LIÈGE université Library

https://lib.uliege.be

https://matheo.uliege.be

# Structured Representation Learning for Cytometry: Cell Annotation and Population Discovery

**Auteur :** Bodart, Fanny
**Promoteur(s) :** Louppe, Gilles
**Faculté :** Faculté des Sciences appliquées
**Diplôme :** Master en ingénieur civil biomédical, à finalité spécialisée
**Année académique :** 2024-2025
**URI/URL :** http://hdl.handle.net/2268.2/23237

*Avertissement à l'attention des usagers :*

*Tous les documents placés en accès ouvert sur le site le site MatheO sont protégés par le droit d'auteur. Conformément aux principes énoncés par la "Budapest Open Access Initiative"(BOAI, 2002), l'utilisateur du site peut lire, télécharger, copier, transmettre, imprimer, chercher ou faire un lien vers le texte intégral de ces documents, les disséquer pour les indexer, s'en servir de données pour un logiciel, ou s'en servir à toute autre fin légale (ou prévue par la réglementation relative au droit d'auteur). Toute utilisation du document à des fins commerciales est strictement interdite.*

*Par ailleurs, l'utilisateur s'engage à respecter les droits moraux de l'auteur, principalement le droit à l'intégrité de l'oeuvre et le droit de paternité et ce dans toute utilisation que l'utilisateur entreprend. Ainsi, à titre d'exemple, lorsqu'il reproduira un document par extrait ou dans son intégralité, l'utilisateur citera de manière complète les sources telles que mentionnées ci-dessus. Toute utilisation non explicitement autorisée ci-avant (telle que par exemple, la modification du document ou son résumé) nécessite l'autorisation préalable et expresse des auteurs ou de leurs ayants droit.*

**University of Liège**
School of Engineering and Computer Science

# Structured Representation Learning for Cytometry: Cell Annotation and Population Discovery

Master's Thesis Conducted by

**Fanny Bodart**

With the aim of obtaining the degree of
*Master of Science in Biomedical Engineering*

Under the supervision of
**Pr. Gilles Louppe**

With the assistance of
**Dr. Adrien De Voeght**

**Academic Year 2024-2025**

# Acknowledgments

I would first like to thank my supervisor, Gilles Louppe, who has been of precious advice throughout the year. Thank you for having, in addition to an outstanding understanding of the subject, the patience and the eagerness to transmit it. You guided and supported me perfectly; you made me deeply love a field I initially wasn't cut out for. It is also your trust in me that built this passion and desire to continue in this direction. I am looking forward to our future work together.

How could I not thank Adrien De Voeght, for having deeply comforted me in my decision to combine the world of engineering with that of medicine, for our discussions in your office and for the discovery of the harsh reality of medical practice (at the cost of a bump on my head). Never leave Montefiore, please.

Thanks to the best engineering students whom I can gladly call friends. Julien, for being cut from the same cloth as I am, but also for your always sharp discussions on a thesis that isn't even yours (teach me please). Our last collaboration hasn't happened yet! To Ahmed, for those roaring laughters, I guess the B52 won't sound the same for quite a while: I cannot thank the university enough for putting you on my path. To Clémence, Linpha, Lucas, Renaux, Maxime, to all of you, you were the perfect people to escape with during breaks. Thank you for sharing my studies: if I had to do it all over again, I would still choose you.

Thank you to my loving parents for their support, encouragement, and understanding at every turn of my academic path. If I've been a burden during all these exam sessions, at least you didn't make me feel like one. Thank you also to Alexis, Elodie, Ben, Brandon & Carla, for reminding me that studies are not the epicenter of life, thank you for being there, simply you: I love you all.

To all of you: this work is also yours.

"*There is no way around the hard work. Embrace it.*" — Roger Federer.

# Contents

# Abstract

Flow cytometry enables the characterization of cell types based on the expression of specific surface and intracellular markers. It is widely used in both research and clinical settings to analyze cell populations. Recent advances in the field now allow the simultaneous measurement of numerous markers, resulting in high-dimensional datasets. Thus, the conventional manual gating approach is no longer suitable for analyzing such complex data. While several machine learning methods have been proposed for automated cell classification, most focus solely on known populations. Conversely, unsupervised methods can discover novel subpopulations but lack interpretability and do not support direct annotation.

In this work, we propose a model capable of addressing these complementary goals within a unified semi-supervised framework. Our approach leverages structured representation learning through a deep generative model to achieve (1) accurate classification of known immune cell populations, (2) discovery of novel subpopulations, and (3) characterization of immune population dynamics across experimental conditions.

We introduce MARVIN: *Structured Representation Learning for Cytometry: Cell Annotation and Population Discovery*, a mixture-based variational autoencoder with a latent space explicitly structured by cell type. By modeling the latent space as a Gaussian mixture, MARVIN enables both annotation and subpopulation discovery within a unified framework.

To evaluate its performance, we benchmark MARVIN on public cytometry datasets and compare it to Scyan (Blampey et al., 2023), a recent generative model designed for cytometry data. We assess MARVIN's ability to recover masked subpopulations specific to peanut allergy and analyze immune response dynamics before and after allergen exposure. MARVIN reliably identifies relevant novel (unseen) subpopulations and captures their shifts across different experimental conditions.

This dual functionality makes MARVIN a powerful tool for both exploratory research and routine clinical analysis. We plan to apply this framework to investigate immune activation patterns in an ongoing clinical trial focused on vaccine response in immunocompromised patients.

# Introduction

## Context and Problem Statement

Flow cytometry can be described as the characterization of cell types based on the expression of specific surface and intracellular markers. It is widely used in both clinical practice and research settings and has contributed significantly to advances in medicine and biology, particularly in disease diagnosis, immune monitoring, and treatment assessment.

In recent years, technological progress in the field has been remarkable, enabling the simultaneous measurement of a large number of markers. This has led to the generation of high-dimensional datasets containing a substantial number of parameters per cell.

Analyzing such data typically involves two objectives: (1) assigning labels to patient cells based on their marker expression profiles to identify the populations present in blood or bone marrow samples, and (2) discovering rare or previously unknown subpopulations of clinical interest, such as specific immune cell subsets or malignant cancer cells.

Traditional analysis methods, such as manual gating, are no longer optimal for such high-dimensional data. This has motivated the development of automated tools, which can be broadly divided into two categories: supervised algorithms, which aim to automatically annotate cells and replicate expert labeling, and unsupervised clustering methods, which group cells based on their intrinsic similarities and enable subpopulation discovery, but without providing interpretable annotations.

## Objectives

The goal of this master's thesis is to develop a model that can simultaneously

- automatically annotate patient cell populations, with the aim of replacing the tedious and biased manual gating process,
- discover novel, rare subpopulations of clinical interest.

This would enable the creation of a unified framework that combines the strengths of both supervised annotation and unsupervised discovery approaches.

## Document Organization

This document is divided into five distinct chapters.

The first, **Analyzing Cytometry Data (1)**, provides a solid theoretical foundation on flow cytometry, detailing how the data are generated and traditionally analyzed. It also includes a review of existing automated techniques and their limitations, highlighting the need for a unified approach such as the one proposed in this work.

The second chapter, **Representation Learning (2)**, introduces the theoretical background of representation learning through deep generative models.

Building on this, the third chapter, **Structured Representation Learning (3)**, presents our strategy of structuring the latent space, therefore tailored to the specific nature of cytometry data. It introduces our model, MARVIN[1], and positions it within existing approaches.

The fourth chapter, **Experiments and Results (4)**, reports the experimental results related to our main objectives, across multiple datasets and experimental settings.

Finally, the last chapter, **Discussion and Perspectives (5)**, concludes and discusses our model by defining how the initial objectives have been met and provides insights of the integration of our model into real-world clinical and research settings.

[1] GitHub : https://github.com/fannybdt/MARVIN

# Chapter 1

# Analyzing Cytometry Data

This chapter establishes the biological and methodological foundations relevant to our developed approach. We begin by introducing the principles of cytometry and its role in enabling precise characterization of cellular phenotypes. Particular attention is given to the standard manual gating process and its inherent limitations, especially in the context of high-dimensional data.

To conclude, a detailed survey of computational methods in cytometry data analysis is presented, highlighting their key contributions alongside persistent challenges.

## 1.1 An Introduction to Cytometry

The analysis of cell characteristics, or *Cytometry*, is a tool tailored for a wide range of disciplines, such as immunology, molecular biology, infectious disease monitoring, and cancer biology. It is primarily used in blood tests to count and characterize cells based on their size, morphology and specifics markers of interest such as their surface or intracellular proteins.

In a clinical context, it is a powerful technique for medical diagnosis, phenotyping, and monitoring minimal residual disease in patients with hematologic cancers for example. Cytometry is also commonly used in fundamental research in cell biology to assess the impact of a treatment on cell populations or to analyze the effect of an immunogen on different immune cells, to name just a few examples (McKinnon, 2018).

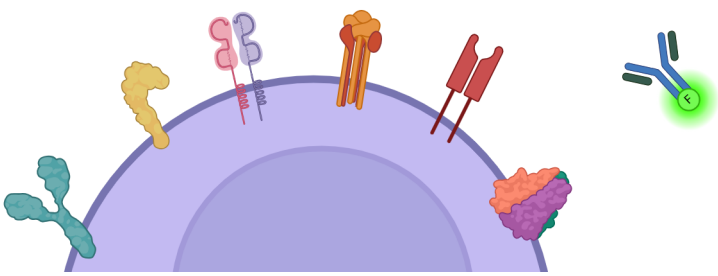


**Figure 1.1: A cell and its surface markers** The expression of surface markers (or proteins) is specific to each cell. An antibody conjugated to a fluorochrome can be used to stain a particular protein.

The reason why cytometry is essential in many areas of biomedical research is that it enables the precise distinction of cell subpopulations that were once assumed to be homogeneous but actually carry unique features, resulting for example in different disease phenotypes. Greater characterization of immune subpopulations allows for more informed decisions regarding the identification of targetable biomarkers and the development of new therapeutic approaches. For example, in the context of acute leukemia (a type of blood cancer), cytometry is used to make the diagnosis, help to discover the precise nature of the leukemia (myeloid, lymphoblastic B or T), select a therapeutical target such as the CD20 (targeted by the Rituximab) and is critical in the management by detecting the minimal residual disease (MRD), refering to the small number of cancerous cells that persist in a patient after treatment and cannot be detected using standard diagnostic methods (e.g. bone marrow smears).

Cytometry exists in two main variants for single-cell analysis: *flow cytometry* and *mass cytometry*. The two techniques differ not in the analysis of the resulting data, but rather in the data acquisition process and the number of cellular markers that can be analyzed.

Both techniques are based on the same concept : the detection of specific cell components (markers) expressed at the surface or inside the cell. The combination of all expressed markers allows to identify a unique signature that characterizes precisely the nature of the cell (Figure 1.1).

## 1.2 Flow Cytometry

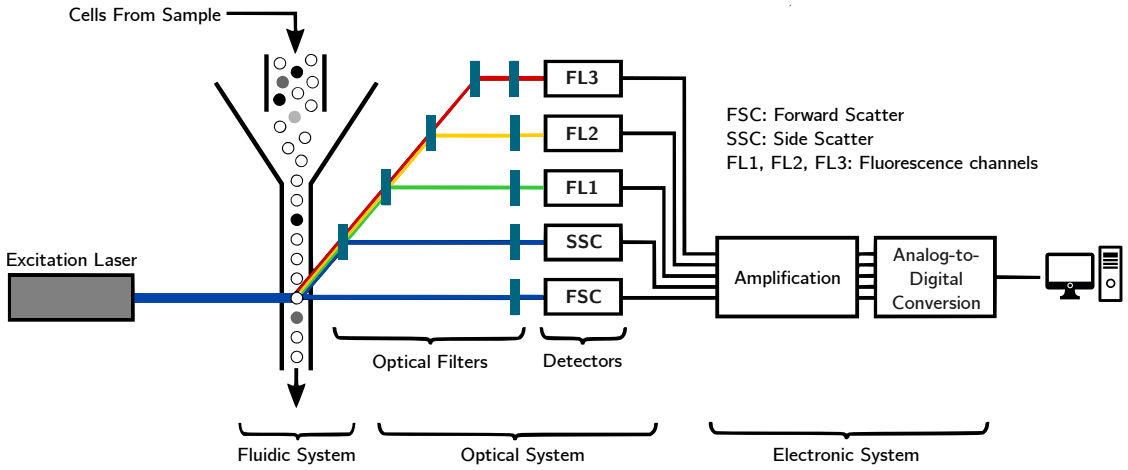



**Figure 1.2: Schematic representation of the flow cytometry process.**[a] Cells are focused into a single stream and pass before the excitation lasers. Laser illumination results in both scatter (forward (FSC), side scatter (SSC)) and fluorescence. Light is directed to the appropriate detectors by dichroic mirrors and filters. Following detection, emitted signals must be processed and analyzed.

[a]Figure modified from ATT Bioquest, Inc

Flow cytometry (Figure 1.2) enables the precise study of isolated particles carried by a liquid stream. It is a technique for individual, quantitative, and qualitative characterization of particles suspended in a liquid. The principle involves analyzing the optical or physical signals emitted by a particle as it intersects a laser beam.

Samples are first prepared for fluorescence measurements by staining them with *fluorescently conjugated antibodies* (Bendall et al., 2012). They are selected in advance to identify populations

of interest: indeed, an antibody is specific to a certain antigen (in this context, a cellular marker such as a surface protein) which allows one cell population to be distinguished from another (see Figure 1.3). This antibody is conjugated to a fluorochrome that emits light at a specific wavelength, ideally one that is easily distinguishable from those of other fluorochromes.

Flow cytometry allows the analysis of two types of parameters: *morphological parameters* and *fluorescence parameters*.

Each cell passes through different excitation lasers. Visible light scatter (independent of fluorescence) is measured in two different directions:

1. at the forward scatter (FSC), which indicates the relative size of the cell
2. at the 90° angle (Side Scatter or SSC), which reflects the internal complexity or granularity of the cell

Fluorescence emission can be spontaneous (cellular autofluorescence) or induced by fluorochromes excited by the lasers. The emitted light is split by a system of mirrors and optical filters and directed into multiple detection channels (varies depending on the cytometer, but typically 4 to 8 channels) , where it is captured by distinct photomultiplier tubes (PMTs), each tuned to a specific wavelength range. These PMTs convert the incoming light into electrical signals corresponding to the intensity of fluorescence in each spectral window.

Once the electrical signal is amplified, the amplitude of the electrical pulse generated by each cell can be measured, with each numerical value corresponding to a channel. Each fluorescence channel is associated with a fluorochrome (through a specific wavelength, 530nm for FITC for ex., a commonly used fluorochrome for immune cells), which is itself conjugated to an antibody specific to a particular biological marker (protein). From the electrical intensity detected in all fluorescence channels, one can precisely characterize the proteins expressed by a cell, as well as its morphological parameters.

The advantage of flow cytometry is that it allows for the analysis of tens of thousands of cells per second, resulting in a total of several hundred thousand to millions of cells analyzed in a single experiment.

### 1.2.1 Spectral Flow Cytometry

A challenge arises in conventional flow cytometry: the fluorochromes used do not emit a narrow and easily detectable spike, but rather a broad emission spectrum. As a result, each detector only measures fluorescence within a specific range of wavelengths : any fluorescence signal outside of that range is not captured. Moreover, due to the broad emission spectra, the fluorescence detected by some channels may include signals from fluorochromes not originally intended for them. This spillover must be corrected through a compensation process, a crucial step in data analysis whose effectiveness depends on the number of overlapping markers used (Drescher et al., 2021).

This spectral overlap in detection channels significantly limits the number of markers that can be simultaneously used in conventional flow cytometry. Spectral flow cytometry was precisely designed to overcome this physical limitation.

In spectral cytometry, many more detectors are used (up to 186): signals from all channels are collected, regardless of the number of fluorochromes being analyzed. For each cell, fluorescence

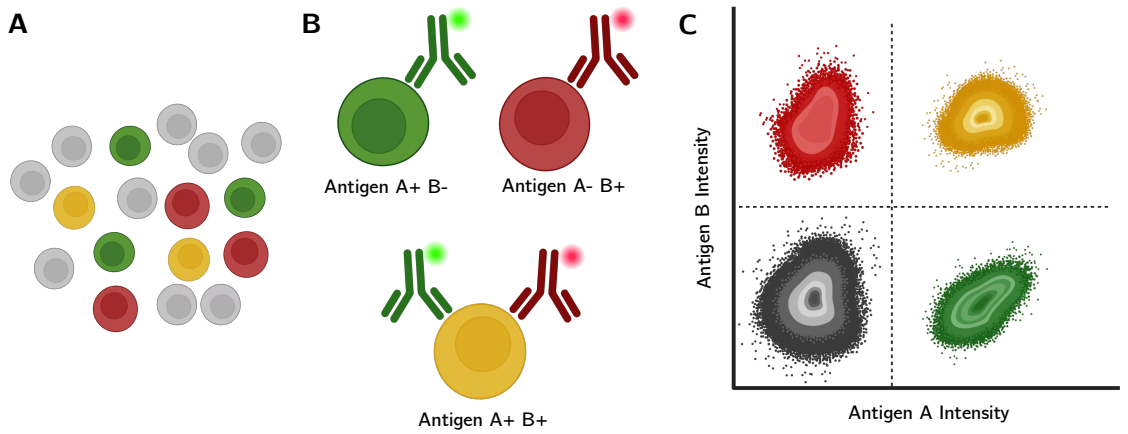



**Figure 1.3: Immunophenotyping with flow cytometry.**[a] **A**, A mixed population of cells, which may be distinguished by the expression of different antigens on their surface. **B**, Staining with fluorochrome-conjugated antibodies, each specific to an antigen - biological markers used to characterize the cell. **C**, Flow cytometry characterizes cell populations via fluorescence intensity from antibody-labeled cellular markers.

[a]Figure from Antibodies, Inc

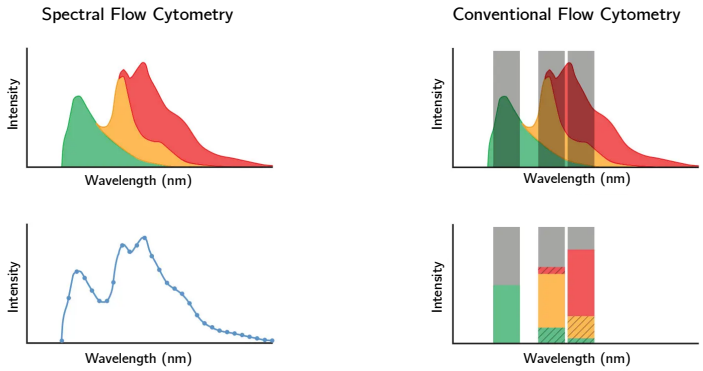



**Figure 1.4: Spectral vs conventional flow cytometry.**[a] Spectral flow cytometry detects the full fluorescence spectrum of each cell rather than fluorescence within predefined detection ranges. This helps resolve the issue of fluorochrome spillover into detection channels not intended for them.

[a]Figure from Sony

is thus measured at numerous points along the spectrum (Figure 1.4), allowing reconstruction of the complete fluorescence profile of the cell (including the emission from all its fluorochromes and its autofluorescence). Then comes the spectral unmixing step: using single-stained samples, the spectral signature of each fluorochrome is determined and separated using a weighted least squares method.

This allows one to combine fluorochromes with similar peak emission but distinct full emission signatures into the same panel, thus greatly expanding upon the flow cytometry capabilities (Liu et al., 2020). This technique has enabled a shift from analyzing a handful of markers to over 40, allowing for detailed characterization of cells, which in turn enables the identification of highly specific subpopulations.

## 1.3 Mass Cytometry

Another, more expensive cytometry technique emerged prior to spectral flow cytometry: mass cytometry (cytometry by time of flight, or CyTOF). The principle involves using antibodies coupled to rare metal isotopes instead of fluorochromes (Bandura et al., 2009). Each metal has a unique mass signature, which is detected by a time-of-flight mass spectrometer, allowing the simultaneous quantification of over 50 markers *without spectral overlap*.

As a result, there is no longer a need for unmixing or compensation algorithms, nor for the tedious selection of a fluorochrome panel: the Time-of-Flight analyzer avoids spectral overlap altogether, since metals have distinct masses (Figure 1.5). This ultimately reduces noise from intrinsic signals and unmixing artifacts. This precise method is however less commonly used due to its higher cost, and is now rivaled by spectral flow cytometry.

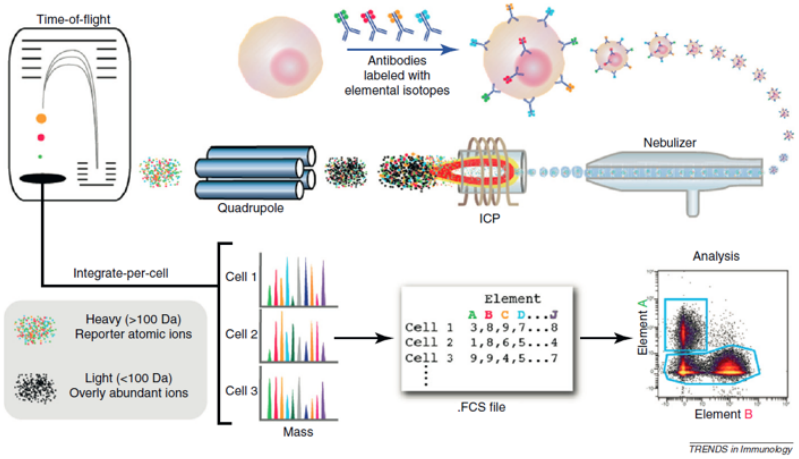



**Figure 1.5: The mass cytometry process.**[a] Cells are tagged with rare metal isotopes, each with a distinct mass measured by the time-of-flight analyzer, allowing simultaneous detection of many markers without spectral overlap.

[a]Figure from Bendall et al., 2012

## 1.4 Manual Gating and its Limitations

Once cytometry data acquisition is complete, the analysis of cell populations is carried out most often by experts through a process known as *manual gating*.

