine the categorical latent variable Y. For the Simple VAE, cluster assignments are obtained

by applying the K-means algorithm to the latent space representation. The results for both models, across different likelihood choices, are summarized in Table 1, assuming the number of clusters K is known.

Likelihoods	**Simple VAE**		**GMVAE**	
	ARI	**NMI**	**ARI**	**NMI**
Poisson	**0.6327**	**0.6972**	0.4026	0.5207
Zero-inflated Poisson	**0.6511**	**0.7054**	0.3684	0.4817
Negative Binomial	**0.5875**	**0.6338**	0.3787	0.4968
Zero-inflated Negative Binomial	**0.6803**	**0.7074**	0.3751	0.4959

Table 1: Comparison of models on the clustering task using different likelihoods.

3.3 Imputation task

Single-cell data often contain significant noise due to technical limitations, and Variational Autoencoders are effective for reconstructing corrupted or noisy data (imputation), helping to reduce uncertainty caused by dropouts and technical noise. To evaluate imputation performance, we follow the methodology described in [2]. Specifically, we corrupt 10% of the nonzero entries by multiplying each selected entry by a Bernoulli random variable Ber(0.9). During reconstruction, we directly take the expectation of the reconstructed values instead of sampling from the likelihood distribution. The distance between the original and imputed values is computed only on the corrupted entries.

The original paper [2] employs the median L1 distance, which is robust to outliers but may overlook large prediction errors. To explore alternatives, we evaluate with L2 distance as well, and the results are summarized in Table 2. Our implementation simplifies the architecture proposed in [2] by omitting batch annotations and considering all genes instead of selecting the most variable ones. While this simplification may lead to lower performance compared to the original, it allows for an effective comparison of our Simple VAE and GMVAE models across different likelihood choices.

Likelihoods	**Simple VAE**		**GMVAE**	
	Median	**Mean**	**Median**	**Mean**
Poisson	**0.8889**	**1.819**	0.9044	1.900
Zero-inflated Poisson	**0.9345**	**2.298**	0.9527	2.393
Negative Binomial	**2.768**	**4.212**	2.779	4.244
Zero-inflated Negative Binomial	**4.481**	**5.665**	5.283	6.649

Table 2: Comparison of models on the imputation task using different likelihoods, and different distance metrics.

4 Discussion

In this section, we will compare the performances of the simple VAE and GMVAE frameworks by interpreting how they perform on the clustering and imputation tasks, discussing the implications of different likelihood choices and questioning the need for a more refined framework such as the GMVAE.

4.1 Clustering Performance

The clustering task highlights significant differences between the simple VAE and GMVAE models, with the simple VAE consistently outperforming the GMVAE across all likelihoods in clustering metrics (Table 1). This discrepancy may stem from a misalignment between the Gaussian mixture prior of the GMVAE and the intrinsic structure of the data. While the GMVAE explicitly enforces cluster separations in the latent space, these clusters might not correspond to the biological annotations in the cortex dataset, suggesting that the GMVAE may be capturing alternative, potentially less biologically relevant patterns.

Additionally, the choice of likelihood remains crucial for clustering quality. The Zero-Inflated Negative Binomial distribution delivers superior performance for both models, reflecting its ability to account for the sparsity and over-dispersion inherent to scRNA-seq data. This further highlights the importance of choosing tailored likelihoods that align with the intrinsic properties of single-cell datasets. However, we do not observe ARI and NMI values as high as those reported in [1], which could be attributed to differences in the architecture of the MLPs or the initialization of the Gaussian prior parameters.

4.2 Imputation Performance and imputation evaluation metrics

Our results reveal contrasting performances for the Simple VAE and GMVAE models depending on the likelihood choice. The results (Table 2) show that models with lower dispersion (Poisson) achieve lower errors compared to those using higher dispersion (Negative Binomial), despite the latter's suitability for overdispersed single-cell RNA-seq data.

Interestingly, zero-inflated models perform worse than their non-zero-inflated counterparts on the imputation task. This raises questions about their ability to distinguish between *true zeros* (biological absence of expression) and *false zeros* (technical dropouts). Zero-inflated models, address this distinction by combining a Bernoulli component, with parameter ρ, for technical zeros, and a Negative Binomial component, with mean μ, for biological expression. In theory, the expectation $(1 - \rho) \cdot \mu$ imputes small but non-zero values for technical zeros while preserving true zeros.