As shown in Figure 1.6, manual gating is typically performed using a series of two-dimensional scatter plots (dot plots), where two markers are selected at a time based on prior biological knowledge—often well-established lineage or activation markers. A region (gate) is drawn around a subset of cells in the plot, identifying them as positive or negative for the selected markers. For example, a gate may isolate cells that are $CD3^+$ and $CD19^-$ to define the T cell compartment, or $CD14^+$ and $CD16^+$ to capture a specific monocyte subset. This process is iterative: after an initial population is gated, additional markers are used in subsequent plots to further refine the cell subset. At each step, cells are either included or excluded based on the presence (positive expression), absence (negative expression), or level (low, intermediate, high) of marker expression. Over time, this strategy builds a hierarchical classification of cell types and subtypes based on complex combinations of marker expression.

The gating strategy (i.e., sequential pairwise marker selection) is specific to each expert and therefore experience-based, even though some general guidelines do exist (Staats et al., 2019, Finak et al., 2016). Nevertheless, in more specific cases such as diagnosis, these guidelines are most often disease-specific, requiring particular expertise for each disease subtype. Moreover, for a given disease, the gating strategy differs between initial diagnosis and residual disease monitoring (Heuser et al., 2021). It is important to note that Figure 1.6 provides a highly simplified view

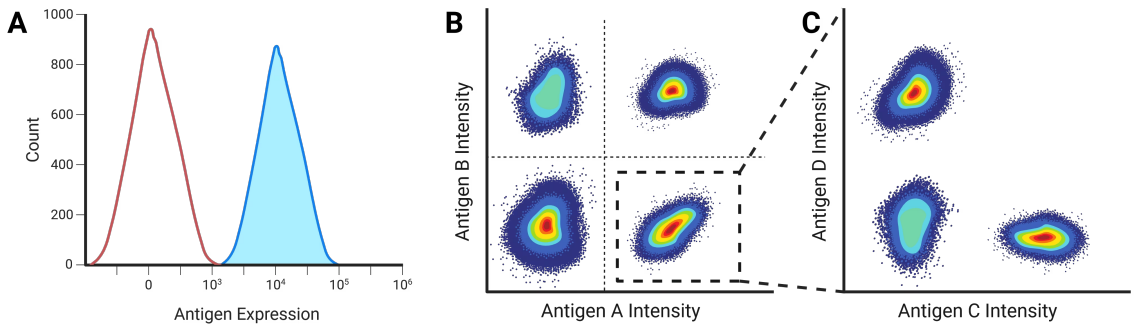



**Figure 1.6: Simplified one and two parameters displays of flow cytometry data.**[a]
**A**, Single-marker plots can be used to separate populations based on the expression of a single antigen. **B**, Two parameter plots identify populations by the combinatorial expression of two markers (antigens). **C**, The iterative gating process: a specific cell population can be further investigated by analyzing the expression of two additional antigens (here, C and D), with the goal of refining into subpopulations. This process can be repeated.
[a]Figure from Antibodies, Inc

that does not reflect biological reality. While some cell populations may occasionally appear clearly separated from others, this is far from the norm. Figure 1.7 illustrates the gating strategy applied across a 40-marker panel for multiple cell populations. Rather than forming well-separated clusters, cell populations tend to form a continuum, with overlapping distributions that rarely result in clean boundaries on bivariate plots. This is where the gating strategy becomes particularly subjective: one must draw a gate to separate two populations that often visibly overlap in terms of marker expression, especially in the tails of their distributions.

This phenomenon arises from the fact that cells are not static entities. Instead, they express dynamic phenotypes that vary depending on factors such as their activation state, maturation stage, cellular environment, or exposure to stimuli. The assumption that a fixed marker expression profile corresponds to a well-defined cell type is overly simplistic. In reality, phenotypic expression exists along a spectrum, making the task of identifying and classifying cells based solely on discrete gating thresholds especially challenging.

Moreover, while cytometry experiments conducted within a single laboratory using a consistent setup and gating strategy can yield reproducible results, extending such analyses across laboratories introduces substantial variability (Kalina, 2020). Even with proposed standardized gating protocols (Finak et al., 2016), significant differences in outcomes still persist.

This issue becomes even more critical as the number of markers increases. With the rise of technologies capable of generating high-dimensional data (with over 40 markers) the task of manual gating has become excessively time-consuming and challenging (illustrated in Figure 1.7). In such high-dimensional settings, not only does manual gating require the user to define a tedious and carefully ordered sequence of marker-based decisions, but it also fails to capture important information arising from interactions between multiple markers. The traditional strategy based on

bivariate marker displays (two-parameter plots) is fundamentally limited in this regard. Indeed, a cell's marker signature is more than the sum of pairwise marker combinations: there is a real need to analyze all relevant dimensions simultaneously in order to fully capture the complex phenotype of a single cell.

These limitations, along with the potential for improvement, have underscored the necessity of developing techniques capable of faithfully analyzing and interpreting high-dimensional cytometry data. An overview of such approaches is presented in Section 1.5.

## 1.5 Automated Analysis of Cytometry Data

This section aims to present the state of the art in automated cytometry data analysis algorithms, both for classification tasks and for cell population discovery. We will highlight the shortcomings of existing techniques and motivate the need to develop an extensive method such as ours.

To address previously highlighted challenges (Section 1.4), numerous automated tools have been developed to support cell clustering and annotation in these high-dimensional cytometry datasets.

These methods can be broadly divided into two categories: supervised and unsupervised approaches.

### 1.5.1 Supervised Algorithms

The supervised models use labeled training data (from manual gating) or marker knowledge to automate cell-type annotation in cytometry. Key methods include:

- LDA (Abdelaal et al., 2019): a simple Linear Discriminant Analysis classifier that achieves competitive performance despite its simplicity,
- CyAnno (Kaushik et al., 2021): trains binary classifiers per cell type using models like XGBoost or SVM,
- DeepCyTOF (Li et al., 2017): employs deep neural networks combining denoising autoencoders with feedforward classifiers,
- HemaGraph (Bini et al., 2024): a graph-based method using Graph Attention Networks (GATs) on a k-nearest neighbor graph of cells for classification,

These models suffer from the major drawback of only being able to annotate cell populations that were present in the training set: they cannot be used to discover new populations.

**Knowledge-based Models** Another class of supervised methods leverages user-defined marker tables encoding known cell type/marker relationships, rather than relying on manual labels. Examples include:

- ACDC (Lee et al., 2017): applies unsupervised clustering (e.g., GMM or k-means) followed by rule-based label assignment using the marker table,

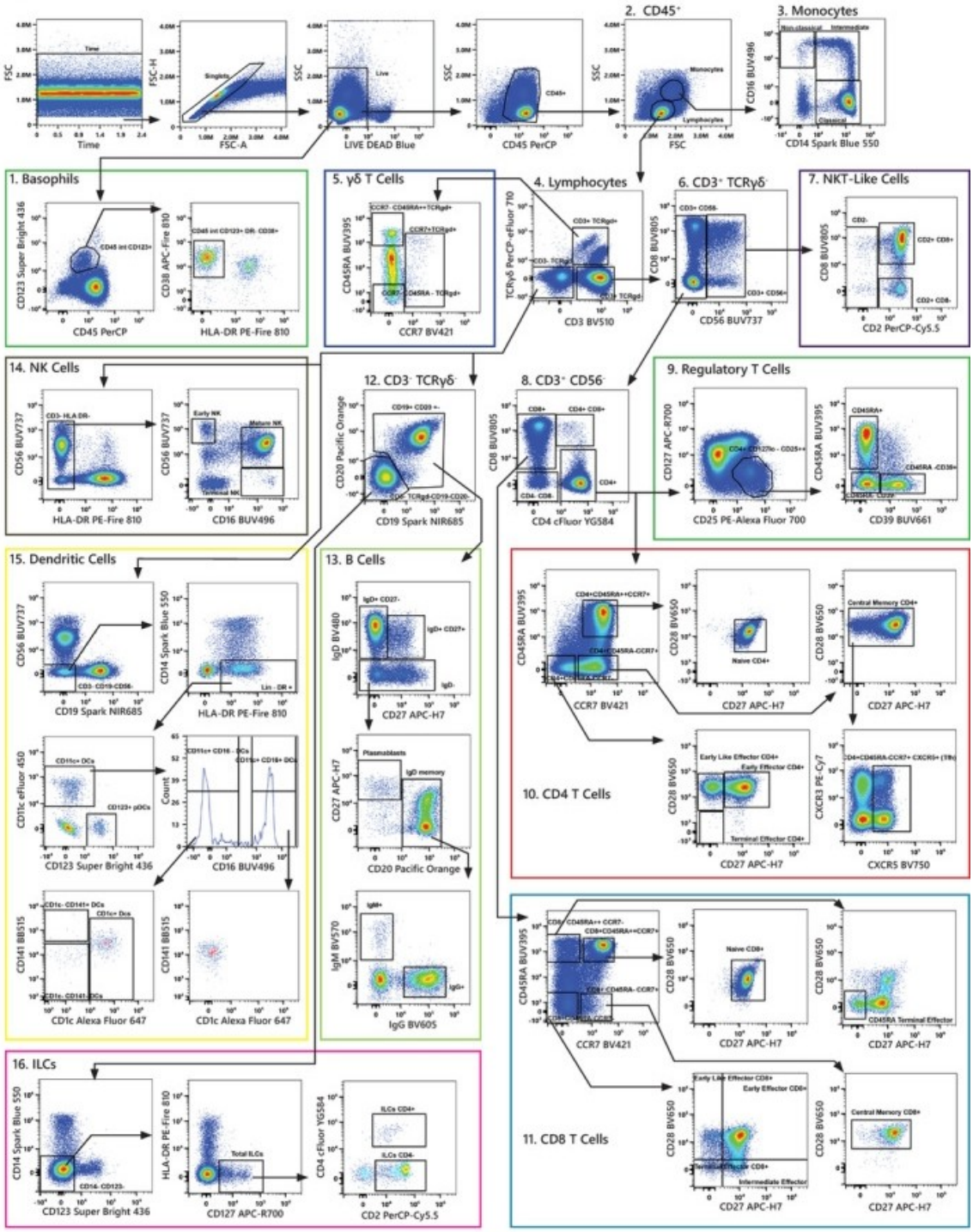



**Figure 1.7: Example of a complete manual gating process, on a 40 color panel.**[a] Manual gating strategy was used to identify and classify major cellular subsets, starting with the exclusion of doublets and dead cells. Populations were sequentially gated and refined to isolate specific subpopulations, including basophils, monocytes, lymphocytes, T cells, B cells, NK cells, dendritic cells, and innate lymphoid cells (ILCs). The gating process involved various steps of subdivision based on marker expression, leading to the identification of distinct subsets.

[a]Park et al., 2020, Figure 1

- MP Bayesian Trees (Ji et al., 2018): builds interpretable probabilistic decision trees where each split is informed by marker knowledge.

While these approaches improve flexibility by allowing user-specified populations, they still cannot autonomously identify novel cell types (i.e. that were not specified a priori in the knowledge table).

It should be noted that, although using a marker table may seem less biased and more convenient than relying on pre-manual gating data, the discrete nature of the marker table (-1, +1, or 0 for "irrelevant for annotation") does not necessarily reflect biological reality or experimental conditions.

In practice, the biological expression of markers is not strictly binary: cells often exhibit a continuum of expression levels due to regulatory noise, activation states, differentiation stages, or environmental cues. This plasticity can lead to intermediate or transient marker expressions that challenge rigid annotations. Moreover, constructing accurate marker tables still relies on expert knowledge and carries a degree of subjectivity, making it important to interpret such tables with caution and to consider more flexible models that capture the continuous and dynamic nature of cellular phenotypes.

### 1.5.2 Unsupervised Clustering Approaches

Another approach consists in using unsupervised automated data analysis algorithms, namely clustering methods.

Unlike manual gating, where populations of interest are separated sequentially, clustering algorithms identify these populations automatically, and it is up to the user to interpret the biological relevance of each cluster based on overall marker expression (Blampey et al., 2023).

As such, a number of automated tools have been developed including

- PhenoGraph (Levine et al., 2015), building a k-nearest neighbor (k-NN) graph from the data and applies the Louvain algorithm for community detection,
- SPADE (Qiu et al., 2011), performing density-dependent downsampling, constructs a minimum spanning tree, and then clusters cells based on hierarchical relationships,
- FlowSOM, combining Self-Organizing Maps (SOMs) for dimensionality reduction and initialization with meta-clustering (typically using hierarchical clustering) to identify cell types,
- FlowMeans (Aghaeepour et al., 2011), a variant of k-means with automatic selection of the number of clusters,
- scVAE (Grønbech et al., 2020), using a variational autoencoder that models single-cell transcriptomic data with a Gaussian mixture prior in the latent space (as described in Section 3.2). It factorizes the posterior to infer soft cluster assignments and trains end-to-end by maximizing a variational lower bound, allowing clustering to emerge purely from the optimization process without any supervision.

Unsupervised approaches allows for the detection of novel cell types and rare cell populations.

However, the issue with such methods is that, although they are meant to address the bias introduced by user-specified labels or marker tables, a manual analysis of marker expression is still required to assign each cluster a meaningful cell type (Levine et al., 2015, Blampey et al., 2023).

### 1.5.3 Baseline: Scyan

One of the most recent models for cytometry data analysis is Scyan (Blampey et al., 2023). Scyan is a deep generative model based on a normalizing flow architecture that maps cytometry marker expression data into a so-called "biological relevant latent space".

The authors describe the model as successful in both the tasks of annotating cell populations and discovering new subpopulations.

Although Scyan claims to be an unsupervised approach, it is in fact a hybrid method: it is a supervised *knowledge-based model* that uses a marker table to structure its latent space, while still allowing for the discovery of subpopulations descending from a user-specified parent population in an unsupervised manner using Leiden clustering (Traag et al., 2019).

We chose Scyan as our main comparator because, like our approach, it relies on a deep generative model. Also, this recent method claims to outperform other state-of-the-art models (ACDC, MP, CyAnno, LDA) in automatic cell type annotation.

Although Scyan is a knowledge-based model, similar to ACDC and MP, it handles the knowledge table more effectively. Specifically, the user provides marker expression values as +1, -1, or intermediate real values, but the model also explicitly accounts for missing values (NaN) when the expression of a marker is unknown or irrelevant for characterizing a given cell type.

In Scyan, each cell is represented by its vector of marker expressions along with any additional covariates (cell autofluorescence, ...). A deep generative model, a *coupling-layers* normalizing flow $f_\phi$, maps these input vectors into a structured latent space.

This latent space shares the same dimensionality as the original marker space and is constructed based on the biological knowledge table. Once a cell is projected into this latent space, annotation is performed by selecting the most probable population, characterized by a Gaussian-like distribution centered on vertices of a hypercube corresponding to known cell types. The generative process of Scyan can described as

$$
\begin{aligned}
& & Z &\sim \text{Categorical}(\boldsymbol{\pi}) \\
\boldsymbol{E} \mid Z = (e_m)_{1\leq m\leq M}, \text{ where } & \begin{cases} e_m = \rho_{Z,m} & \text{if } \rho_{Z,m} \neq \text{NA}, \\ e_m \sim \mathcal{U}([-1,1]) & \text{otherwise}, \end{cases} \\
& & \boldsymbol{H} &\sim \mathcal{N}(\mathbf{0}, \sigma \boldsymbol{I}_M) \\
& & \boldsymbol{U} &= \boldsymbol{E} + \boldsymbol{H} \\
& & \boldsymbol{X} &= f_\phi^{-1}(\boldsymbol{U}),
\end{aligned}
$$

where $Z$ is the random variable corresponding to a cell type among the P possible ones, $\boldsymbol{E}$ is a population-specific variable whose terms are either known according to the expert knowledge table $\rho$ or drawn from a uniform distribution and $\boldsymbol{H}$ contains cell-specific terms (covariates). Finally, $\boldsymbol{U}$ is the cell's latent expressions, summing a population-specific component and a cell-specific one.

**Discussion.** In Blampey et al., 2023, Scyan is described as an unsupervised technique that leverages the bias introduced by annotations derived from manual gating. However, this characterization is misleading. The definition of the so-called knowledge table is itself inherently biased, as it is constructed based on a strategy that inevitably reflects expert decisions. As such, Scyan's cell

annotation process is, in fact, supervised, even though it does not rely directly on cell-by-cell labels.

Moreover, Scyan claims that its latent space is biologically interpretable and that downstream analyses on cell types can be directly performed within this space. However, if a marker is not specified in the knowledge table, Scyan samples its expression uniformly for the corresponding cell type ($e_m \sim \mathcal{U}([-1, 1])$), which compromises the interpretability of that marker and may introduce significant biases in the analysis. As a result, the information carried by non-specified markers becomes essentially discarded, even though these markers could still contribute meaningfully to the analysis despite not being included in the detection strategy defined by the knowledge table.

Scyan also claims to address the task of subpopulation discovery. This is somewhat surprising, as its generative model is not inherently designed for such a purpose. Instead, this task is handled separately, via an external unsupervised algorithm: Leiden clustering (Traag et al., 2019). A specific population is selected and subdivided using this independent clustering method. The resulting subpopulations are then characterized using Scyan's latent space, which, as previously discussed, lacks robust biological interpretability. Therefore, Scyan does not provide a truly *unified framework* for both annotation and subpopulation discovery, despite such claims.

### 1.5.4 Summary and Introduction of a Mixed Approach

Unsupervised techniques offer the ability to objectively group cells according to their marker expression profiles, thus avoiding biases associated with expert-driven annotation strategies. This enables the discovery of subpopulations characterized by distinct features that may not have been considered through traditional expert approaches. However, the annotation of these clusters must still be performed afterward, which reintroduces a degree of subjectivity and supervision. Moreover, the interpretability of how certain clusters are formed can be limited (Liu et al., 2020).

Supervised techniques have the drawback of requiring labeled data (or user-specified marker tables), which inevitably introduce bias, especially when the manual gating used to annotate the data becomes tedious due to the high dimensionality of the dataset. However, these methods allow for the automatic annotation of cell populations, effectively replacing the manual gating process. Nonetheless, the cell populations that these models can annotate are limited to those defined in the training set or the knowledge table, which means they cannot discover novel subpopulations, something that can be highly valuable from a clinical or research perspective.

After highlighting the various advantages and disadvantages of using supervised or unsupervised algorithms, it becomes clear that **there is a need to combine the best of both worlds.** This is precisely what Scyan attempted to address by introducing a discovery mechanism alongside its annotation process. However, the model is not end-to-end: the discovery component operates independently from the classification module.

This highlights the need for a semi-supervised model that would combine both approaches in a **unified framework**.

# Chapter 2

# Representation Learning

The capacity of a model to perform complex reasoning or prediction tasks depends heavily on how it internally represents the data. **Representation Learning** refers to the process by which a model discovers and structures useful features of the input data, often in a lower-dimensional or more abstract "latent" space. These representations are intended to reflect the meaningful structure of the data and can serve as a foundation for tasks such as classification, clustering, or generative modeling.

In practice, such representations are often learned through the use of **Deep Learning**, which is a subfield of Machine Learning that leverages Neural Networks. They are computational architectures composed of interconnected units called neurons, arranged in successive layers. Each layer applies a parametric transformation to its input, and non-linear activation functions are inserted between layers to enable the learning of complex, non-linear functions. This layered structure allows the network to build hierarchical representations of the input data, where each layer extracts increasingly abstract features. The learning objective is to adjust the network's parameters so that the final output aligns with a desired target, whether it be a prediction, a reconstruction, or another objective depending on the task. The training process relies on the principle of *empirical risk minimization*, which consists in minimizing a loss function specifically defined for the task and architecture at hand. Optimization is performed through variants of gradient descent (most commonly mini-batch Stochastic Gradient Descent) which iteratively updates the parameters in the direction that reduces the objective function, namely the loss.

We begin with the concept of unsupervised learning, followed by an introduction to auto-encoders as a means of learning compressed data representations. We then present the Variational Auto-Encoder (VAE), a probabilistic model that extends auto-encoders by incorporating variational inference, allowing the model to learn structured latent spaces suited for both generation and representation.

## 2.1 Deep Unsupervised Learning

Common framework for many Machine Learning tasks is to use *supervised learning*, i.e. find a function $f_\theta$ of the inputs $\mathbf{x} \in \mathcal{X}$ that approximate at best the outputs $y \in \mathcal{Y}$. The mapping is built from a *labeled* dataset $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, and the ultimate goal is to estimate the conditional $P(Y = y|X = \mathbf{x})$ for any new inputs-output pair $(\mathbf{x}, y)$ (generalization performance).

However, this framework is inherently constrained by the type of supervision it receives: even though a model trained with supervised learning can generalize to unseen data, it can only effectively learn to represent those aspects of the data for which it has received an explicit supervision signal, or label.

This contrasts with how the human brain learns. We do not rely solely on labeled examples to build our understanding of the world. Instead, we learn by observing, predicting, interacting and extracting regularities from data without external supervision. This ability to build internal representations from raw sensory input, or to extrapolate using common sense, is what current artificial systems often lack.

Learning such internal and task-agnostic representations is precisely the goal of **unsupervised learning**. The idea is to model the underlying structure of the data in such a way that the learned features (or representations) capture meaningful and reusable information, even in the absence of labels.

Deep unsupervised learning refers to the use of deep models to learn such representations. Broadly speaking, it can be divided into two main paradigms:

- Self-supervised learning, or predicting any part of the input from any other part (future from past, occluded from the visible) for any tasks that require semantic understanding
- Generative models, that aims to recreate the raw data distribution.

Both paradigms aim to learn representations that reflect the essential structure of the data, but this work focuses on the second approach: generative models.

**Generative Models.** A deep generative model is a probabilistic model $p_\theta$ that can be used as a simulator of the data. The models defines a probability distribution $p_\theta(\mathbf{x})$ over the data $\mathbf{x} \in \mathcal{X}$, where the parameters $\theta$ are learned to match the full unknown data distribution $p(\mathbf{x})$. They can be used to produce new samples (generative process), evaluate densities (to perform inference) and encode complex priors.

## 2.2 Auto-Encoders

A key strategy in representation learning is to encode high-dimensional data into a lower-dimensional or more abstract space that preserves the most relevant features. Auto-encoders are specifically designed for this purpose.

An auto-encoder is a composite function made of

- an *encoder* $f$ from the original space $\mathcal{X}$ to a **latent space** $\mathcal{Z}$
- a *decoder* $g$, mapping from the latent space back to $\mathcal{X}$,

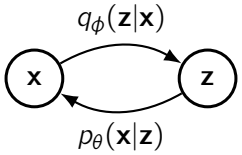
such that, after training, $g \circ f$ is close to the identity on the data.

The *latent space* refers to a transformed representation space in which a model learns to encode the essential features of the input data. It is therefore an learned abstraction that allows the model to uncover underlying structures in the data without necessarily having access to all the explicit information.

## 2.3 Variational Auto-Encoders

Standard auto-encoders learn a deterministic mapping from data $\mathbf{x}$ to a latent representation $\mathbf{z}$, optimized solely to minimize a simple reconstruction error. However, they do not constrain the geometry of the latent space: the encoded representations may form an irregular and discontinuous manifold, making it unsuitable for generative purposes.

Addressing these issues and adopting the optimization principles of variational inference, one obtains a powerful class of generative models: the **Variational Auto-Encoders (VAEs)**. They replace pointwise encodings with distributions $q_\phi(\mathbf{z}|\mathbf{x})$ and introduce a prior over the latent variables, $p(\mathbf{z})$.




In this framework, a neural network is used to model the probabilistic decoder $p_\theta(\mathbf{x}|\mathbf{z})$, which defines a stochastic generative process that maps latent variables $\mathbf{z} \in \mathcal{Z}$ to observations $\mathbf{x} \in \mathcal{X}$. This generative perspective mirrors the auto-encoder architecture, but now within a fully probabilistic setting where the latent space captures the underlying structure of the data in a principled way.

To fit $\theta$ parameters, a VAE amounts to maximizing the marginal likelihood of the data

$$p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})\, d\mathbf{z},$$

but this integral is generally intractable. A naive Monte Carlo estimate using samples $\mathbf{z}_i \sim p(\mathbf{z})$ leads to poor approximations in high dimensions, where most samples are unlikely under the posterior. This motivates the use of **variational inference**, which provides a tractable solution to this inference problem.

**Variational Inference.** The core idea is to approximate the intractable posterior $p_\theta(\mathbf{z}|\mathbf{x})$ and amortize the inference process by learning a second neural network $q_\phi(\mathbf{z}|\mathbf{x})$, a **variational distribution** conditioned on the observed input data $\mathbf{x}$ and parameterized by $\phi$. Instead of sampling from $p(\mathbf{z})$, the samples come from $q_\phi(\mathbf{z}|\mathbf{x})$ and one can rewrite the marginal likelihood using importance sampling

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

Then, using Jensen's inequality, that states that a concave function of the expectation of data $y$ is greater than or equal to the expectation of the function of the data, the log-likelihood can be lower bounded

$$\log p_\theta(\mathbf{x}) = \log \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\frac{p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] \geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] \quad (\text{ELBO}(\mathbf{x};\theta,\phi)),$$

which is the **evidence lower bound objective**, or ELBO of the log-likelihood. Maximizing the log-likelihood returns to maximizing the ELBO (which serves as a lower bound), which is a tractable surrogate objective. Although we do not maximize the exact log-likelihood, the ELBO is a principled and efficient alternative that ensures the model learns meaningful latent representations.

Using the Kullback-Leibler divergence $\mathrm{KL}(q(\mathbf{x})\|p(\mathbf{x})) = \mathbb{E}_{q(\mathrm{x})}\left[\log \frac{q(\mathrm{x})}{p(\mathrm{x})}\right]$, one can rewrite the ELBO as

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

a decomposition where the first term is the reconstruction loss (how well the decoder can reconstruct the input data) and the second is a prior-matching term : $q_\phi(\mathbf{z}|\mathbf{x})$ should be as close as possible to the prior over the latent space $p(\mathbf{z})$ in order to minimize the KL divergence.

Another interpretation of the ELBO can also be obtained using the Bayes rule

$$\mathrm{ELBO}(\mathbf{x};\theta,\phi)) = \log p_\theta(\mathbf{x}) - \mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z}|\mathbf{x})),$$

meaning that in order to maximize $\log p_\theta(\mathbf{x})$, the KL term should be as close to 0 as possible, so $q_\phi(\mathbf{z}|\mathbf{x})$ should be as close as possible to the true posterior $p_\theta(\mathbf{z}|\mathbf{x})$.

We use neural networks to approximate these distributions (both the encoder $q_\phi(\mathbf{z}|\mathbf{x})$ and the decoder $p_\theta(\mathbf{x}|\mathbf{z})$) because such models are well suited to capturing complex, high-dimensional, and non-linear probability distributions. Neural networks have proven highly effective at modeling intricate probability distributions in large datasets, making them ideal for parameterizing the distributions involved in variational inference.

**Training Process.** The encoder and decoder are trained by maximizing the ELBO in expectation over the data distribution $p(\mathbf{x})$.

$$\begin{aligned}
\theta^*, \phi^* &= \arg\max_{\theta,\phi} \mathbb{E}_{p(\mathrm{x})}[\mathrm{ELBO}(\mathbf{x};\theta,\phi)] \\
&= \arg\max_{\theta,\phi} \mathbb{E}_{p(\mathrm{x})}\left[\mathbb{E}_{q_\phi(\mathrm{z}|\mathrm{x})}\left[\log \frac{p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right]\right] \\
&= \arg\max_{\theta,\phi} \mathbb{E}_{p(\mathrm{x})}\left[\mathbb{E}_{q_\phi(\mathrm{z}|\mathrm{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] - \mathrm{KL}\left(q_\phi(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})\right)\right].
\end{aligned}$$

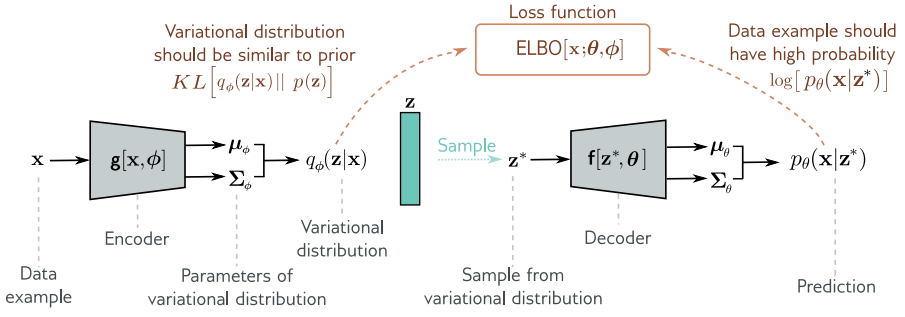



**Figure 2.1: Variational auto-encoder.**[a] The encoder maps a training example $\mathbf{x}$ to the latent space $\mathcal{Z}$ by predicting the parameters $\mu_\phi, \Sigma_\phi$ of the variational distribution $q_\phi(\mathbf{z}|\mathbf{x})$. Using the reparametrization trick (not displayed in the figure), $\mathbf{z}^*$ is sampled from this distribution and decoded back to the initial input space, using the parameters $\mu_\theta, \Sigma_\theta$ from the generating process $p_\theta(\mathbf{x}|\mathbf{z})$.

[a]Modified from Figure 17.9 of (Prince, 2023)

**Neural Networks.** The neural networks used to represent the generative and inference processes both rely on Gaussian assumptions. Indeed, a VAE (Kingma and Welling, 2022) can be fully described as a deep latent variable model where

- The prior $p(\mathbf{z})$ is prescribed and usually chosen as standard Gaussian.
- The likelihood $p_\theta(\mathbf{x}|\mathbf{z})$ is parametrized with a generative network $\text{NN}_\theta(\mathbf{z})$ (the decoder) that takes as input $\mathbf{z}$ and outputs parameters to the data distribution, e.g.

$$p_\theta(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}; \mu_\theta, \sigma_\theta^2 \mathbf{I})$$

- The approximate posterior $q_\phi(\mathbf{z}|\mathbf{x})$ is parametrized with an inference network $\text{NN}_\phi(\mathbf{z})$ (the encoder) that takes as input $\mathbf{x}$ and outputs the parameters of the variational distribution, e.g.

$$q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \mu_\phi, \sigma_\phi^2 \mathbf{I})$$

# Chapter 3

# Structured Representation Learning

Learning useful and interpretable representations is central to unsupervised and semi-supervised learning. While standard VAEs usually rely on an isotropic Gaussian prior in the latent space, this assumption is often too simplistic to capture the true structure of heterogeneous data such as cytometry.

Cytometry datasets are inherently multi-modal, consisting of complex mixtures of biologically distinct cell populations, each characterized by specific marker expression profiles. Modeling this diversity with a simple, unimodal prior fails to reflect the underlying biological organization and would lead to unstructured, entangled latent spaces.

Using *structured* latent representation aim to address this limitation by introducing more expressive priors (such as Gaussian mixtures) or by incorporating discrete latent variables. These modifications encourage the latent space to organize around distinct modes, with each mode potentially corresponding to a meaningful biological population. In the context of cytometry, this latent structure:

- promotes the discovery of distinct or rare cell subtypes,
- enables clustering directly in the latent space,
- facilitates semi-supervised learning by grounding latent modes in expert-labeled populations,
- and improves generative quality by modeling the multi-modal distribution of cell phenotypes more faithfully.

In this chapter, we introduce our model MARVIN, and review existing approaches that are closely related to ours, all aiming to extend the VAE framework toward structured latent spaces.

## 3.1 Our Approach: MARVIN

This section aims to present the methodological framework as well as the entire process of development and design of the model introduced in this master thesis.

The previous chapters and sections have provided a comprehensive theoretical background on the task to be accomplished and the appropriate methods to address it. Here, we will theoretically justify our approach, the chosen architecture, and the training process of our model.

We propose **MARVIN**: *Structured Latent Representation for Cytometry: Cell Annotation and Population Discovery*, a generative deep learning model that offers the advantage of simultaneously combining cell classification and population discovery tasks in flow cytometry data.

The objectives of our approach are complementary and can be described as

- the automatic annotation of known immune cell populations,
- the discovery of novel cellular subpopulations,
- the identification of interactions between cell sub-populations under varying experimental conditions.

MARVIN is a Mixture-based Variational Autoencoder designed to model cytometry data through a structured latent space conditioned on cell type. Each data point is assigned to a specific latent Gaussian distribution via a discrete variable, allowing the model to associate each cell population with its own latent component. This formulation enables both the classification of known populations and the discovery of novel ones within a **unified framework**. Our method builds upon the Variational Autoencoder (VAE) paradigm, extending it with a mixture-based latent structure.

### 3.1.1 Motivation: Biological Assumptions Behind the Latent Structure

To motivate our approach, we draw a parallel between the structure of our model and the biological nature of cytometry data. As discussed in Chapter 1, the relationship between marker expression and cell identity is more complex than a one-to-one mapping: cell populations often display overlapping and entangled distributions in marker space, due to biological plasticity, activation states or transitional phenotypes.

However, manual gating itself is based on a simplifying assumption: cell types can be distinguished by regions of marker expression and assigned discrete labels accordingly. In the same spirit, our model adopts the hypothesis that cell populations can be represented by distinct clusters in a latent space.

Specifically, we assume that the latent structure follows a mixture of $K$ Gaussian components, each corresponding to a distinct cell population. While this abstraction does not capture the full biological continuity between cell states, it provides a tractable and interpretable framework for representation learning, annotation, and discovery.

Such a structured latent space not only enables interpretability through the grouping of populations, as introduced at the beginning of this chapter, but also allows the injection of inductive biases into the model architecture. We believe this benefits the model (Teney et al., 2025) by providing a representation that more faithfully captures the biological structure of the data, potentially improving both generative capabilities and classification performance in semi-supervised settings. These hypotheses will be thoroughly examined and discussed in the following sections and chapters.

### 3.1.2 Generative and Inference Processes


**(A) Generative process** **(B) Inference process**

**Figure 3.1: Graphical representation of our generative and inference processes.** Grey nodes represent observed variables, white ones describe latent variables, where $c$ is discrete and $\mathbf{z}$ is normally distributed. Black nodes denote the network's parameters.

**Generative process.** A cluster $c$ is first sampled from a categorical distribution parameterized by $\pi$ and learned during the training, representing the prior $p_\pi(c)$ over clusters. As we will show in the following subsections, this learnable prior is optimized jointly with the other components of the model, using both a supervision signal and the evidence lower bound (ELBO) objective. This choice will have important downstream implications, as detailed in Chapter 4, such as enabling the segregation of subpopulations or, conversely, the annotation of unlabeled data into known cell types. It also naturally provides information on the cellular proportions within the dataset.

A sample $\mathbf{z}$ is drawn from the latent distribution $p_\beta(\mathbf{z}|c)$ associated with the selected cluster $c$, where the parameters of the distribution are output by a deep neural network. This latent sample is then passed through a decoder network $p_\theta(\mathbf{x}|\mathbf{z})$ to generate a synthetic observation $\mathbf{x}$ in the input space.

We employ deep neural networks for both the encoder and decoder components due to their ability to model complex, non-linear relationships in high-dimensional data. In the context of cytometry, where the data distribution is often non-linear and structured by intricate biological processes, deep networks provide the flexibility needed to learn expressive latent representations and accurate reconstructions.

The generative model (3.1A) can be finally described as

$$p_{\beta,\theta}(\mathbf{x}, \mathbf{z}, c) = p_\theta(\mathbf{x}|\mathbf{z})\, p_\beta(\mathbf{z}|c)\, p_\pi(c) \tag{3.1}$$

$$p_\pi(c) = \mathrm{Cat}(\pi), \quad \pi \in \Delta^K \tag{3.2}$$

$$p_\beta(\mathbf{z}|c) = \mathcal{N}(\mathbf{z} \mid \mu_\beta(\mathbf{e}_k), \mathrm{diag}(\sigma^2_\beta(\mathbf{e}_k))) \tag{3.3}$$

$$p_\theta(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x} \mid \mu_\theta(\mathbf{z}), \mathrm{diag}(\sigma^2_\theta(\mathbf{z}))), \tag{3.4}$$

where $\mu_\beta, \sigma^2_\beta, \mu_\theta$ and $\sigma^2_\theta$ are the means and variances predicted by two deep decoders.

The definition of the generative process, as depicted in Figure 3.1A, is justified by the fact that $\mathbf{x}$ and $c$ are conditionally independent given $\mathbf{z}$: the choice of $c$ influences the latent representation, which in turn determines the synthetic observation. Practically, this also enables greater generalization: the decoder $p_\theta(\mathbf{x} \mid \mathbf{z})$ does not need to specialize for each class $c$; instead, it learns to decode any $\mathbf{z}$ independently of its origin.

In other words, the latent variable $\mathbf{z}$ captures all the necessary information to generate $\mathbf{x}$, and the

role of $c$ is solely to introduce a multi-modal structure in the latent space. Once $\mathbf{z}$ is sampled, the generator no longer needs to know $c$ to produce $\mathbf{x}$.

**Inference process.** We use variational inference to approximate the true posterior $p(\mathbf{z}, c \mid \mathbf{x})$ with the variational distribution $q_{\omega,\phi}(\mathbf{z}, c \mid \mathbf{x})$. Given our inference model (Figure 3.1B), this approximate posterior is factorized as

$$q_{\omega,\phi}(\mathbf{z}, c \mid \mathbf{x}) = q_\omega(c \mid \mathbf{x}) q_\phi(\mathbf{z} \mid \mathbf{x}, c) \tag{3.5}$$

$$q_\omega(c \mid \mathbf{x}) = \mathrm{Cat}(c \mid \boldsymbol{\alpha}_\omega(\mathbf{x})), \quad \boldsymbol{\alpha}_\omega(\mathbf{x}) = \mathrm{Softmax}(\mathrm{MLP}_\omega(\mathbf{x})) \tag{3.6}$$

$$q_\phi(\mathbf{z} \mid \mathbf{x}, c) = \mathcal{N}(\mathbf{z} \mid \mu_\phi(\mathbf{x}, \mathbf{e}_k), \mathrm{diag}(\sigma^2_\phi(\mathbf{x}, \mathbf{e}_k))), \tag{3.7}$$

where $\sum_{c=1}^{K} q_\omega(c \mid \mathbf{x}) = 1$, so that our method directly predicts the categorical posterior $q_\omega(c \mid \mathbf{x})$ from a sampled data point $\mathbf{x}$ thanks to a deep neural network. The posterior of the latent variable $\mathbf{z}$ given the observed variable $\mathbf{x}$ and the discrete class latent variable $c$ is a Gaussian mixture, where $\mu_\phi(\mathbf{x}, \mathbf{e}_k)$ and $\sigma^2_\phi(\mathbf{x}, \mathbf{e}_k)$ are the mean and the variance of the variational posterior parametrized by a deep neural network, and $\mathbf{e}_k$ is a one-hot vector with a 1 at the $k_{\text{th}}$ position.

The *categorical posterior* $q_\omega(c|\mathbf{x})$ acts as the classifier head of the model: it maps a marker intensity tensor to a probability distribution over clusters. Instead of selecting the most likely class via `argmax`, a cluster (or cell type) is sampled *probabilistically* (e.g., using multinomial sampling), which allows the model to express uncertainty in ambiguous cases and encourages diverse cluster usage, thus mitigating mode collapse.

### 3.1.3 Evidence Lower Bound Objective

Since we perform *variational inference*, the objective function is the expectation of the ELBO over the data distribution

$$\mathbb{E}_{p(\mathrm{x})}\left[\mathrm{ELBO}\right] = \mathbb{E}_{p(\mathrm{x})}\mathbb{E}_{q_\phi(\mathrm{z},c|\mathrm{x})}\left[\log \frac{p_{\beta,\theta}(\mathbf{x}, \mathbf{z}, c)}{q_{\phi,\omega}(\mathbf{z}, c \mid \mathbf{x})}\right]. \tag{3.8}$$

This objective is optimized jointly with respect to all model parameters: $\phi$ and $\omega$ for the encoder networks, which define the approximate posterior $q(\mathbf{z}, c|\mathbf{x})$ and consequently the inference process, as well as $\theta$ and $\beta$ for the decoder and prior networks, which define the generative process. This expression can be factorized using Eq. 3.1 and Eq. 3.5 as

$$\mathbb{E}_{p(\mathrm{x})}\left[\mathrm{ELBO}\right] = \mathbb{E}_{p(\mathrm{x})}\mathbb{E}_{q_\phi(\mathrm{z},c|\mathrm{x})}\left[\log \frac{p_\theta(\mathbf{x}|\mathbf{z})\, p_\beta(\mathbf{z}|c)\, p_\pi(c)}{q_\omega(c \mid \mathbf{x}) q_\phi(\mathbf{z} \mid \mathbf{x}, c)}\right] \tag{3.9}$$

$$= \mathbb{E}_{p(\mathrm{x})}\mathbb{E}_{q_\phi(\mathrm{z},c|\mathrm{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] - \underbrace{\mathbb{E}_{p(\mathrm{x})}\left[\mathrm{KL}\left(q_\omega(c \mid \mathbf{x})\, q_\phi(\mathbf{z} \mid \mathbf{x}, c) \| p_\beta(\mathbf{z}|c)\, p_\pi(c)\right)\right]}_{\star}. \tag{3.10}$$

The second term can be rewritten as:

$$\star = \mathbb{E}_{p(\mathrm{x})}\mathbb{E}_{q_\omega(c|\mathrm{x})}\mathbb{E}_{q_\phi(\mathrm{z}|\mathrm{x},c)}\left[\log \frac{q_\omega(c \mid \mathbf{x})\, q_\phi(\mathbf{z} \mid \mathbf{x}, c)}{p_\beta(\mathbf{z}|c)\, p_\pi(c)}\right] \tag{3.11}$$

$$= \mathbb{E}_{p(\mathrm{x})}\mathbb{E}_{q_\omega(c|\mathrm{x})}\left[\log \frac{q_\omega(c \mid \mathbf{x})}{p_\pi(c)} + \mathbb{E}_{q_\phi(\mathrm{z}|\mathrm{x},c)}\left[\log \frac{q_\phi(\mathbf{z} \mid \mathbf{x}, c)}{p_\beta(\mathbf{z}|c)}\right]\right] \tag{3.12}$$

$$= \mathbb{E}_{p(\mathrm{x})}\left[\mathrm{KL}(q_\omega(c \mid \mathbf{x}) \| p_\pi(c)) + \mathbb{E}_{q_\omega(c|\mathrm{x})}\left[\mathrm{KL}(q_\phi(\mathbf{z} \mid \mathbf{x}, c) \| p_\beta(\mathbf{z}|c))\right]\right]. \tag{3.13}$$

**our Evidence Lower Bound Objective**

$$\mathbb{E}_{p(\mathrm{x})}\mathbb{E}_{q_\phi(\mathrm{z},c|\mathrm{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] - \mathbb{E}_{p(\mathrm{x})}\left[\mathrm{KL}(q_\omega(c \mid \mathbf{x}) \| p_\pi(c)) + \mathbb{E}_{q_\omega(c|\mathrm{x})}\left[\mathrm{KL}(q_\phi(\mathbf{z} \mid \mathbf{x}, c) \| p_\beta(\mathbf{z}|c))\right]\right] \tag{3.14}$$

### 3.1.4 Interpretation

The terms of the objective function can be interpreted as follows.

**First term.** $\mathbb{E}_{p(\mathrm{x})}\mathbb{E}_{q_\phi(\mathrm{z},c|\mathrm{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right]$ is the *reconstruction loss*, i.e. the likelihood of a sampled data point $\mathbf{x}$ with respect to the latent variable $\mathbf{z}$. Using Monte Carlo approximation, it can be estimated by

$$\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{K} q_\omega(c \mid \mathbf{x}_i)\mathbb{E}_{\mathrm{z}\sim q_\phi(\mathrm{z}|c,\mathrm{x})}\left[\log p_\theta(\mathbf{x}_i \mid \mathbf{z})\right],$$

where $N$ is the batch size and $K$ the number of clusters. Knowing that $p_\theta(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x} \mid \mu_\theta(\mathbf{z}), \mathrm{diag}(\sigma_\theta^2(\mathbf{z})))$, the log likelihood has the closed form

$$\log p_\theta(\mathbf{x} \mid \mathbf{z}) = \log\left(\frac{1}{\sigma_\theta(\mathbf{z})\sqrt{2\pi}}\exp\left(-\frac{1}{2}\left(\frac{\mathbf{x}-\mu_\theta(\mathbf{z})}{\sigma_\theta(\mathbf{z})}\right)^2\right)\right),$$

meaning that we need to minimize

$$\frac{1}{2}\left(\log(\sigma_\theta(\mathbf{z}) + \frac{(\mathbf{x}-\mu_\theta(\mathbf{z}))^2}{\sigma_\theta^2(\mathbf{z})}\right).$$

Altogether, the reconstruction loss comes down to minimize

$$\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{K} q_\omega(c \mid \mathbf{x}_i)\mathbb{E}_{\mathrm{z}\sim q_\phi(\mathrm{z}|c,\mathrm{x})}\left[\frac{1}{2}\left(\log(\sigma_\theta) + \frac{(\mathbf{x}-\mu_\theta)^2}{\sigma_\theta^2}\right)\right]. \tag{3.15}$$

In practice, the expectation $\mathbb{E}_{\mathrm{z}\sim q_\phi(\mathrm{z}|c,\mathrm{x})}[\cdot]$ is approximated using Monte Carlo sampling with the *reparameterization trick*. Specifically, for each data point $\mathbf{x}$ and cluster $c$, a latent variable $\mathbf{z}$ is sampled from the variational distribution $q_\phi(\mathbf{z} \mid c, \mathbf{x})$, which is typically modeled as a Gaussian distribution whose parameters (mean and variance) are given by the encoder network. This sampled $\mathbf{z}$ is then used to compute $\log p_\theta(\mathbf{x} \mid \mathbf{z})$ via the decoder.