However, our results suggest that artificially introduced dropouts may be misinterpreted as true zeros, leading to poorer performances. A comparison of normalized data with bulk RNA-seq differential expression would help clarify whether the model effectively handles technical zeros. Until such validation, the use of the *median* rather than the mean remains justified for evaluating imputation performance, as it is more robust to errors caused by outlier imputation.

References

[1] Christopher Heje Grønbech et al. "scVAE: variational auto-encoders for single-cell gene expression data". In: *Bioinformatics* 36.16 (May 2020), pp. 4415–4422. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btaa293. eprint: https://academic.oup.com/bioinformatics/article-pdf/36/16/4415/50677086/btaa293.pdf. URL: https://doi.org/10.1093/bioinformatics/btaa293.

[2] Romain Lopez et al. "Deep generative modeling for single-cell transcriptomics". eng. In: *Nature methods* 15.12 (2018), pp. 1053–1058. ISSN: 1548-7091.

A Link to the Github repository

https://github.com/marc-chevriere/PGM-single-cell

B Contributions

Category	Entries	Paul	Marc	Marion
Code	Models		✓	✓
	Evaluation Task	✓		✓
	Experience Launching	✓	✓	
Final Report	Introduction			✓
	Discussions	✓	✓	✓
	Methods		✓	✓
	Experiments	✓	✓	
	Poster	✓		✓

C Proofs

C.1 Classical ELBO

To begin, we write:

$$\log p_\theta(x) = \log \int p_\theta(x, z)\, dz.$$

Rewriting using an auxiliary distribution $g(z)$:

$$\log p_\theta(x) = \log \int g(z) \frac{p_\theta(x, z)}{g(z)}\, dz.$$

Applying Jensen's inequality:

$$\log p_\theta(x) \geq \mathbb{E}_{g(z)} \left[\log \frac{p_\theta(x, z)}{g(z)} \right].$$

Choosing $g(z) = q_\phi(z \mid x)$, this becomes the variational posterior and we recover the ELBO:

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

Expanding this further, we write:

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

Reorganizing:

$$\text{ELBO} = \mathbb{E}_{q_\phi(z|x)} \left[\log p_\theta(x) \right] + \mathbb{E}_{q_\phi(z|x)} \left[\log \frac{p_\theta(z \mid x)}{q_\phi(z \mid x)} \right].$$

The first term simplifies to $\log p_\theta(x)$, leaving:

$$\text{ELBO} = \log p_\theta(x) - \text{KL} \left(q_\phi(z \mid x) \,\|\, p_\theta(z \mid x) \right).$$

Thus, we have:

$$\log p_\theta(x) = \text{ELBO} + \text{KL} \left(q_\phi(z \mid x) \,\|\, p_\theta(z \mid x) \right).$$

Now, expanding $p_\theta(x, z)$ using $p_\theta(x, z) = p_\theta(x \mid z)p_\theta(z)$, we write:

$$\text{ELBO} = \mathbb{E}_{q_\phi(z|x)} \left[\log \frac{p_\theta(x \mid z)p_\theta(z)}{q_\phi(z \mid x)} \right].$$

Breaking this into separate terms:

$$\text{ELBO} = \mathbb{E}_{q_\phi(z|x)} \left[\log p_\theta(x \mid z) \right] - \text{KL} \left(q_\phi(z \mid x) \,\|\, p_\theta(z) \right).$$

C.2 ELBO for GMM

To begin, we write:

$$\log p_\theta(x) = \log \int p_\theta(x, y, z)\, dy\, dz.$$

Rewriting using an auxiliary distribution $g(y, z)$:

$$\log p_\theta(x) = \log \int g(y, z) \frac{p_\theta(x, y, z)}{g(y, z)}\, dy\, dz.$$

Applying Jensen's inequality:

$$\log p_\theta(x) \geq \mathbb{E}_{g(y,z)} \left[\log \frac{p_\theta(x, y, z)}{g(y, z)} \right].$$

Choosing $g(y, z) = q_\phi(z, y|x)$, this becomes the variational posterior and we recover the ELBO:

$$\log p_\theta(x) \geq \mathbb{E}_{q_\phi(z,y|x)} \left[\log \frac{p_\theta(x, y, z)}{q_\phi(z, y \mid x)} \right].$$