**Second term.** $-\mathbb{E}_{p(\mathrm{x})}\left[\mathrm{KL}(q_\omega(c \mid \mathbf{x}) \| p_\pi(c))\right]$ is the term forcing the categorical posterior $q_\omega(c \mid \mathbf{x})$ to be close to the learnable prior $p_\pi(c)$ because maximizing -KL returns to minimizing it, so we want it to be 0. With MC estimation, it evaluates to

$$\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{K} q_\omega(c \mid \mathbf{x}_i)\left(\log(q_\omega(c \mid \mathbf{x}_i)) - \log(p_\pi(c))\right). \tag{3.16}$$

**Third term.** $-\mathbb{E}_{p(\mathbf{x})}\mathbb{E}_{q_\omega(c|\mathbf{x})}\left[\mathrm{KL}(q_\phi(\mathbf{z} \mid \mathbf{x}, c)\|p_\beta(\mathbf{z}|c))\right]$ is the classical KL term forcing the learned latent representation $q_\phi(\mathbf{z} \mid \mathbf{x}, c)$ to be close to the prior we postulated about the form of this same latent space $p_\beta(\mathbf{z}|c)$. The MC estimation writes

$$\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{K} q_\omega(c \mid \mathbf{x}_i)\,\mathrm{KL}(q_\phi(\mathbf{z} \mid \mathbf{x}_i, c)\|p_\beta(\mathbf{z}|c)). \tag{3.17}$$

Since both $q_\phi(\mathbf{z} \mid \mathbf{x}_i, c)$ and $p_\beta(\mathbf{z}|c)$ are Gaussian distributions, the KL has the closed form (which is derived in Appendix A)

$$\mathrm{KL}(q_\phi(\mathbf{z} \mid \mathbf{x}, c)\|p_\beta(\mathbf{z}|c)) = \frac{1}{2}\log\left(\frac{\sigma_\beta^2(\mathbf{e}_k)}{\sigma_\phi^2(\mathbf{x}, \mathbf{e}_k)}\right) + \frac{1}{2}\cdot\frac{\sigma_\phi^2(\mathbf{x}, \mathbf{e}_k) + \left(\mu_\phi(\mathbf{x}, \mathbf{e}_k) - \mu_\beta(\mathbf{e}_k)\right)^2}{\sigma_\beta^2(\mathbf{e}_k)} - \frac{1}{2}. \tag{3.18}$$

The third term is finally evaluated as follows

$$\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{K} q_\omega(c \mid \mathbf{x}_i)\left(\frac{1}{2}\log\left(\frac{\sigma_\beta^2(\mathbf{e}_k)}{\sigma_\phi^2(\mathbf{x}_i, \mathbf{e}_k)}\right) + \frac{1}{2}\cdot\frac{\sigma_\phi^2(\mathbf{x}_i, \mathbf{e}_k) + \left(\mu_\phi(\mathbf{x}_i, \mathbf{e}_k) - \mu_\beta(\mathbf{e}_k)\right)^2}{\sigma_\beta^2(\mathbf{e}_k)} - \frac{1}{2}\right). \tag{3.19}$$

### 3.1.5 Semi-supervised Training

Our data are in the form of a training set $\{\mathbf{x}_i, c_i\}$, where $\mathbf{x}$ represents the signal intensity of the various markers of the cell labeled by $c_i$. However, these labels $c_i$ are not always defined: the clinician may not have annotated all cells during the manual gating process.

Since it is essential for our model to automatically annotate cells, we needed to define a semi-supervised framework, allowing us to handle the data not analyzed by the expert in the training set. This can be easily achieved by adding a supervision signal to the head $q_\omega(c \mid \mathbf{x})$.

Indeed, since our model learns the categorical posterior $q_\omega(c|\mathbf{x})$, we were able to inject a supervision signal into this component. This introduces an additional structuring cue via the provided labels and allows $q_\omega(c|\mathbf{x})$ to act as a classifier, rather than simply a soft assignment head as it is often seen in related works (Section 3.2).

Here, we introduce a supervised loss term added to the ELBO, namely the classical cross-entropy loss, which can be computed, using Monte Carlo estimation, as

$$\mathcal{L}_{\text{supervised}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}\delta_{k,c_i^{\text{true}}}\log q_\omega(k \mid \mathbf{x}_i) \tag{3.20}$$

$$= -\frac{1}{N}\sum_{i=1}^{N}\log q_\omega(c_i^{\text{true}} \mid \mathbf{x}_i), \tag{3.21}$$

where $N$ is the batch size, and $\delta_{k,c_i^{\text{true}}}$ is the one-hot indicator equal to 1 if cluster $k$ corresponds to the true label $c_i^{\text{true}}$ and 0 otherwise. This formulation naturally arises from one-hot encoding the ground truth labels. This supervised loss penalizes the model whenever the predicted categorical distribution $q_\omega(c \mid \mathbf{x})$ assigns low probability to the true class, thereby guiding the model towards more accurate cluster assignments.

By encouraging the model to correctly classify labeled examples, this supervision signal guides the encoder to structure the latent space in a way that better reflects the known class boundaries.

As a result, it promotes a clearer separation between clusters in the latent space (Section 4.2.3), which also benefits the unsupervised data by pulling them toward more discriminative regions.

Ultimately, our loss is a combination of a supervised and an unsupervised loss, $\mathcal{L} = \mathcal{L}_{\text{unsupervised}} + \mathcal{L}_{\text{supervised}}$, that is

$$\mathcal{L} = \mathcal{L}_{\text{supervised}} - \text{ELBO}\,.$$

Thanks to this strategy, our model is inherently capable of handling missing labels, which is a major advantage in cytometry. It can be trained on partially labeled datasets where a clinician may not have annotated all cells during the manual gating process. Alternatively, it can be used in scenarios where only a single patient's cell populations have been gated, while the remaining patients are entirely unlabeled, leading to a significant time-saving benefit for the practitioner.

Unlabeled cells are thus processed by the model solely through the unsupervised loss term (ELBO), and not through the supervised loss. Nevertheless, they are still mapped to the most compatible cluster in the latent space based on their intrinsic characteristics, enabling subsequent automatic annotation of previously unlabeled cells.

We will evaluate the model's performance under varying levels of supervision during training, and discuss the influence of the unsupervised loss in the classification in Section 4.2.3.

### 3.1.6 Network Architecture

To implement our model, we designed a set of neural network architectures tailored to the different components of the variational framework. These include the encoder networks $q_\omega(c|\mathbf{x})$ and $q_\phi(\mathbf{z}|\mathbf{x}, c)$ that approximate the posterior distributions over latent variables, and the decoder networks $p_\beta(\mathbf{z}|c)$ and $p_\theta(\mathbf{x}|\mathbf{z})$ that (re)construct the input data or generate auxiliary outputs. In this section, we provide a short description of the neural networks used for each probabilistic head of the model.

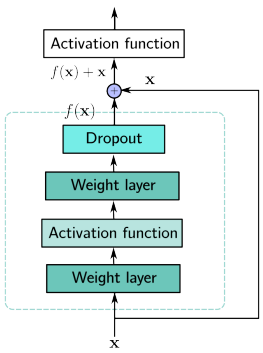



**Figure 3.2: Residual Block.** The input of each residual block (dashed frame) $\mathbf{x}$ is added back to its output $f(\mathbf{x})$, which is passed via a parallel computational path called a *residual* or *skip* connection (He et al., 2015).

As previously mentioned, our inference and generative processes are parameterized by deep neural networks, all of which are Residual Multi-Layer Perceptrons (MLP composed of residual blocks, Figure 3.2). The concept of residual learning was first introduced by He et al., 2015, using skip connections to shortcut layers and thus allow gradients to flow without vanishing. This greatly simplifies the loss landscape and increases model stability. To prevent overfitting and improve the

generalization performance of the model, dropout is applied at the end of each residual block. Dropout works by clamping a random subset of hidden units to zero at each iteration of stochastic gradient descent (Srivastava et al., 2014). This regularization technique reduces the network's reliance on any individual hidden unit, thereby limiting the impact of their presence or absence and encouraging the model to learn more robust, distributed representations (Prince, 2023).

Figure 3.3 shows each head of our overall architecture, with a representation of the inputs and outputs for each one. The model was implemented using PyTorch (Paszke et al., 2019).

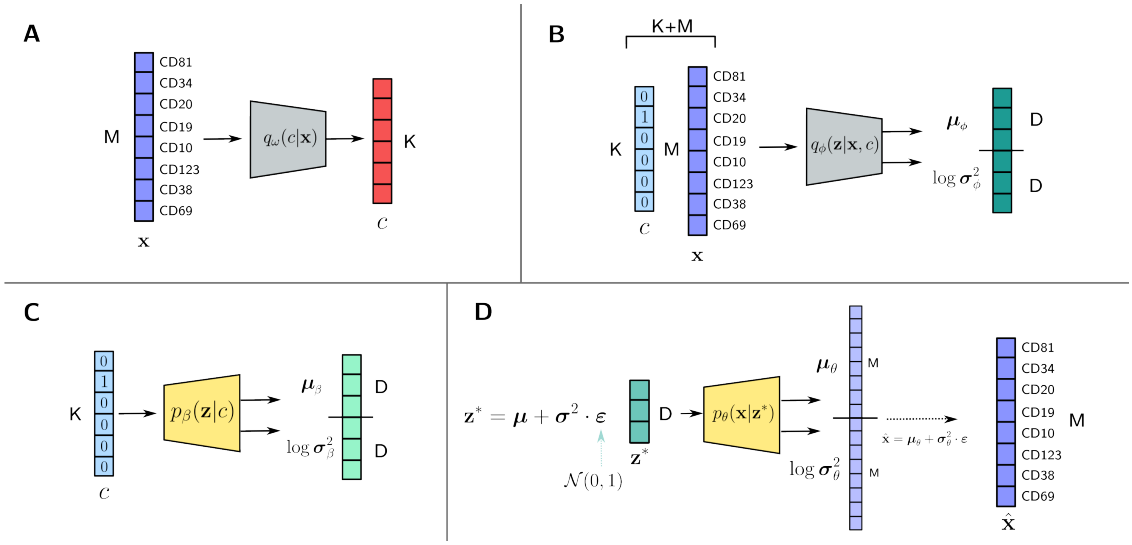



**Figure 3.3: Building blocks of our network.** $M$ is the number of markers, $D$ the latent space dimension ($\mathbf{z}$) and $K$ the number of clusters (or cell types).
**A.** Categorical posterior $q_\omega(c|\mathbf{x})$ (classification) head. Takes as input $\mathbf{x}$, the per-marker signal for the cell, and outputs logits corresponding to each cell class. **B.** Variational posterior $q_\phi(\mathbf{z}|\mathbf{x}, c)$ head. Takes as input the one-hot encoding of the class $c$ concatenated with $\mathbf{x}$, and outputs the parameters of the variational distribution: $\mu_\phi$ and $\log \sigma^2_\phi$. **C.** Prior over the latent space $p_\beta(\mathbf{z}|c)$ head. Takes as input the one hot encoding of a class, and outputs the parameters of the prior distribution for this specific class, that is $\mu_\beta$ and $\log \sigma^2_\beta$. **D.** Likelihood $p_\theta(\mathbf{x}|\mathbf{z})$ head. This module is part of the reconstruction process: a latent vector $\mathbf{z}^*$ is sampled from the variational posterior $q_\phi(\mathbf{z}|\mathbf{x}, c)$ using the *reparameterization trick*, and then passed through the decoder to produce the parameters $\mu_\theta$ and $\sigma^2_\theta$ of the likelihood distribution, mapping the latent representation back to the input space. In generation mode, $\mathbf{z}^*$ is instead sampled from the prior $p_\beta(\mathbf{z}|c)$.
All heads are implemented as residual MLPs.

### 3.1.7 Training Process

This subsection covers all the specific strategies and configurations used during the training of our deep neural networks.

Figure 3.4 provides an overview of the training strategy.

**Marginalizing Over Cluster Assignments in the ELBO.** As discussed in Section 3.1.4, the ELBO is evaluated in three components, namely Equations 3.15, 3.16, and 3.19.

As shown in these equations, our training strategy does not rely on hard cluster assignments where

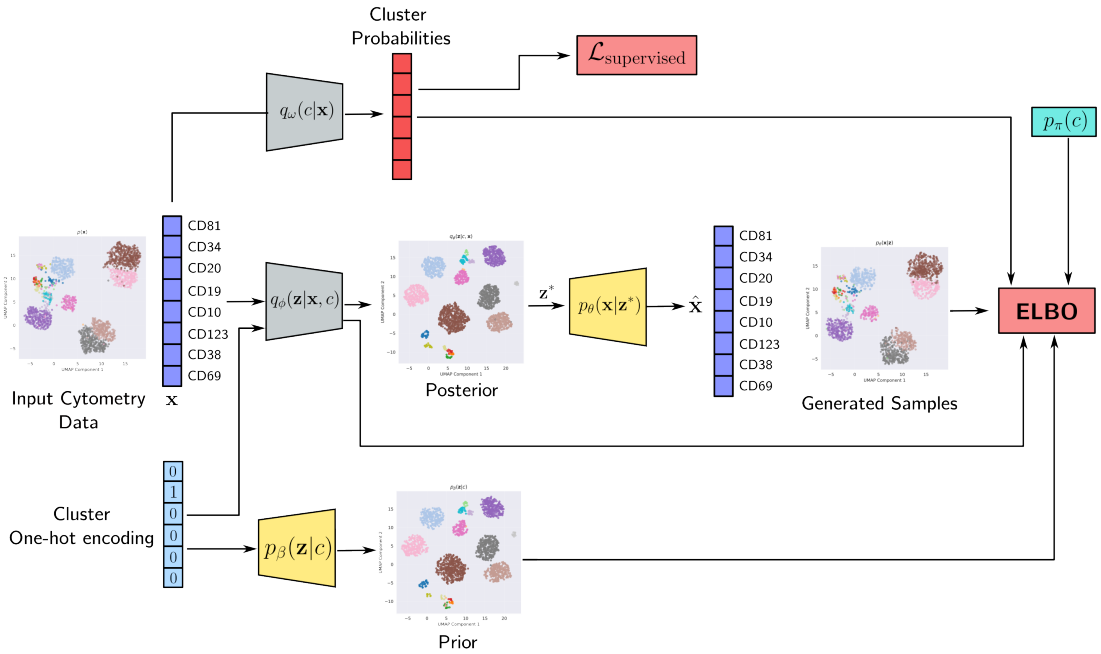



**Figure 3.4: Flow chart of the training process.** Given a set of cells and their per-marker expression $\mathbf{x}$, the posterior categorical distribution $q_\omega(c|\mathbf{x})$ is first computed. Iterating over all possible clusters, a one-hot encoding of the current cluster is generated and used to compute both the variational distribution $q_\omega(\mathbf{z}|\mathbf{x}, c)$ and the prior $p_\beta(\mathbf{z}|c)$. A latent sample is drawn from the variational distribution and used to compute the conditional likelihood $p_\theta(\mathbf{x}|\mathbf{z})$. The ELBO is then evaluated according to Equation 3.14, by marginalizing over all possible clusters. If ground-truth labels are provided for $\mathbf{x}$, the distribution $q_\omega(c|\mathbf{x})$ is also used to compute the supervised loss as described in Equation 3.21.

one would have sampled a single cluster $c \sim q_\omega(c|\mathbf{x})$ for each cell in the current batch. Instead, for the ELBO, we compute an expectation over all possible clusters, weighted proportionally to the categorical
posterior $q_\omega(c = k \mid \mathbf{x})$.

This approach allows us to marginalize over all plausible cluster assignments, resulting in a fully differentiable and low-variance ELBO. It also ensures that uncertainty in the cluster assignment is properly taken into account through a soft expectation.

In our application domain of cytometry, datasets are often *highly imbalanced* (see Table 4.3). While other models rely on techniques such as oversampling or emphasizing rare populations every few epochs (Blampey et al., 2023) to avoid mode collapse, our model requires no such tricks. The marginalization over all clusters ensures that every cluster receives gradient updates during training, which helps prevent mode collapse while naturally incorporating uncertainty.

**Semi-supervised Framework.** Artificially or not, the model can operate within a semi-supervised framework, meaning it is capable of handling missing labels (i.e., cells not annotated by experts), while also allowing manual control over the degree of supervision incorporated into the model.

In other words, using a masking mechanism (See Section 4.2.3), one can choose to hide a pre-defined percentage of labels in order to make the model less dependent on the supervision signal. The cells whose labels are masked will therefore not contribute to $\mathcal{L}_{\text{supervised}}$ and will instead only

be used in the ELBO.

Artificially retaining only a subset of the labels can help reduce the bias introduced by the practitioner's subjective gating, as the model would not rely *entirely* on the provided labels. Hiding certain cell populations is also useful to demonstrate the model's ability to recover them later on, providing concrete evidence of its capacity to discover subpopulations whose labels are never seen in the training set (see Section 4.3 and Subsection 4.2.3).

**Optimizer and Learning rate.** The model was trained using the AdamW optimizer (Kingma and Ba, 2017, with decoupled weight decay regularization Loshchilov and Hutter, 2019). We set $\beta_1 = 0.9$, while treating $\beta_2$ as a tunable hyperparameter. This decision was motivated by the observation that, when using large batch sizes (which is common when the dataset contains millions of cells), the loss tended to spike periodically. This phenomenon was also reported by Zhai et al., 2023, who found that reducing $\beta_2$, which controls the smoothing of the second moment estimate in Adam, can stabilize training. The authors further noted that popular architectures such as ViT Radford et al., 2021 also benefit from a lower $\beta_2$ (0.98 instead of 0.999). In our experiments, we observed improved training stability and the disappearance of these spikes when decreasing $\beta_2$ for larger batch sizes (e.g., 1024 and 2048), with no significant effect for smaller batch sizes.

For the learning rate, we use a warm-up phase in which the learning rate is linearly increased from a small base value ($10^{-5}$) to the target learning rate (set to the optimal value found in the hyperparameter tuning phase).

After this warm-up, we apply a learning rate scheduler (`StepLR` from PyTorch), which halves the learning rate every 10 epochs.

**Clamping of the Log-variance.** We chose to clamp $\log \sigma_\theta^2$ between -6 and 3, which corresponds to keeping $\sigma_\theta^2$ within approximately $[0.002, 20]$. This design choice offers several advantages:

- It prevents the model to be overconfident: it cannot learn a reconstruction variance too close to zero, which, besides making training unstable (due to near-zero division in the reconstruction loss, Equation 3.15), would lead to near-deterministic predictions and minimize the reconstruction loss through overconfidence, even when the reconstruction $\mu_\theta$ is potentially poor.
- By placing an upper bound on the variance, we prevent overly tolerant reconstructions: a model that constructs an observation while specifying an excessively large variance would result in unrealistic error estimation.

In practice, this trick leads to much more stable training (no NaNs caused by overconfident models or excessively large reconstruction gradients, nor near-zero divisions), and simply better overall performance.

## 3.2 Related Work

Our model was developed independently to address the specific challenges of cytometry data. After its design, we identified several existing extensions of the standard VAE that share similar goals, particularly regarding the introduction of latent structures suitable for clustering. In this section, we briefly review the most relevant of these models for comparison.

### 3.2.1 GMVAE

The Gaussian Mixture Variational Autoencoder or GMVAE (Dilokthanakul et al., 2017), extends the classical VAE by explicitly introducing a discrete latent variable $c$ that represents a cluster assignment of each data point. The goal is to encourage the emergence of distinct subgroups in the latent space. To achieve this, GMVAE replaces the standard unimodal Gaussian prior $p(\mathbf{z})$


**(A) Generative process** **(B) Inference process**

**Figure 3.5: Graphical models for the Gaussian Mixture Variational Autoencoder**.

with a more expressive prior: a mixture of Gaussians. The generative model involves three latent variables: a class variable $\mathbf{c}$, a continuous auxiliary variable $\mathbf{w}$, and the main latent variable $\mathbf{z}$. The full generative process is defined as

$$
\begin{aligned}
p(\mathbf{x}, \mathbf{z}, \mathbf{c}, \mathbf{x}) &= p_\theta(\mathbf{x} \mid \mathbf{z}) p_\beta(\mathbf{z} \mid \mathbf{c}, \mathbf{w}) p(\mathbf{w}) p(\mathbf{c}) \\
p(\mathbf{w}) &= \mathcal{N}(0, \mathbf{I}) \\
p(\mathbf{c}) &= \text{Mult}(\boldsymbol{\pi}) \\
p_\beta(\mathbf{z} \mid \mathbf{c}, \mathbf{w}) &= \prod_{k=1}^{K} \mathcal{N}\left(\boldsymbol{\mu}_{c_k}(\mathbf{w}; \beta), \text{diag}\left(\boldsymbol{\sigma}^2_{c_k}(\mathbf{w}; \beta)\right)\right)^{c_k} \\
p_\theta(\mathbf{x} \mid \mathbf{z}) &= \mathcal{N}\left(\boldsymbol{\mu}(\mathbf{z}; \theta), \text{diag}\left(\boldsymbol{\sigma}^2(\mathbf{z}; \theta)\right)\right) \text{ or } \text{Ber}(\boldsymbol{\mu}(\mathbf{z}; \theta)).
\end{aligned}
$$

As pointed out by Shu, 2016, marginalizing out $\mathbf{w}$ from $p(\mathbf{z}|\mathbf{c}, \mathbf{w})$ yields a mixture of complex, non-Gaussian distributions due to the nonlinear dependencies in $\mu_{c_k}$. This deviates from the intended interpretation of a true Gaussian mixture prior, making the term "GMVAE" somewhat misleading.

Their inference process can be described as

$$
\begin{aligned}
q(\mathbf{z}, \mathbf{w}, \mathbf{c} \mid \mathbf{x}) &= q_\phi(\mathbf{z} \mid \mathbf{x})\, q_\xi(\mathbf{w} \mid \mathbf{x})\, p_\beta(\mathbf{c} \mid \mathbf{z}, \mathbf{w}) \\
q_\phi(\mathbf{z} \mid \mathbf{x}) &= \mathcal{N}\left(\boldsymbol{\mu}_z(\mathbf{x}; \phi), \text{diag}\left(\boldsymbol{\sigma}^2_z(\mathbf{x}; \phi)\right)\right) \\
q_\xi(\mathbf{w} \mid \mathbf{x}) &= \mathcal{N}\left(\boldsymbol{\mu}_w(\mathbf{x}; \xi), \text{diag}\left(\boldsymbol{\sigma}^2_w(\mathbf{x}; \xi)\right)\right) \\
p_\beta(\mathbf{c} \mid \mathbf{z}, \mathbf{w}) &\propto p(c)\, p_\beta(\mathbf{z} \mid \mathbf{c}, \mathbf{w})
\end{aligned}
$$

### 3.2.2 Variational Clustering (VC)

Another more recent approach was introduced by Prasad et al., 2020. The idea remains to use a fixed prior consisting of a mixture of Gaussians, but to have a categorical posterior $q(c \mid \mathbf{x})$ that learns the cluster assignment probabilities *during training* instead of expressing it as an approximation using Bayes' rule.


**(A) Generative process** **(B) Inference process**

**Figure 3.6: Graphical models for Variational Clustering (VC).**

The generative model considered in VC is

$$\begin{aligned} p(\mathbf{x}, \mathbf{z}, c) &= p_\theta(\mathbf{x}|\mathbf{z}) p_\beta(\mathbf{z}|c) p(c) \\ p(c) &= \text{Cat}(c \mid \boldsymbol{\pi}) \\ p_\beta(\mathbf{z} \mid c) &= \mathcal{N}\left(\mathbf{z} \mid \boldsymbol{\mu}_\beta(c), \boldsymbol{\sigma}_\beta(c)^2 \mathbf{I}\right) \\ p_\theta(\mathbf{x} \mid \mathbf{z}) &= \text{Ber}\left(\mathbf{x} \mid \boldsymbol{\mu}_\theta\right) \quad \text{or} \quad \mathcal{N}\left(\mathbf{x} \mid \boldsymbol{\mu}_\theta, \boldsymbol{\sigma}_\theta^2 \mathbf{I}\right). \end{aligned}$$

And for the inference model, as in the modifications of GMVAE proposed by Rui Shu, 2016 in his blog on Gaussian Mixture VAEs, it writes

$$\begin{aligned} q(\mathbf{z}, c \mid \mathbf{x}) &= q_\omega(c \mid \mathbf{x}) q_\phi(\mathbf{z} \mid \mathbf{x}, c) \\ q_\omega(c \mid \mathbf{x}) &= \text{Cat}(c \mid \boldsymbol{\alpha}_\omega(\mathbf{x})), \quad \boldsymbol{\alpha}_\omega(\mathbf{x}) = \text{Softmax}(\text{MLP}_\omega(\mathbf{x})) \\ q_\phi(\mathbf{z} \mid \mathbf{x}, c) &= \mathcal{N}(\mathbf{z} \mid \mu_\phi(\mathbf{e}_k, \mathbf{x}), \text{diag}(\sigma_\phi^2(\mathbf{e}_k, \mathbf{x}))), \end{aligned}$$

where $\mathbf{e}_k$ is a one-hot vector with a 1 at the $k_{\text{th}}$ position.

As we discovered *a posteriori*, these generative and inference models are structurally identical to ours. However, we differ in our strategy to learn the prior $p_\pi(c)$ and in our use of a supervision signal to guide the posterior $q_\omega(c \mid \mathbf{x})$.

They express their Evidence Lower Bound Objective as

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

so the categorical posterior is learned directly thanks to the expected entropy term (second one). The maximization of such an entropy term is thought to be anti-clustering by nature (according to the authors, and as later supported by us, this highlights the advantage of using a supervision signal to guide $q_\omega(c \mid \mathbf{x})$ and of learning $p_\pi(c)$), since this regularization forces the information stored in $q_\omega(c \mid \mathbf{x})$ to be distributed among the clusters. This is however mitigated by the reconstruction term and has the advantage of preventing mode collapsing.

### 3.2.3 Semi-supervised Latent Structure: the M2 Model

While the models described above aim to structure the latent space through a categorical variable in a purely unsupervised setting, another line of work has explored how incorporating label information can further guide the structure of the latent space. This leads to semi-supervised approaches where the latent space is shaped not only by reconstruction, but also by class-level supervision.

A key contribution in this direction is the M2 model proposed by Kingma et al., 2014, which extends the VAE framework to support semi-supervised learning by introducing a discrete latent variable $c$ representing the class label, alongside a continuous latent variable $\mathbf{z}$ capturing intra-class variations. The generative model is defined as:

$$p(c) = \text{Categorical}(c \mid \boldsymbol{\pi}); \quad p(\mathbf{z}) = \mathcal{N}(\mathbf{z} \mid \mathbf{0}, \mathbf{I}); \quad p_\theta(\mathbf{x} \mid c, \mathbf{z}) = f(\mathbf{x}; c, \mathbf{z}, \boldsymbol{\theta}),$$

so that the likelihood now depends explicitly on both $c$ and $\mathbf{z}$.

This model, when combined with a standard VAE architecture (referred to as M1), results in a two-level generative process:

$$p_\theta(\mathbf{x}, c, \mathbf{z}_1, \mathbf{z}_2) = p(c)\, p(\mathbf{z}_2)\, p_\theta(\mathbf{z}_1 \mid c, \mathbf{z}_2)\, p_\theta(\mathbf{x} \mid \mathbf{z}_1).$$

Here, the generative network is composed of two neural networks responsible for decoding from $\mathbf{z}_2$ to $\mathbf{z}_1$, and from $\mathbf{z}_1$ to $\mathbf{x}$.

The inference model approximates the posterior for both latent variables via:

$$\begin{aligned} q_{\omega,\phi}(\mathbf{z}, c \mid \mathbf{x}) &= q_\omega(c \mid \mathbf{x}) q_\phi(\mathbf{z} \mid \mathbf{x}, c) \\ q_\omega(c \mid \mathbf{x}) &= \text{Cat}(c \mid \boldsymbol{\alpha}_\omega(\mathbf{x})), \quad \boldsymbol{\alpha}_\omega(\mathbf{x}) = \text{Softmax}(\text{MLP}_\omega(\mathbf{x})) \\ q_\phi(\mathbf{z} \mid \mathbf{x}, c) &= \mathcal{N}(\mathbf{z} \mid \mu_\phi(\mathbf{x}, \mathbf{e}_k), \text{diag}(\sigma^2_\phi(\mathbf{x}, \mathbf{e}_k))), \end{aligned}$$

In this setting, the model first infers the class assignment $c$ given the input $\mathbf{x}$, then estimates the distribution over $\mathbf{z}$ conditioned on both $c$ and $\mathbf{x}$. As a result, the variational distribution $q_\phi(\mathbf{z} \mid c, \mathbf{x})$ becomes a mixture of Gaussians, capable of predicting missing labels obtained from the inferred multinomial posterior distribution $q_\phi(c \mid \mathbf{x})$.

It is important to note, however, that the M2 model alone still assumes a standard Gaussian prior over $\mathbf{z}$, i.e. $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$, meaning that the latent structure is induced only through the inference model rather than the prior itself.

# Chapter 4

# Experiments and Results

This chapter presents the results related to the different objectives defined by our approach.

#### Structure of the Chapter

The first section, **4.1.1 Evaluation Methods**, presents the metrics and hyperparameter tuning results that guided the optimization of our architecture.

The second section, **4.2 Automatic Annotation of Cells**, evaluates the annotation performance of our model. We compare MARVIN against Scyan and study the impact of the supervision on the model's performance.

The third section, **4.3 Discovery of Novel Cellular Subpopulations**, highlights the ability of the model to isolate rare, phenotypically distinct cell populations without explicit supervision.

The fourth section, **4.4 Patients' Cellular Dynamics Across Experimental States**, describes how the model can be used to perform a comprehensive, patient-specific analysis of immune populations across two experimental conditions: before and after stimulation with an allergen.

The fifth section, **4.5 Anomaly Detection**, explores the capacity of the model to detect abnormal or deviant cellular states assessing their reconstruction.

The final section, **4.6 Comparison with Classical VAE**, contrasts the performance of our structured latent model with a standard VAE, emphasizing the benefits of explicitly structuring the latent space with $K$ different components.

## 4.1 Evaluation Methods

This section focuses on evaluating the performance of our architecture. We first detail the methodology used to assess the annotation performances of our model. We then present the results of hyperparameter tuning, which was conducted to ensure optimal training conditions for our proposed approach.

### 4.1.1 Metrics

During training, the loss is evaluated on the training set $\mathcal{T}_{\text{train}}$ to assess whether the model is learning correctly, stably, and without stagnation. For hyperparameter tuning, the loss is measured on a separate validation set $\mathcal{T}_{\text{val}}$. Finally, after training, a final evaluation is performed on a test set $\mathcal{T}_{\text{test}}$ to assess both the final performance of the model and its generalization capabilities.

In addition to the overall loss $\mathcal{L}$, three supplementary metrics are computed on $\mathcal{T}_{\text{val}}$ and $\mathcal{T}_{\text{test}}$. These metrics aim to evaluate the cell annotation performance of our model.

Since manual gating is a sequential refinement process of cell populations, the resulting datasets are often highly imbalanced. Moreover, in certain applications such as the detection of residual disease in cancer patients, the most important populations are the minority ones. It is therefore crucial to use metrics that take into account the imbalance of the dataset.

**Accuracy.** It is a global metric that does not take into account class imbalance, as a model can achieve high accuracy while completely failing to detect a minority class.

$$\text{Accuracy} = \frac{\text{correct predictions}}{\text{total predictions}} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{4.1}$$

**$F_1$-score.** In the context of imbalanced data, the primary objective is often to improve the true positive rate (recall) for the minority class. However, this improvement can come at the cost of an increased number of false positives, thereby lowering precision. To balance these two aspects, the $F_1$-score, defined as the harmonic mean of precision and recall, is commonly used as a more informative metric than accuracy in such settings (Buckland and Gey, 1994).

$$F_1\text{-score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{4.2}$$

In the case of multiclass problems (as ours), the $F_1$-score can be calculated independently for each class and then averaged without any weights to assess that all classes are equally important, even if they are imbalanced.

$$F_1^{\text{macro}} = \frac{1}{K} \sum_{i=1}^{K} F_{1_i} \tag{4.3}$$

**Precision and Recall**

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

**Balanced Accuracy.** This metric computes the average recall for each class.

$$\text{Balanced Accuracy} = \frac{1}{K} \sum_{i=1}^{K} \frac{\text{TP}_i}{\text{TP}_i + \text{FN}_i} \tag{4.4}$$

### 4.1.2 Hyperparameter Tuning

To achieve optimal model performance, we conducted a tuning of selected hyperparameters. This process involved performing a grid search, where various combinations of hyperparameters were systematically explored (see Table 4.1). The model was trained for 30 epochs per configuration. Performance was evaluated on the validation set, and we selected the parameters that yielded the highest balanced accuracy, while also maintaining strong F1 scores and a low validation loss.

| **Hyperparameter** | **Explored values** |
|---|---|
| Learning rate ($\eta$) | $\{10^{-4}, 10^{-3}, 2 \cdot 10^{-3}, 5 \cdot 10^{-3}\}$ |
| Dropout ($p_{\text{dropout}}$) | $\{0.0, 0.1, 0.2, 0.3\}$ |
| Latent space dimension ($D$) | $\{16, 32, 64, 128\}$ |
| Batch size ($N$) | $\{32, 64, 128, 512, 1024\}$ |
| Adam's momentum ($\beta_2$) | $\{0.98, 0.999\}$ |
| Number of residual blocks ($n_{\text{blocks}}$) | $\{1, 2, 4\}$ |
| Upscaling factor for hidden layers (factor) | $\{1, 2, 4, 8\}$ |

**Table 4.1: Hyperparameters explored during grid search.**

A thorough analysis was conducted across all datasets used in this study. To ensure robustness and reproducibility across different contexts, we propose a standard set of recommended hyperparameters:

The learning rate is consistently set to $2 \cdot 10^{-3}$, with an initial value of $10^{-5}$ at the beginning of the warm-up phase. As previously described, the learning rate is halved every 10 epochs.

For regularization, we recommend a dropout probability of 30%, i.e., $p_{\text{dropout}} = 0.3$.

The choice of batch size $N$ and latent space dimension $D$ depends on the size and complexity of the dataset. For smaller datasets (fewer than 500,000 cells), we recommend using a batch size of 64, while larger datasets benefit from a higher batch size of 1024 to reduce training time without compromising too much performance. Regarding the latent space dimension $D$, if the number of cell populations is small (which is typically associated with fewer markers) $D$ can be kept around 16, close to the number of expected clusters. For datasets with approximately 20 or more cell populations, we recommend increasing $D$ to 64. In practice, model performance remains stable across a range of values, indicating robustness to this choice.

Finally, for the AdamW optimizer, as discussed in the previous section, we suggest reducing the momentum term $\beta_2$ to 0.98 when using large batch sizes (e.g., 1024), instead of the default value of 0.999.

As for architectural details, we recommend using 2 residual blocks per encoder/decoder head and an upscaling factor of 8 for the hidden layers (i.e., the ratio between hidden and input layer dimensions).

## 4.2 Automatic Annotation of Cells

This section is dedicated to the analysis of the performance of our model in the task of automatic cell annotation.

### 4.2.1 Cytometry Datasets

The datasets used to assess the classification performance of our model consist of three public datasets:

| Dataset | Number of cells | Number of markers | Number of cell populations |
|---|---|---|---|
| AML | 104,184 | 32 | 14 |
| BMMC | 61,725 | 13 | 19 |
| POISED | 4,178,320 | 39 | 22 |

**Table 4.2: Summary of the characteristics of the three benchmark datasets.**

**AML** This dataset originates from Levine et al., 2015 (Benchmark Data Set 2) and consists of samples from two healthy adult donors (note: despite the name "AML" referring to the original study, these particular donors are healthy). The data were acquired using mass cytometry, measuring 32 surface markers on bone marrow mononuclear cells (BMMCs).

The samples were manually gated into 14 distinct immune cell types: the dataset comprises a total of 104,184 fully annotated cells.

**BMMC** The second dataset comes from Bendall et al., 2011, consisting of 61,725 bone marrow mononuclear cells from a single individual.

This dataset was also obtained via mass cytometry, with 13 markers measured to delineate 19 distinct cell populations.

**POISED** This final benchmark dataset originates from Chinthrajah et al., 2019, a clinical study on oral immunotherapy for peanut allergy. It includes 30 CyTOF (mass cytometry) samples derived from human PBMCs (peripheral blood mononuclear cells) of peanut-allergic individuals.

The 30 samples were collected from 15 patients under two experimental conditions: one time point corresponding to an untreated state, and another following peanut stimulation. The panel used comprised 39 markers, including 21 lineage markers and 18 functional markers.

In total, 22 distinct cell types were manually gated, among which peanut-specific immune cells were identified. Notably, the noncanonical peanut-reactive T cells analyzed in this study were CD69+ CD40L+ CD4+ T cells (TCD4 Peanut Reactive cells) and CD69+ CD8+ T cells (TCD8 Peanut Reactive cells).

In total, 4,178,320 cells were analyzed. Among them, 3,195,579 were labeled through a manual gating process, while 982,741 cells were left ungated, meaning they either had an unknown or irrelevant phenotype, or were known cell types that the expert simply chose not to annotate.

We will detail this dataset thoroughly, as it will be used in several distinct parts of our analyses and results. POISED is a complex dataset, due to both its high dimensionality and its strong imbalance. As shown in Table 4.3, some cell populations are overrepresented, such as naïve CD4/CD8 T cells, whereas the populations of interest (cells indicative of the immune response to peanuts) are underrepresented, appearing in less than 0.1% of the dataset for TCD4 Peanut reactive cells, for

example. It is indeed quite difficult to incorporate the importance of these subpopulations into the model, since they are minorities and are therefore rarely encountered during training.

Handling the unlabeled data is also a challenge in itself, as no information is available regarding the nature of these populations, which could be highly diverse or very similar to the labeled ones.

| Population | Proportion (%) |
|---|---|
| Unknown (Ungated) | 23.52 |
| TCD4 Naive | 21.13 |
| TCD8 Naive | 17.81 |
| TCD4 CM | 9.12 |
| NK | 7.21 |
| cMonocytes | 5.78 |
| B Naive | 3.78 |
| TCD8 CM | 2.01 |
| gdTCR Mem | 1.40 |
| Treg Mem | 1.38 |
| TCD8 EM | 1.31 |
| TCD4 EM | 1.23 |
| Treg Naive | 1.13 |
| B Mem | 0.84 |
| mDC1 | 0.55 |
| gdTCR Naive | 0.48 |
| NKT | 0.44 |
| iMonocytes | 0.32 |
| TCD8 PeaReactive | 0.25 |
| ncMonocytes | 0.18 |
| TCD4 PeaReactive | 0.09 |
| mDC2 | 0.03 |
| pDCs | 0.03 |

**Table 4.3: Cell populations from the POISED dataset.**

### 4.2.2 Comparison Against Scyan

We chose to compare our model to Scyan (described in Section 1.5.3, from Blampey et al., 2023), since an extensive comparison with other popular supervised and unsupervised models (Sections 1.5.1 and 1.5.2) was already performed in the Scyan paper. In that analysis, Scyan ranked first in terms of classification performance across all datasets, except for the balanced accuracy on the POISED dataset, where the LDA model performed slightly better.

We decided to compare ourselves exclusively to Scyan because it is also a modern deep generative model, aiming to achieve high classification performance without relying on explicit cell type labels, but instead leveraging a knowledge table that describes marker expression profiles for each cell type. This enables the classification of cells that were left unlabeled by experts.

**Data Preprocessing.** To ensure a reliable comparison with Scyan, the input data were preprocessed in the same way as described by Blampey et al., 2023. The marker expression levels were transformed using the *asinh* transformation, namely $\mathbf{x} \to \text{asinh}(\mathbf{x}/5)$, a common practice for mass

cytometry data (used in Bendall et al., 2011, Levine et al., 2015, Lee et al., 2017, and extensively described in the guidelines presented by Nowicka et al., 2019).

Mass cytometry (CyTOF) data can span several orders of magnitude in marker expression. The $asinh(\mathbf{x}/5)$ transformation compresses high-intensity signals while preserving low-intensity ones.

The data were then standardized, as this is required in the case of Scyan to ensure compatibility with the knowledge table.

**Training Setup.** To ensure a fair comparison between our model and Scyan, it is essential to align the training setup. Indeed, despite the availability of numerous markers in the various datasets, the knowledge table defined for each dataset does not make use of the full marker panel. In practice, only 14 out of the 32 available markers were used for the AML dataset, and 19 out of 39 for the POISED dataset. This strategy reflects the fact that not all markers are necessarily relevant for the manual annotation of the chosen populations.

However, this need to refine the selected markers can be tedious, potentially biased, and introduces an additional preprocessing step. In contrast, we demonstrate in Figure 4.3 that when using the full set of markers provided in the datasets, the performance of our model does not degrade much.

**Results.** MARVIN outperforms Scyan across all datasets and metrics. For the AML, BMMC, and POISED datasets, MARVIN achieves superior accuracy (0.99 vs 0.98, 0.99 vs 0.96, 0.95 vs 0.81), balanced accuracy (0.98 vs 0.89, 0.93 vs 0.80, 0.90 vs 0.78), and F1-scores (0.97 vs 0.83, 0.92 vs 0.75, 0.80 vs 0.58), respectively (Figure 4.1). The performance gap is most pronounced in the complex POISED dataset.

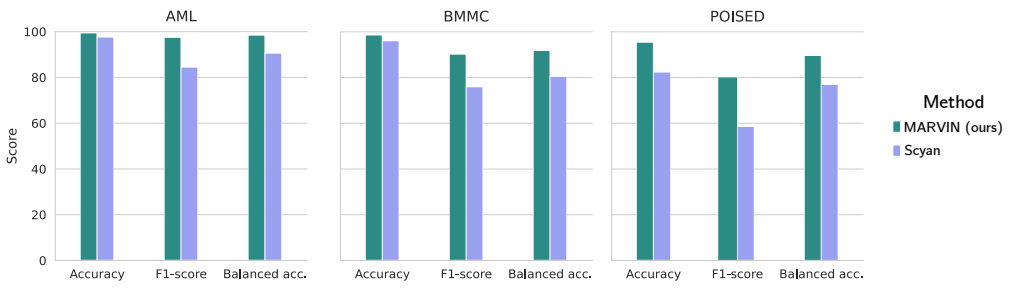



**Figure 4.1: Cell classification performances against Scyan on three public mass cytometry datasets.**

Additional results are provided in Appendix B, including per-cell-type metrics (Recall and F1-score), confusion matrices for each dataset and method, as well as UMAP visualizations of the input data space with overlaid annotations, similar to Figure 4.2, but for the AML and BMMC datasets.

**A Note on Markers Preselection.** Since we found it surprising that many models (Scyan, CyAnno, LDA, ...) do not use the full panel of markers but instead refine it to retain only those most relevant for identifying specific populations, we decided to conduct a study comparing the performance of our model when using the full marker panel versus when refining it in the same way as done in Scyan.