Furthermore, we have:

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

Hence:

$$\log p_\theta(x) = \text{ELBO} + \text{KL}\big(q_\phi(y, z \mid x) \,\|\, p_\theta(y, z \mid x)\big).$$

In the other hand, we have:

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

Using the fact that $q_\phi(z, y \mid x) = q_\phi(z \mid x, y) \cdot q_\phi(y \mid x)$ and $p_\theta(x, y, z) = p_\theta(x \mid z)p_\theta(z \mid y)p_\theta(y)$, we have:

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

Breaking the expectation into terms:

$$\begin{aligned}\text{ELBO} = &\int \log p_\theta(x \mid z) q_\phi(z \mid x, y) q_\phi(y \mid x)\, dz\, dy \\ &+ \int \log \frac{p_\theta(z \mid y)}{q_\phi(z \mid x, y)} q_\phi(z \mid x, y) q_\phi(y \mid x)\, dz\, dy \\ &+ \int q_\phi(z \mid x, y)\, dz \, \log \frac{p_\theta(y)}{q_\phi(y \mid x)} q_\phi(y \mid x)\, dy.\end{aligned}$$

Using that $q_\phi(z \mid x, y)$ is a probability distribution in z and reorganizing into expectations:

$$\text{ELBO} = \mathbb{E}_{q_\phi(y|x)} \mathbb{E}_{q_\phi(z|x,y)} \left[\log p_\theta(x \mid z)\right] - \mathbb{E}_{q_\phi(y|x)} \left[\text{KL}\left(q_\phi(z \mid x, y) \,\|\, p_\theta(z \mid y)\right)\right]$$

$$-\text{KL}\left(q_\phi(y \mid x) \,\|\, p_\theta(y)\right)$$

D Metrics for Clustering

Adjusted Rand Index (ARI) The ARI measures the similarity between two clusterings by quantifying their agreement while adjusting for chance. The formula is given by:

$$\text{ARI} = \frac{\sum_{ij} \binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{n}{2}},$$

where n_{ij} is the number of samples in both predicted cluster i and true cluster j, a_i and b_j are the sums over rows and columns of the contingency table, and n is the total number of samples. The ARI ranges from -1 to 1, where 1 indicates perfect agreement.

Normalized Mutual Information (NMI) NMI evaluates the similarity between predicted and true clusterings as a ratio of their mutual information to the average entropy. It is defined as:

$$\text{NMI} = \frac{I(P; T)}{\sqrt{H(P) H(T)}},$$

where P and T are the predicted and true cluster label distributions, $I(P; T)$ is their mutual information, and $H(P)$ and $H(T)$ are their entropies. NMI ranges from 0 to 1, with 1 indicating perfect agreement.

Silhouette Score The silhouette score assesses how well samples are clustered by comparing the average intra-cluster distance $a(i)$ to the average nearest-cluster distance $b(i)$:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}},$$

where $a(i)$ is the average distance from sample i to others in the same cluster, and $b(i)$ is the average distance to the nearest other cluster. It ranges from -1 to 1, where 1 indicates well-separated clusters.

Homogeneity Homogeneity measures whether each cluster contains only samples of a single class. It is given by:

$$\text{Homogeneity} = 1 - \frac{H(C \mid K)}{H(C)},$$

where $H(C \mid K)$ is the conditional entropy of the classes given the clusters, and $H(C)$ is the entropy of the classes. A score of 1 indicates perfect homogeneity.

Completeness Completeness ensures that all samples of a single class are assigned to the same cluster. It is defined as:

$$\text{Completeness} = 1 - \frac{H(K \mid C)}{H(K)},$$

where $H(K \mid C)$ is the conditional entropy of the clusters given the classes, and $H(K)$ is the entropy of the clusters. A score of 1 indicates perfect completeness.

V-Measure The V-measure is the harmonic mean of homogeneity and completeness:

$$\text{V-Measure} = 2 \cdot \frac{\text{Homogeneity} \cdot \text{Completeness}}{\text{Homogeneity} + \text{Completeness}}.$$

It balances the two aspects and ranges from 0 to 1, with higher values indicating better clustering quality.