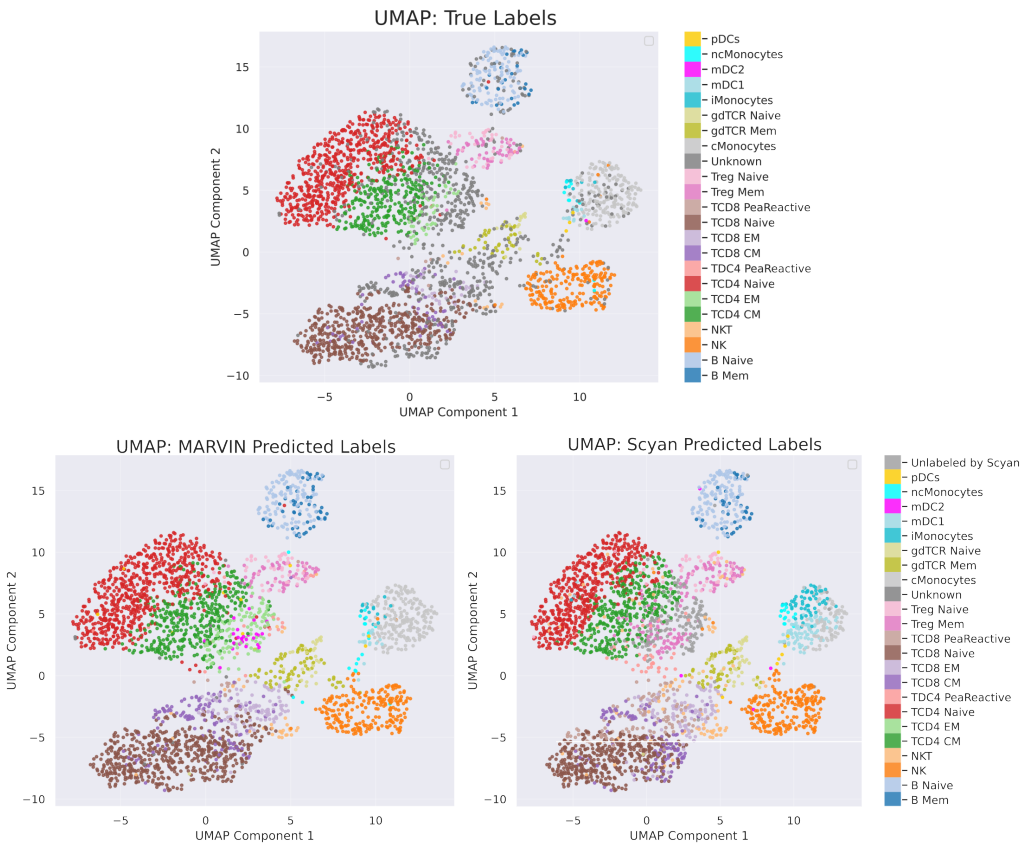



**Figure 4.2: UMAP (McInnes et al., 2020) display of 3000 cells from POISED dataset (input space).** Cells are colored by cell-type assignments established by manual gating, MARVIN and Scyan.

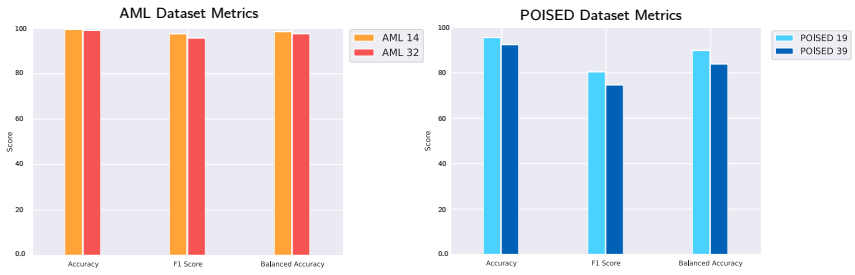



**Figure 4.3: Study of the impact of the number of markers used on the model's performance.** The model was trained on the AML dataset using either the panel of 14 markers selected by Scyan or the full marker panel. For POISED, 19 markers had been selected by Scyan. Metrics are assessed on the test set $\mathcal{T}_{\text{test}}$.

As can be seen in Figure 4.3, the performance of the model decreases slightly when no marker preselection is applied. However this selection requires expert input to determine which markers are relevant for annotating the cell types of interest. In contrast, our goal is to develop a model that is as unbiased as possible: one that could be used by any clinician wishing to analyze cytometry data, whether or not they are an expert. Figure 4.3 shows that the performance does not drop drastically, and that even when using the full marker panel, our model still outperforms Scyan, despite the latter benefiting from a refined panel.

### 4.2.3 Effect of the Supervision

Here, we evaluate the model's performance under different levels of supervision. By level of supervision, we refer to the percentage of each class that has been "seen" by the model during training, that is, the proportion of labels that contributed to the supervised loss $\mathcal{L}_{\text{supervised}}$. For the *10% supervised model*, for example, 90% of the labels in each class were artificially masked: these cells contribute to training solely through their effect on the ELBO. Masking a percentage of cells per class is, in expectation, equivalent to masking the same percentage of the dataset overall, but this approach ensures that the model sees at least one label from each class, some of which may be very underrepresented.

One important note about expert annotations is that they are likely not independent: the gate is placed subjectively based on a threshold defined by the expert, and some cells that lie in the tails of their distribution (i.e., with the same label but slightly different and rarer phenotypic expression) are probably annotated less frequently or with less certainty. For simplicity, the mask applied on the label is here drawn at random.

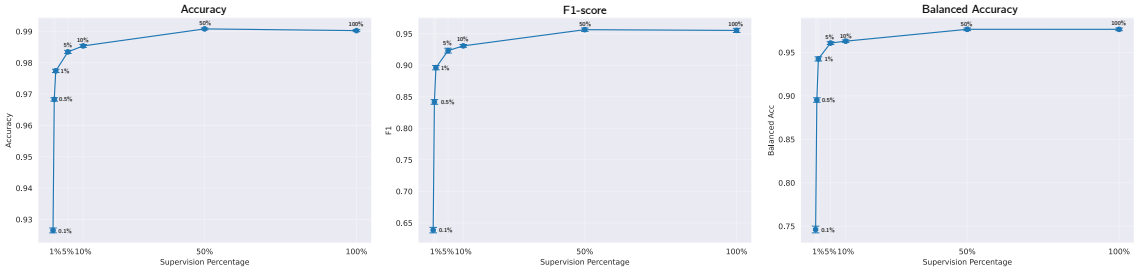



**Figure 4.4: Effect of the supervision level on the model's annotation performance, on AML dataset.** The metrics are assessed on the test set $\mathcal{T}_{\text{test}}$.

Figure 4.4 displays the performance of models trained with different levels of supervision (0.1%, 0.5%, 1%, 5%, 10%, 50%, and 100%) on the AML dataset, using the full panel of 32 markers. The metrics are reported as mean ± standard deviation to reflect model uncertainty due to probabilistic class sampling.

These results highlight that the model remains highly efficient even with minimal supervision. Notably, the 50% supervised model slightly outperforms the fully supervised one across all metrics, which may suggest that partial masking of the labels helps reduce overfitting to potentially biased expert annotations. The model is encouraged to rely more on the underlying structure of the data through the unsupervised objective. This acts as a form of implicit regularization, helping the model to generalize better. Additionally, masking a portion of the labels forces the model to balance both supervised and unsupervised signals, which can result in a more robust latent representation.

Figure 4.5 shows that, regardless of the level of supervision, the latent representation $q_\phi(\mathbf{z}|\mathbf{x}, c)$ remains disentangled: even when the model has seen only a small number of labeled cells, it is still able to organize the latent space into meaningful clusters.

However, when supervision is extremely limited (i.e., 0.1%), some clusters that are typically close yet still distinguishable (such as $CD16^+$ and $CD16^-$ NK cells) become entangled and indistinguishable. These two populations are phenotypically very similar, so their proximity in latent space is expected. Still, with only a handful of labeled examples, the model fails to separate them and

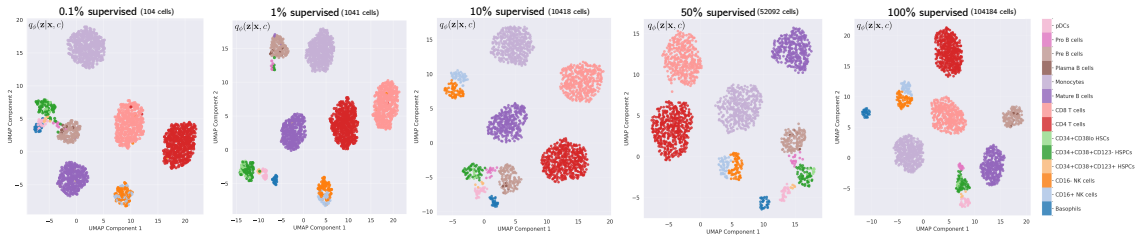



**Figure 4.5: Latent UMAP representation $q_\phi(\mathbf{z}|\mathbf{x}, c)$ across five levels of supervision.** All representations are derived from the same batch of 2000 cells. Each latent space is generated using the corresponding model's predicted labels $c$ and the input marker expression $\mathbf{x}$. To ensure a fair comparison between conditions, cells are colored according to their ground-truth labels rather than model predictions.

merges them into a single cluster, even in classification. This behavior is not surprising given the minimal supervision available. Starting at just 1% supervision, the model already shows robust performance: it begins to clearly separate similar cell types into distinct and well-defined clusters. From that level onward, clusters are easily distinguishable in all cases. When the model is fully supervised, the clusters become slightly more compact, although the UMAP parameters used to visualize the embeddings are kept identical across all subfigures. This highlights the benefit of incorporating even weak supervision, which significantly improves cluster separability and reinforces the structured organization of the latent space.

This section has demonstrated the robustness of our model within a semi-supervised learning framework. It shows that MARVIN can significantly reduce the expert's workload in cytometry data analysis. In a clinical study setting, it would be sufficient to annotate the cellular populations of just a single patient: the model can then automatically annotate the populations of all remaining patients in the cohort. This drastically lowers the annotation burden while maintaining strong performance, and offers a scalable solution for large-scale cytometry analysis where manual gating would otherwise be prohibitively time-consuming.

**Theoretical Justification.** We can provide a theoretical justification for the fact that the model's performance degrades only slightly when the number of labeled examples seen during training is reduced. Due to the construction of our ELBO, the unsupervised component of the loss still contributes to classification. This means that even if some cells do not provide a direct supervision signal through $q_\omega(c|\mathbf{x})$, they still help improve the model's annotation capabilities.

In practice, the parameters of $q_\omega(c \mid \mathbf{x})$ are updated through several components of the unsupervised loss. The divergence $\mathrm{KL}(q_\omega(c \mid \mathbf{x}) \| p_\pi(c))$ acts as a regularization term that prevents mode collapse by encouraging the posterior to remain close to the prior class distribution. Additionally, because the ELBO includes an expectation over all possible cluster assignments, the term

$$\mathbb{E}_{q_\omega(c|\mathrm{x})}\left[\mathrm{KL}\left(q_\phi(\mathbf{z} \mid \mathbf{x}, c) \| p_\beta(\mathbf{z} \mid c)\right)\right]$$

encourages $q_\omega(c \mid \mathbf{x})$ to assign higher probability to the classes for which the variational posterior aligns well with the prior in latent space. A similar mechanism applies to the reconstruction term, where the latent variable $\mathbf{z}$ is structured according to cell type, which in turn influences reconstruction quality.

As a result, the model is still able to learn accurate cluster assignments and update the gradients of $q_\omega(c|\mathbf{x})$ in directions that support classification, even in the absence of explicit supervision.

**Unsupervised Model.** Although it is not intended as a model for annotation, we found it important to display the latent representation of a model trained in a fully unsupervised manner, as this reflects the framework in which VC (Prasad et al., 2020), the unsupervised model that is similar to ours, was originally proposed. Figure 4.6 shows the latent representation of $q_\phi(\mathbf{z}|\mathbf{x}, c)$ for the fully

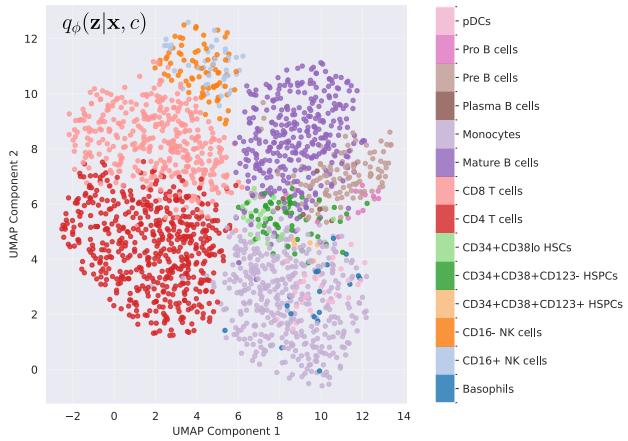



**Figure 4.6: Latent UMAP representation $q_\phi(\mathbf{z}|\mathbf{x}, c)$ for the unsupervised model.** The 2000 cells are colored by cell-type assignments established by manual gating (ground truth labels).

unsupervised model. Since the model is not capable of annotating cells by itself ($q_\omega(c|\mathbf{x})$ having never received any supervision, it only enables soft cluster assignments), the cells are colored using the ground truth labels.

Despite the fact that the model never received any supervision signal during training, we can observe that cells from the same class are grouped together, indicating that the model gathers phenotypically similar cells in latent space. This behavior arises simply because it is beneficial in terms of reconstruction loss and the prior-matching term, which structures the latent space as a Gaussian mixture.

However, the clustering of cell populations is much less pronounced than when a supervision signal is provided. While cells from the same class appear close to one another, we still observe significant overlap between similar populations, and the separations are not as sharp as one might expect. For example, all HSPC cells appear gathered together, regardless of their CD38 and CD123 expression profiles, which normally allow for clear separation when supervision is injected into $q_\omega(c|\mathbf{x})$.

Two main insights can be drawn from this observation. First, without supervision, the model has no incentive to separate highly similar populations, since they are already close in the input space: the lack of separation suggests that reconstruction alone does not provide enough pressure to disentangle subtly different but phenotypically close cell types.

In Section 4.6, we will compare this result to the latent space of a standard VAE, using Figure 4.6 as a point of reference.

## 4.3 Discovery of Novel Cellular Subpopulations

The detection of cellular subpopulations is a critical task in cytometry. The use of manual gating to identify such highly specific populations is particularly labor-intensive, as the expert must

define a sequential strategy to precisely isolate these subsets. It is therefore clear that having a model capable of automatically partitioning cells into subpopulations is essential for enabling high-throughput and unbiased analysis of complex datasets.

Many unsupervised clustering methods have been developed for this purpose (see Section 1.5.2). However, they do not perform cell annotation simultaneously and thus still require expert interpretation to analyze the resulting clusters.

Our objective is to develop a model within a unified framework that allows for both classification and the discovery of novel subpopulations that are distinct from annotated ones.

The model should therefore be able, in fully or partially labeled datasets, to identify subclusters of cells with specific features and to separate them from other populations.

### 4.3.1 Fixing p(c)

To achieve this, one might consider simply increasing the number of clusters $K$ specified in the model. However, a problem arises: in the second term of the ELBO, $-\mathbb{E}p(\mathbf{x})\left[\mathrm{KL}(q\omega(c \mid \mathbf{x})|p(c))\right]$ (Eq. 3.16), the prior $p_\pi(c)$ is encouraged to match the empirical cluster assignment frequencies observed in $q_\omega(c|\mathbf{x})$ through the supervised loss (Eq. 3.21). If a class $k$ never receives any supervision signal, the network, attempting to match $p_\pi(c)$, will tend to redistribute mass across the clusters seen during supervision, effectively diluting the unused ones.

In our model, we perform an expectation over all clusters rather than relying on hard assignments (see Section 3.1.7). At the beginning of training, since the prior is initialized uniform, $q_\omega(c|\mathbf{x})$ tends to spread across all clusters. If a cluster never receives a supervision signal (either because it corresponds to a supplementary cluster or because its class has been artificially masked) the model has no incentive to update $q_\omega(c = k|\mathbf{x})$, and will thus ignore that cluster.

However, in the case of an artificially masked class, if this class is prevalent in the dataset, the reconstruction loss worsens when this class is diluted. To optimize reconstruction and the prior-matching term, the model will eventually start using the additional cluster. However, this behavior is not observed for minority populations, which are precisely the ones we are most interested in.

To enable the discovery of rare subpopulations that resemble known cell types in several respects but differ by specific marker expressions, and to prevent their collapse into already supervised categories, we need to address the tendency of the model to assign cells preferentially to clusters that receive supervision signals.

A straightforward solution is to stop learning the prior $p_\pi(c)$ and instead fix it at the beginning of training. When the model is used in a subpopulation discovery context, the prior can be initialized following a biologically-informed strategy.

Indeed, the proportions of immune cells in the blood of a healthy individual are relatively standardized, so we can make reasonable assumptions (more or less broad) about the expected distribution of cell types in a typical patient.

To allow for the emergence of novel subpopulations, we extend the number of clusters beyond the number of expected known populations (e.g., if 22 populations have already been identified and we want to discover 2 potential new subpopulations, we set $K = 24$). We then allocate a small prior mass to these additional "discovery clusters", under the hypothesis that such subpopulations

are rare. In practice, after normalization, the prior probability assigned to each discovery cluster is fixed to 0.1%.

### 4.3.2 Experiment Setup

To design an experiment targeting the discovery of relevant subpopulations, we selected the POISED dataset. This dataset is particularly suited for subpopulation discovery, as it contains immune cells from peanut-allergic patients that exhibit non-canonical phenotypes associated with allergen-specific responses. Specifically, the dataset includes two rare cell types annotated as TCD4 Peanut Reactive and TCD8 Peanut Reactive cells.

These cells are phenotypically close to major populations such as TCD4/8 cells and T-Regulatory Memory cells (Treg Mem), yet differ through the expression of specific activation markers: CD69+ CD40L+ CD4+ T cells for the TCD4 Peanut Reactive population, and CD69+ CD8+ T cells for the TCD8 Peanut Reactive population.

They are extremely underrepresented in the dataset, accounting for only 0.1% (TCD4 Peanut Reactive) and 0.25% (TCD8 Peanut Reactive) of the total cells (see Table 4.3). These cells, which are specific to the peanut-induced allergic response, are therefore ideal candidates to test our model's ability to isolate minority populations with well-defined but not easily detectable characteristics.

To demonstrate the strength of our model, we chose to focus on the most minority population of interest, namely the TCD4 Peanut Reactive cells (present in 0.1% of the dataset).

**Removing Ungated Cells.** Since no information was provided by Chinthrajah et al., 2019 regarding the ungated cells, it is impossible to know whether they consist of multiple, highly heterogeneous subpopulations or if they resemble the already annotated cell types. In Section 4.2, Figure 4.2 shows that ungated cells are scattered among phenotypically close clusters, due to the phenomenon explained in Subsection 4.3.1. However, this does not guarantee compatibility between ungated cells and annotated types. Thus, it is difficult to set a prior on cell populations or to determine how many additional clusters to use to separate the population of interest without any information about the ungated cells. For this reason, we chose to remove these cells from the dataset when the task is to segregate a particular subpopulation. However, to ensure a rigorous approach, we also conducted an additional analysis, presented in Appendix C, in which the ungated cells were kept in the dataset for the same task.

**Training Setup.** To demonstrate the ability of our model to separate a cell population that was not explicitly distinguished from others through a supervision signal, we proceeded as follows:

- The TCD4 Peanut Reactive labeled cells were completely masked during training, meaning that these cells contributed only to the unsupervised loss (ELBO) but did not contribute to the supervised loss $\mathcal{L}_{\text{supervised}}$. The rest of the labels were fully used, which means that for all other cells, the training was fully supervised.
- To account for the possibility that other subpopulations distinct from the TCD4 Peanut Reactive cells may exist in the dataset, we set $K = 24$. Since the model is exposed to 21

annotated cell populations (with the TCD4 Peanut Reactive cells being masked), this leaves three clusters available for the model to map potential discovered subpopulations.

- The prior probability of these clusters is set to 0.1%, which reflects the observed frequency of TCD4 Peanut Reactive cells in the dataset. For the remaining clusters, for the sake of simplicity, the prior $p(c)$ is set to match the empirical frequency of each annotated cell population in the dataset.

### 4.3.3 Results

In this section, we will analyze the subpopulations isolated by our model within the clusters designated for this purpose. We will also examine the variation of the discovered population of interest, the TCD4 Peanut Reactive cells, between stimulated and non-stimulated patients.

From a theoretical standpoint, it is important to understand that these populations are separated from others because they have a distinct phenotypic profile, but not only that: the model also benefits from this split in terms of reconstruction quality.

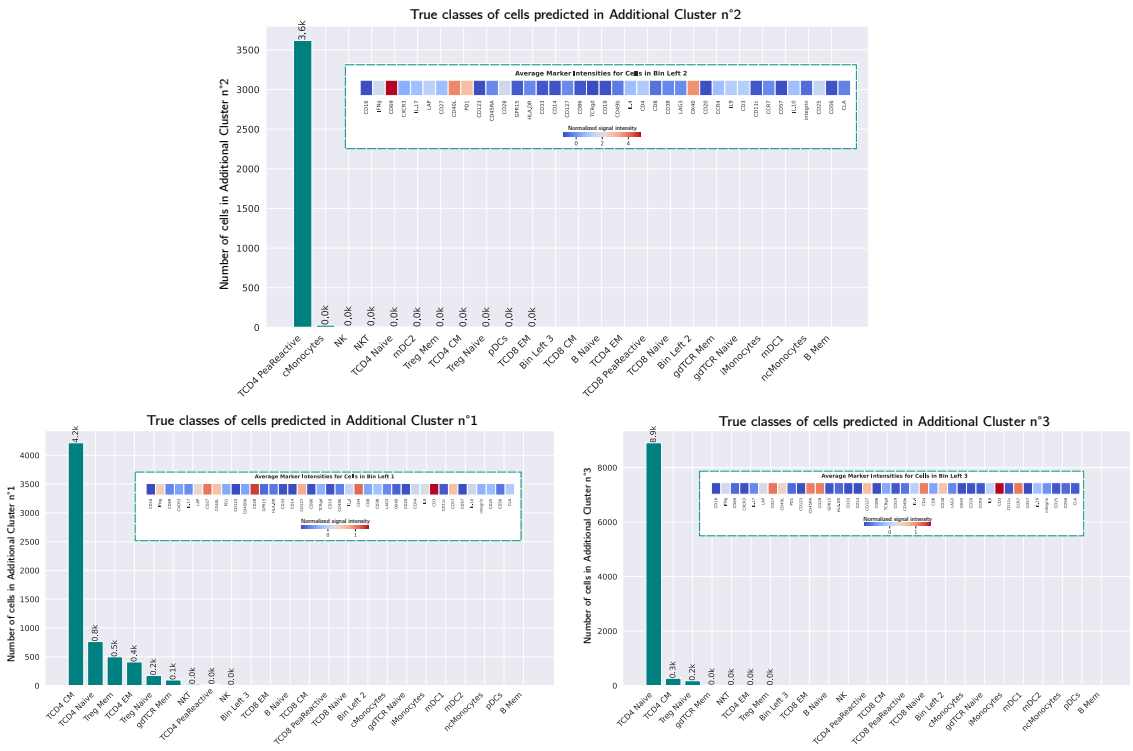


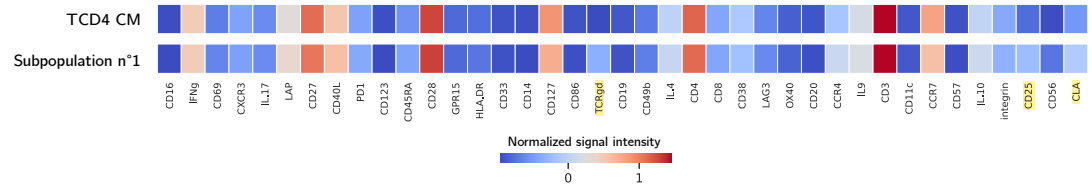

**Figure 4.7: The three subpopulations discovered by the model.** Subfigures displays the initial labels of the cells split into new subpopulations. The average marker expression is shown to allow experts to assess their differences compared to other cell types.
Top: the TCD4 Peanut Reactive population identified by the model and mapped to the second additional cluster. Left: a subpopulation emerging from the TCD4 CM cells. Right: a subpopulation emerging from the TCD4 Naive cells. See Appendix C for more information.

Figure 4.7 displays the three subpopulations discovered by our model. In the second additional cluster, the model successfully recovered and isolated the TCD4 Peanut Reactive population,

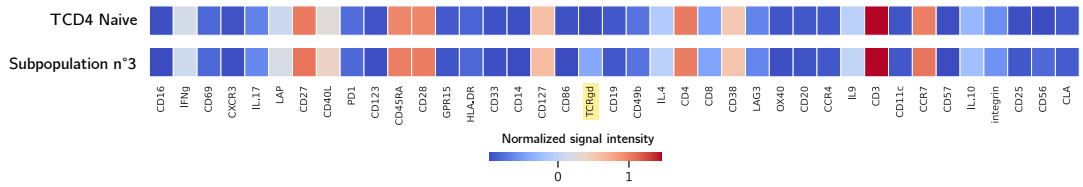
even though it was entirely masked during training. According to Chinthrajah et al., 2019, this population was manually gated based on its positive expression of CD69 and CD40L.

TCD4 Peanut Reactive cells are, by definition, CD4+ T lymphocytes, like TCD4 Naive, Central Memory (CM), and Effector Memory (EM) cells. Since they are memory cells involved in the immune response to peanuts, they also share similarities with Treg Memory cells (Regulatory Memory T cells). Despite their rarity, the model was able to successfully distinguish them from phenotypically similar populations.

Interestingly, although TCD8 Peanut Reactive cells also express CD69, the model did not map them into this same cluster, suggesting that it leveraged more than just a single marker to define the subpopulation.




**(A) TCD4 CM cells vs. its emerging subpopulation (n°1).** The emerging subpopulation differs from its average expression of TCRgd, CD25, and CLA, as well as slightly from CCR7.




**(B) TCD4 Naive cells vs. its emerging subpopulation (n°3).** The emerging subpopulation differs from its average expression of TCRgd and a bit from CD40L.

**Figure 4.8: Difference between marker expression in global populations and emerging subpopulations.**

The two other subpopulations isolated by the model come from from broader T cell categories: TCD4 Central Memory (CM) and TCD4 Naive cells, respectively, as shown in Figure 4.7. A more detailed analysis of their marker expression profiles (Figure 4.8) helps interpret the phenotypic differences that led the model to separate these subpopulations.

For instance, subpopulation 1 (Figure 4.8A) which was initially gated predominantly as TCD4 CM by the expert, differs from this broader group through the expression of several markers, including higher levels of TCR$\gamma\delta$, CD25, CLA, and a slightly altered expression of CCR7. These phenotypic distinctions likely explain why the model identified this subgroup as a separate population.

Similarly, the subpopulation emerging from the TCD4 Naive cells (Figure 4.8B) also displays a more pronounced expression of TCR$\gamma\delta$ compared to the original naive population.

It is important to highlight that although these subpopulations express higher levels of TCR$\gamma\delta$, they are not classified as gdTCR Memory or gdTCR Naive cells (i.e., TCR$\gamma\delta$+ cells). This indicates that the model does not rely solely on a single marker to define a population. Instead, it captures complex, high-dimensional cellular signatures by integrating multiple marker expression patterns

simultaneously within its latent space.

What is also interesting is that despite the presence of a supervisory signal encouraging subpopulations 1 and 3 to align with broader cell categories, the model still chose to separate them into distinct clusters. This likely reflects an additional benefit in terms of reconstruction accuracy, suggesting that the model favors biologically meaningful separations when they improve overall data representation.

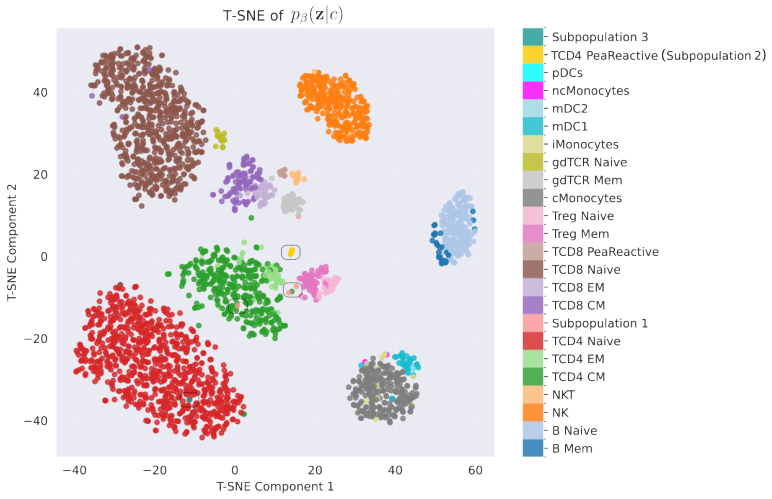



**Figure 4.9: T-SNE Latent representation of $p_\beta(\mathbf{z}|c)$ for the POISED dataset.** 3000 cells were sampled from $p_\pi(c)$ and a latent representation $\mathbf{z}$ was displayed for each.

Additional analyses regarding the relevance of removing Ungated cells from the dataset were conducted and are detailed in Appendix C.

Figure 4.9 displays the latent prior distribution learned by the model. Specifically, 3000 cell classes were sampled from the categorical prior $p_\pi(c)$ and passed through the conditional prior $p_\beta(\mathbf{z} \mid c)$ to visualize the latent space using t-SNE (Maaten and Hinton, 2008), a dimensionality reduction technique that projects the 64-dimensional latent space onto a 2D plane for easier interpretation.

We observe that the model successfully separates the second subpopulation (TCD4 Peanut Reactive) from the rest, despite its close phenotypic similarity to TCD4 Effector/Memory (EM/CM) and regulatory memory T cells (TReg Mem) that are therefore close in the latent space. In contrast, subpopulations 1 and 3, which emerge from canonical cell types, are less distinct in the 2D representation. They appear visually merged with their parent populations, suggesting that their separation is more subtle and less pronounced compared to TCD4 Peanut Reactive cells.

**Conclusion.** This section has demonstrated the model's ability to isolate rare subpopulations of interest, even when they are highly underrepresented in the dataset. By projecting these discovered clusters back into the original marker expression space (for instance, using a heatmap of average marker intensities within the subpopulation) the model also enables biological interpretability of such findings. This allows experts to not only detect novel cellular phenotypes but also to characterize them in a biologically meaningful way.

## 4.4 Patients' Cellular Dynamics Across Experimental States

This section builds on the previous one (Section 4.3), using the same model that was trained to identify cellular subpopulations, in particular the non-canonical TCD4 Peanut-reactive cells. Beyond isolating phenotypically distinct subsets from the broader cellular landscape, our goal here is to demonstrate the model's ability to characterize differences between cell populations under varying experimental conditions.

The POISED dataset comprises 15 patients, each profiled under two conditions: peanut-stimulated and unstimulated. The dataset originates from a clinical study by Chinthrajah et al., 2019, which aimed to assess immune cell responses and the effects of peanut allergy following oral immunotherapy. In this section, we show that our model can autonomously analyze and distinguish cellular responses across these conditions, substantially reducing the expert's workload by automatically identifying phenotypic shifts induced by stimulation.

### 4.4.1 Results

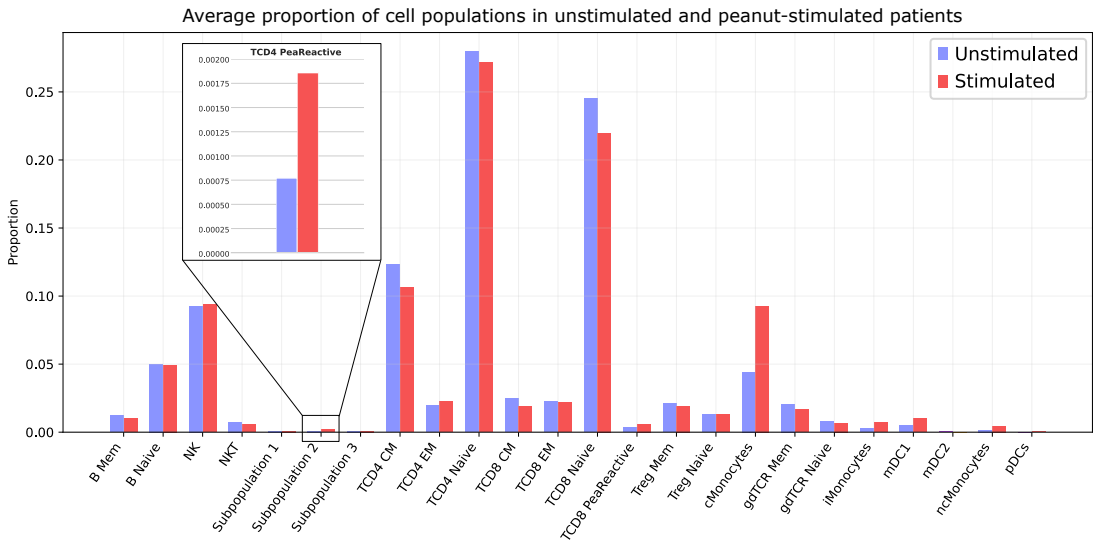



**Figure 4.10: Proportion of cell populations under unstimulated vs. peanut-stimulated conditions, averaged across all patients.** The discovered TCD4 Peanut-Reactive population is highlighted.

Figure 4.10 shows the average proportions of the different cell populations across all peanut-allergic patients, under stimulated (peanut ingestion) and unstimulated (baseline) conditions. The TCD4 cell population labeled as Peanut Reactive and highlighted by our model is clearly more abundant after stimulation, which is expected: in allergic patients, ingestion of the allergen leads to rapid activation and clonal expansion of peanut-specific CD4 T lymphocytes. Notably, a non-negligible proportion of these cells is also detected in the absence of stimulation. This reflects a well-known immunological phenomenon called *immune memory*: in sensitized individuals, the immune system retains a memory of the allergen. In other words, a pool of memory TCD4 cells specific to peanut is present at baseline and is capable of mounting a rapid response upon re-exposure.

Indeed, CD69, CD40L and OX40 are canonical markers of activated T cells (Lemieux et al., 2024). However, in the unstimulated condition, Peanut Reactive TCD4 cells are likely to exhibit

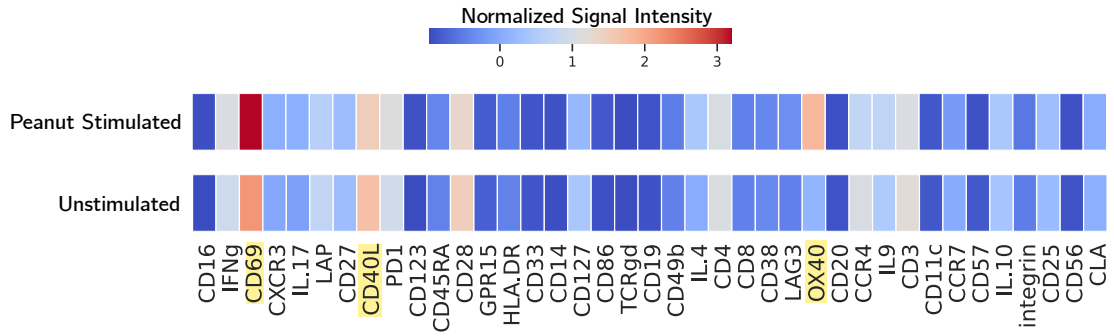



**Figure 4.11: Difference in average marker expression of TCD4 Peanut Reactive cells between peanut-stimulated and unstimulated patient samples.**

a phenotype resembling that of their post-stimulation counterparts, without necessarily expressing high levels of activation markers like OX40 and CD69, while still retaining their antigen specificity.

This is precisely what our model captures, as shown in Figure 4.11: despite noticeable differences in average marker expression between stimulated and unstimulated conditions, these TCD4 Peanut Reactive cells are consistently mapped to the same cluster.

Trained on high-dimensional expression data, the model identifies as Peanut Reactive not only the activated (CD69+, CD40L+ and OX40+) cells after allergen exposure, but also the memory cells present at baseline that share sufficiently similar transcriptional or phenotypic features.

This capacity to generalize illustrates the model's ability to capture the latent characteristics of peanut-reactive TCD4 cells, independently of their activation state at the time of sampling.

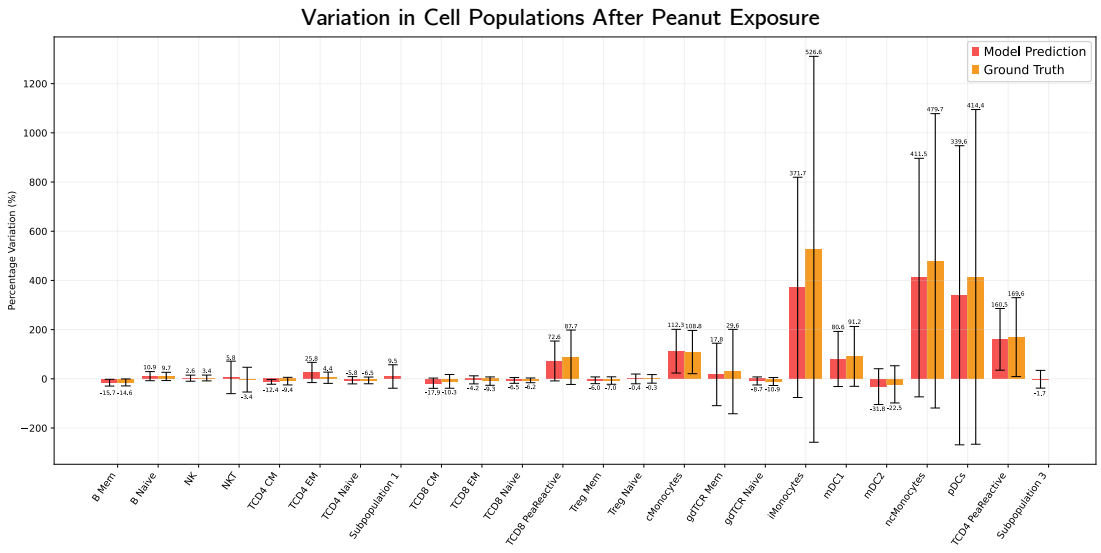



**Figure 4.12: Variation in cell populations after peanut stimulation.** The lines indicate the standard deviation across the 15 patients, while the percentage displayed above/below each bar represents the *average* variation across patients.

These analyses of population dynamics following stimulation can be further explored. To study

the responses to stimulation while appropriately visualizing changes even in rare subpopulations, we examine Figure 4.12. To account for inter-patient variability, the percentage change in the different cell populations is expressed as the average across the 15 patients, with the inter-patient standard deviation displayed on the graph.

Indeed, it is immediately apparent that the immune response to stimulation is not homogeneous across all patients: the standard deviation is very high for almost all cell populations. This observation reflects the intrinsic heterogeneity of human immune responses. Some patients exhibit strong cellular activation following peanut exposure, while others show moderate or even minimal changes. This variation likely stems from differences in individual sensitivity thresholds, which determine how readily a person's immune system reacts to a given allergen dose.

In addition, genetic and epigenetic variability between individuals plays a critical role (Cardenas et al., 2023). Genes involved in allergen recognition, intracellular immune signaling pathways, and cytokine production or regulation can differ considerably between patients. These genetic differences influence the activation and recruitment of specific immune cell subsets upon allergen exposure, contributing to the observed inter-individual variability.

Furthermore, previous exposure history, baseline immune status, and potential co-existing conditions (e.g., asthma, eczema, other allergies) may further modulate the immune response, amplifying variation in cell population dynamics after stimulation.

Moreover, even though the experimental techniques are rigorous, differences can arise from sample quality, the number of cells analyzed, or cytometer performance. All these factors introduce experimental noise.

Drawing conclusions based solely on average behavior is challenging, as each patient's immune response varies considerably. Nonetheless, Figure 4.12 reveals that, on average (despite large inter-patient differences) there is a noticeable increase in monocyte populations, including classical (cMonocytes), intermediate (iMonocytes), and non-classical monocytes (ncMonocytes). This trend is expected, as an elevated presence of monocytes reflects the activation of a systemic inflammatory response, characterized by the production of inflammatory signals and antigen-presenting activity. Monocytes support the activation and differentiation of CD4+ T lymphocytes.

We also observe an increase in dendritic cell populations (pDCs and mDC1), which are likewise specialized in presenting antigens to CD4+ T cells.

Our results can be validated and compared with changes in cell populations identified by manual gating, which also display high variability and reveal immune response mechanisms consistent with the predictions made by our model. It is worth noting that our model identifies two additional subpopulations not present in the ground truth labels, which may partly explain the observed discrepancies and variability.

These analyses of cell population shifts following peanut stimulation can be extended beyond population-level averages to a *per-patient perspective*. Our model captures the heterogeneity in immune responses across individual patients, which is particularly valuable in a clinical study setting. Such insights enable the identification of patients whose responses deviate from the norm, paving the way for a more personalized medicine approach.

As shown in Figure 4.13, patient responses are highly variable, including their baseline cell population proportions. For instance, the TCD4 Peanut Reactive population increases after stimulation in nearly all patients, except for patient P116 and P111, suggesting an atypical immune profile.

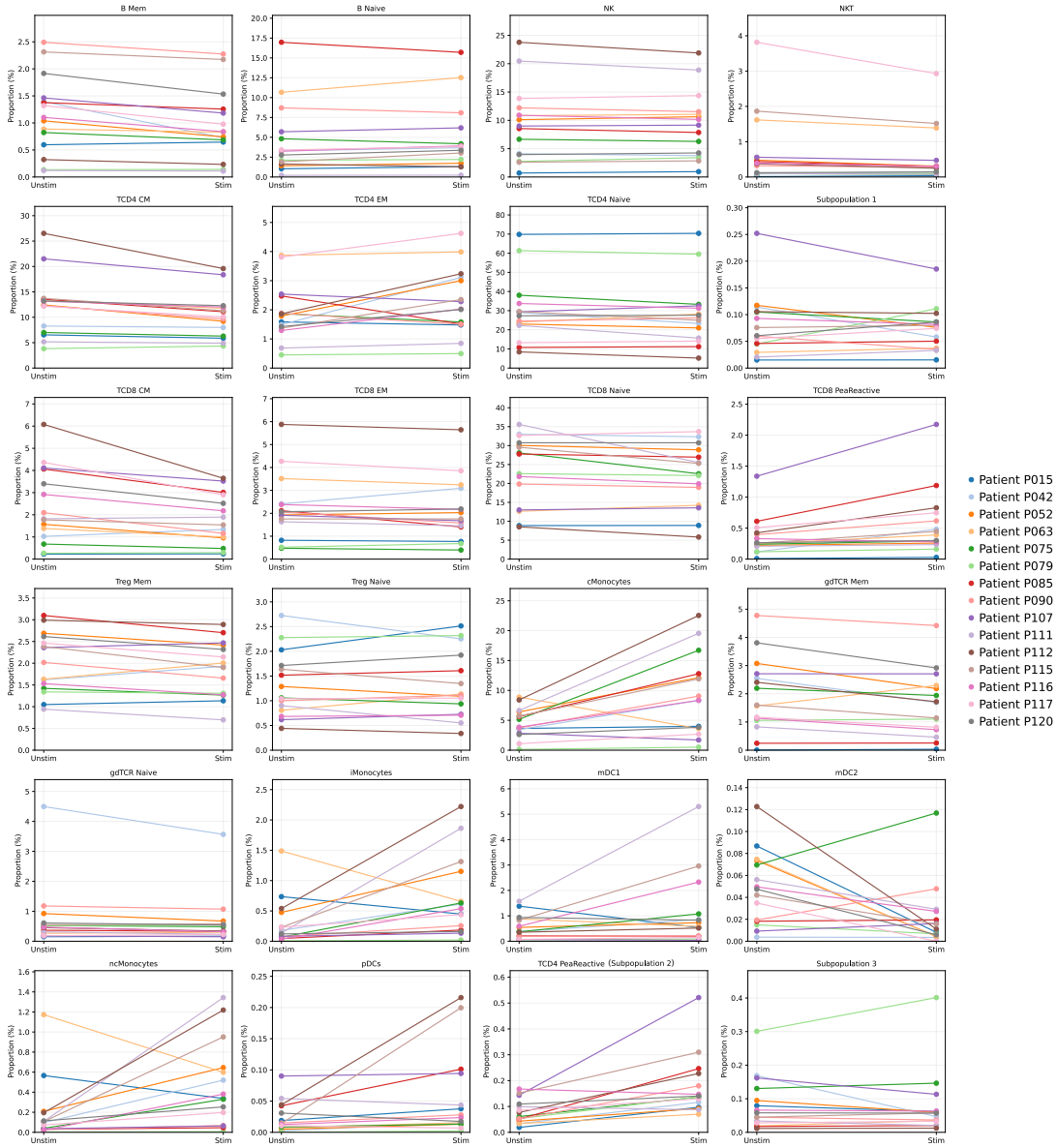



**Figure 4.13: Patient-specific evolution of cell population proportions between the unstimulated and peanut-stimulated conditions.**

Similarly, classical monocytes increase in all patients except P063 and P107, further illustrating inter-patient diversity.

This type of analysis enables a more detailed characterization of the immune response to peanut allergens, helping to understand why some allergic patients respond better or worse to immunotherapy. Ultimately, this can guide the development of more targeted treatments and improve patient stratification in clinical trials.

## 4.5 Anomaly Detection

One of the clinical applications of cytometry is the assessment of Minimal Residual Disease (MRD), referring (in the context of leukemia) to the small number of cancerous leukemic cells that remain in a patient after treatment but fall below the detection threshold of conventional morphology-based diagnostics, such as microscopic examination of blood or bone marrow smears.

Although a patient may be considered in complete remission based on clinical or hematological criteria, these residual cells can cause relapse. Therefore, detecting MRD is critical for prognosis and for guiding therapeutic decisions.

Our model can be leveraged to address three distinct tasks related to minimal residual disease (MRD):

- **Classical MRD detection**, where the model is trained on leukemic cells and performs as a classifier to detect residual malignant populations;
- **Discovery of leukemia subtypes or residual disease**, by identifying novel subpopulations not previously annotated;
- **Anomaly detection**, which involves identifying abnormal cell populations, such as cancerous cells, that differ from those seen during training.

In this section, we focus on the last task. The first two have already been addressed in Sections 4.2 and 4.3, respectively.

Anomaly detection is a well-known application of standard VAEs (Nguyen et al., 2024), where the reconstruction loss is used to assess whether an input fits the learned data distribution. Inputs with high reconstruction error are assumed to lie outside this distribution and are therefore considered potential anomalies.

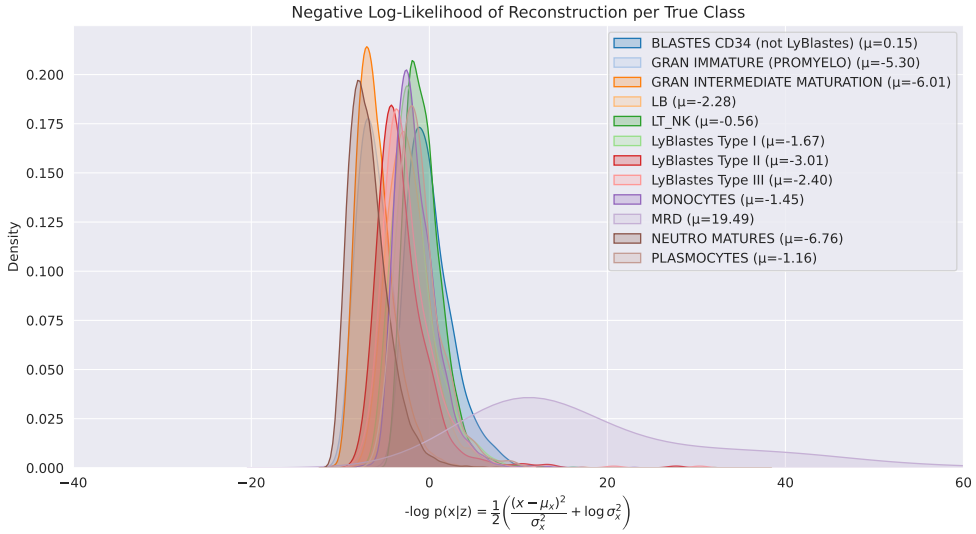
This section will exploit this idea in the case of our structured-VAE, MARVIN.

**Dataset.** The dataset comprises anonymized bone marrow samples collected at the CHU of Liège. It includes samples from three healthy donors, encompassing all major normal cell populations, as well as minimal residual disease (MRD) leukemic cells from four patients.

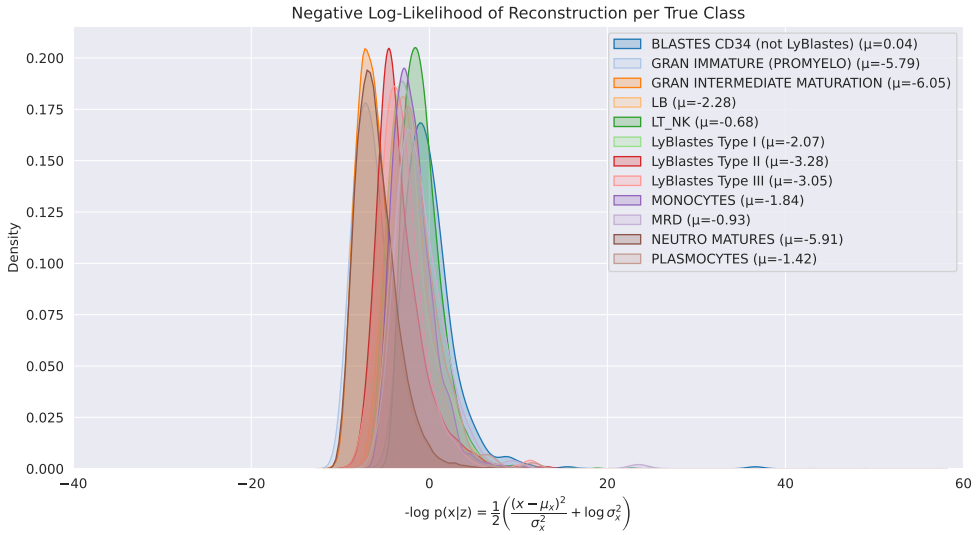
In total, the dataset contains 6,667,147 cells, of which 6,656,925 are healthy cells (from the three donors) and 10,222 are leukemic MRD cells, making the leukemic cells a clear minority despite originating from four different patients. The data span 12 distinct cell populations and 8 markers.

The data were transformed and standardized using the same preprocessing as previous datasets (including *asinh* transformation).

**Training Setup.** The model was trained exclusively on healthy patient data, with $K = 11$, the number of healthy populations. At test time, leukemic cells from different patients were introduced to evaluate the model's ability to detect anomalies. In parallel, a second model was trained on the full dataset, including both healthy and cancerous cells, in order to provide a reliable baseline for comparison of reconstruction performance on MRD. In both cases, cell type labels were kept and not masked during training.




**(A) Model trained exclusively on healthy patients.**




**(B) Model trained on all cell types.**

**Figure 4.14: Reconstruction error for each class, in both training settings.** The negative log-likelihood was computed for 500,000 cells and visualized "per-cell-type" using kernel density estimation (KDE).

**Results.** The figure 4.14 shows the negative log-likelihood (i.e., reconstruction loss) for each cell type, in both training settings. Unlike standard autoencoders that optimize a pointwise reconstruction via MSE, Variational Autoencoders (VAEs) rely on a probabilistic generative framework: the decoder explicitly models a conditional distribution $p_\theta(\mathbf{x}|\mathbf{z})$, predicting both the mean and variance of a Gaussian for each reconstructed feature. This allows the model to capture uncertainty and adjust its reconstruction confidence depending on the input. The reconstruction error is therefore derived from the full negative log-likelihood of this distribution, which includes both a squared error term and a variance-dependent penalty. The distributions per cell type reveal how well each population is represented in the latent space: narrow, left-skewed distributions indicate faithful and consistent reconstruction, while broader or right-shifted ones may reflect intra-class heterogeneity, underrepresentation, or biologically rare phenotypes that are harder to model.

As shown in Figure 4.14A, the model fails to accurately reconstruct leukemic cells, which were not included in the training set. These cells are naturally mapped to a different cluster, since the model was trained using only healthy cell populations, and the number of clusters was fixed accordingly. The distribution of their negative log-likelihood is broader and more right-skewed compared to other cell types, highlighting the model's inability to reconstruct them faithfully. This suggests that leukemic cells deviate significantly from the learned manifold of healthy cells, making them appear anomalous from the model's perspective.

Figure 4.14B demonstrates the model's ability to accurately reconstruct MRD cells when they are included in the training set. In this setting, the reconstruction loss distribution for MRD cells is narrow and left-skewed, comparable to that of the other cell types, indicating that the model successfully captures their underlying structure. This result confirms that the model is in fact capable of representing and reconstructing these cells when it has been exposed to them during training. Therefore, its failure to do so in the unseen setting cannot be attributed to an intrinsic limitation, such as phenotypic heterogeneity or underrepresentation (0.15% of dataset), but rather to the absence of these cells from the training data.

In conclusion, MARVIN can be effectively used as an anomaly detector for cellular populations. By training the model on a reference cohort of "typical" patients, it can subsequently identify atypical populations in new patients, such as deviant or abnormal cell types, based on their reconstruction profiles. In particular, anomalous populations tend to exhibit broader and right-skewed distributions of the negative log-likelihood, reflecting poor reconstruction and thus deviation from the learned latent structure. One could also consider using the mean reconstruction loss as a quick diagnostic signal: for instance, in Figure 4.14A, the mean of the reconstruction loss for leukemic cells (i.e., $-\log p_\theta(\mathbf{x}|\mathbf{z})$) is markedly higher, around 20, compared to the range of approximately $[-6, 0]$ observed for well-represented healthy populations.

## 4.6 Comparison with Classical VAE

We have previously motivated, both theoretically and intuitively, the importance of structuring the latent representations using a cell-type variable $c$. This justifies the use of a conditional encoder that depends on $c$, as well as a latent prior modeled both as a Gaussian mixture, rather than a standard isotropic Gaussian.

Theoretically, this leads to a more expressive latent prior compared to the classical VAE, where the prior is fixed with a simple isotropic standard Gaussian. In such a setting, the latent space is forced, via the prior-matching term in the ELBO, to resemble a standard normal distribution, without regard for any underlying structure in the data. As a result, the latent representations are not organized according to meaningful biological variation, such as cell-type identity, but instead are distributed to match a simplistic distribution.

Naturally, our model enables tasks such as cell-type annotation and the discovery of novel subpopulations, and these applications are not feasible with a classical VAE: simply because it does not enable classification, or the specification of supplementary clusters due to the lack of an additional categorical variable $c$. However, we still found it important to compare our framework to a vanilla VAE, particularly in terms of latent space organization and reconstruction performance. Even though the latent space in a classical VAE is unstructured (in a clustering fashion), the decoder is trained to minimize the reconstruction loss. As a result, the decoder may become more powerful, compensating for the encoder's limited capacity, which is constrained by the simplicity of the prior.

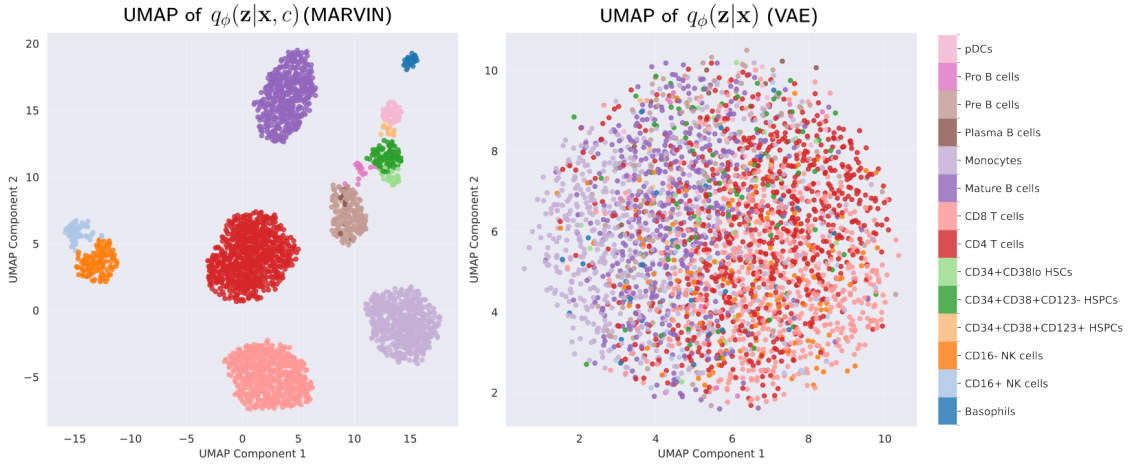



**Figure 4.15: UMAP Latent Representation of $q_\phi(\mathbf{z}|\mathbf{x}, c)$ for MARVIN, and $q_\phi(\mathbf{z}|\mathbf{x})$ for the Vanilla VAE.** The 3000 cells are colored by cell-type assignments established by manual gating (ground truth labels)

**Training Setup.** The vanilla VAE was trained on the AML dataset using the same latent dimensionality, the same training procedure (number of epochs, learning rate, scheduler, etc.), and the same architecture for both the encoder and decoder. This included the same number of residual blocks and the same upscaling factors for the hidden layers, in order to ensure a fair comparison with our model.

To assess the structured nature of MARVIN's latent space, we fully exploited its capabilities by incorporating supervision through cell-type labels. The model was therefore trained in a supervised setting. Figure 4.6 shows the structure of the latent space when no labels were used during training on the AML dataset.

**Results.** As shown in Figure 4.15, the latent representation is clearly more structured in the case of MARVIN, where cells are grouped into distinct clusters corresponding to their respective cell types. This structured space also provides a degree of interpretability: clusters that are close to one another tend to represent phenotypically similar populations. For example, $CD16^+$ and $CD16^-$ NK cells appear close together in the latent space, while being more distant from other, more distinct populations.

In contrast, the latent representation of the vanilla VAE, $q_\phi(\mathbf{z}|\mathbf{x})$ shows no clear cluster structure and instead resembles an isotropic Gaussian distribution, as expected. One can still observe some loose grouping of cells from the same class, indicated by color, but with significant overlap between populations.

When comparing this to Figure 4.6, which displays $q_\phi(\mathbf{z}|\mathbf{x}, c)$ for MARVIN in a fully unsupervised setting, we see that MARVIN still manages to cluster cells of the same type more distinctly than the VAE. Although some overlap remains, it is mostly limited to closely related populations. This suggests that even without direct supervision, the soft assignments produced by $q_\omega(c|\mathbf{x})$ allow MARVIN to structure the latent space more effectively than a standard VAE.

Figure 4.16 qualitatively illustrates the reconstruction capabilities of both models. By visualizing the input data space alongside the reconstructions produced by each model, we can observe how

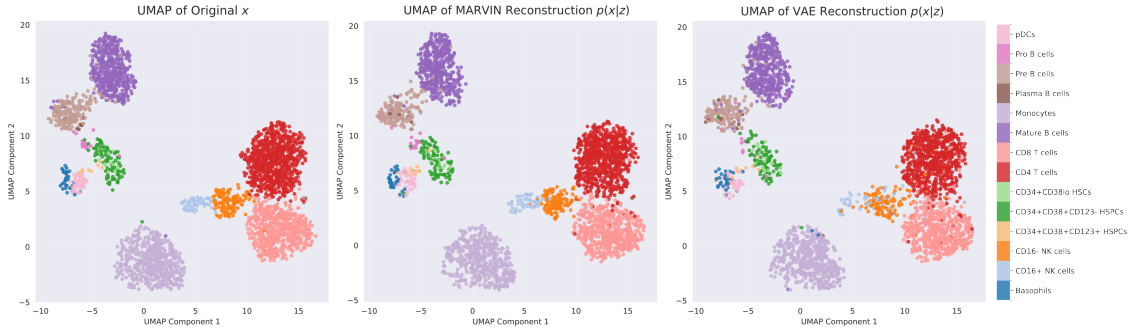



**Figure 4.16: Reconstruction Capabilities of MARVIN vs Vanilla VAE.** The UMAP was fitted on the input space of the 3000 cells, and the same transformation was applied to the reconstructions produced by both models, enabling a reliable comparison.

well the original data are recovered. MARVIN offers more reliable reconstructions, especially for phenotypically similar cell types that differ only in the expression of one or two markers (e.g., NK cells), compared to the vanilla VAE. However, the reconstruction quality achieved by the VAE remains impressive, despite its unstructured latent space. As we had intuitively anticipated, the powerful decoder in the classical VAE compensates for the limited expressiveness of its encoder by still providing reasonably accurate reconstructions, though generally of lower quality than those produced by MARVIN.

In conclusion, the motivation for structuring the latent space extends beyond enabling tasks such as cell-type annotation or subpopulation discovery. It also introduces inductive biases aligned with the intrinsic structure of cytometry data, which is naturally organized into discrete cell populations. This structure ultimately allows for more precise reconstructions of closely related cell types, as the model learns to treat them as distinct, either through supervision signals or simply by specifying a sufficient number of clusters. Furthermore, analyzing the prior distribution $p_\beta(\mathbf{z} \mid c)$, modeled as a Gaussian mixture, provides insights into canonical cell populations, their relative proportions (via sampling from $p_\pi(c)$), and their similarities or separations: capabilities that are absent in the standard VAE. Still, the vanilla VAE remains a strong and simple generative baseline, capable of producing high-quality reconstructions, and could be considered for tasks such as anomaly detection or pure data generation.

# Chapter 5

# Discussion and Perspectives

This chapter serves as the conclusion of this Master's Thesis. It discusses the current limitations of our model, as well as the context in which it will be applied in future research. Description of ongoing collaborations are also presented, providing insight into the model's future integration in real-world settings.

## 5.1 General Discussion

Our model has demonstrated both robustness and originality within its intended application domain. It enables automatic annotation of cell populations, providing a substantial benefit to practitioners by significantly reducing annotation time and improving reproducibility. This is achieved with only a minimal number of labeled examples, thanks to the strength of its semi-supervised learning framework.

At the same time, MARVIN enables the discovery of novel subpopulations, including rare and phenotypically distinct cell types that are often challenging to identify manually. This is made possible through high-dimensional analysis and a structured latent space that captures complex biological variability.

Furthermore, the model supports the analysis of cellular dynamics across different experimental states, paving the way for patient-specific insights and a more personalized interpretation of immune responses and cell population behaviors.

However, the theoretical framework underlying our model can be questioned. The assumption of a structured latent space composed of distinct components, each corresponding to a specific cell population yet all lying on the same level of abstraction, is clearly a simplification. As previously discussed, biological reality is extremely complex, and the relationships between populations (defined as groups of cells with similar phenotypic profiles) are far more entangled than they may initially appear.

It is certainly possible to enrich the theoretical framework by introducing more structured latent spaces, for example using hierarchical representations. This would account for the fact that certain cells share common features before diverging into distinct subpopulations, mirroring a tree-like or hierarchical structure. In truth, the continuum of cell populations is likely even more intricate, but incorporating additional inductive biases into the model architecture could enhance its robustness

and capacity to reflect this biological complexity.

Given that the model is intended for clinical use, thoroughly investigating its interpretability, as well as the clarity of the latent space and classification decisions, is a promising avenue for future work. Improving transparency would ease its adoption in medical contexts and foster greater confidence among practitioners.

## 5.2 Future Work and Perspectives

This work was initially conducted to contribute to a specific phase of a clinical study, which we will describe in more detail below. However, our model fits into a much broader context, and this section also aims to outline its other potential future applications.

### 5.2.1 Liège CHU Clinical Trial

Our model is part of a clinical study led by Dr. Adrien De Voeght as part of his PhD thesis, which aims to demonstrate the benefits of increasing the influenza vaccine dose in immunocompromised patients (suffering from hematological cancers). This study also seeks to investigate a poorly understood phenomenon: in approximately 15% of patients who have undergone treatments intended to completely deplete their immune cell populations (particularly B lymphocytes, by chemotherapy or immunotherapy), humoral immunity is still observed. This is particularly puzzling given the apparent absence of immune cells capable of producing antibodies. It calls for a thorough investigation of the immune landscape: specifically, the cell populations and subpopulations present in patients who show an appropriate response to vaccination despite a severely compromised immune system. The underlying hypotheses will be detailed in the Phase 3 section.

The following sections will provide a detailed overview of the various phases of the clinical study.

#### Phase 1: Vaccination and Analyses

The study includes 180 patients, among whom 140 are immunocompromised and 40 are healthy individuals over the age of 65. The trial is randomized: one group of patients receives the standard vaccine dose, while the other group receives a dose that is four times higher. The central question is whether increasing the vaccine dose leads to a stronger immune response, keeping in mind that immunocompromised patients are not initially expected to mount a significant response.

Peripheral blood samples are collected on day 0 (before vaccination), and on days 1, 7, 28, and 180. Several analyses are performed, including flow cytometry, transcriptomics, signaling pathway profiling (cytokines), serology (quantification of influenza-specific antibody production), as well as functional assays on the patient's immune cells.

Flow cytometry will be performed using the spectral cytometer ID7000, with a panel of over 50 markers (!) developed by KU Leuven. All immune cell subsets will be investigated, including their various activation states, using markers specific to immune activation and immunomodulatory genes (such as CD28, CD38, CTLA-4, HLA-DR, and others).

The resulting data will be complex and high-dimensional, with a substantial number of cell populations to identify and characterize.

#### Phase 2: Systems Biology and Statistical Analyses

This phase focuses on comprehensive analysis by integrating the data collected in the previous phase. Its goal is to characterize the patients' vaccine response using classical statistical models.

#### Phase 3: Data Analysis with MARVIN

The previous phases may not provide a precise answer to the research question: *why do approximately 15% of patients respond to vaccination, even though they are not expected to?* Two main hypotheses could explain this phenomenon:

- in these patients, a rare subpopulation may emerge that is weakly expressed and not easily detectable by standard flow cytometry analyses, or
- some immune cells may have acquired a gain of function, allowing them to contribute to humoral immunity despite not being originally responsible for it.

This is where our generative model comes into play. It may enable the discovery of clearly distinct cellular subpopulations, or subtle differences between cells with a classical phenotype and others with altered activation profiles or gain-of-function characteristics. MARVIN is already ready to be applied directly to cytometry data, and we also aim to adapt it for use with transcriptomic data.

As outlined in the previous section, flow cytometry data will be highly complex due to the large number of markers (>50) and piles of cell populations involved. Our model will greatly facilitate the analysis of such data by enabling automatic annotation of even rare cell populations, with minimal labels required from domain experts. This will significantly reduce the burden on clinicians. In cases where full manual annotation is still performed, MARVIN can assist in post hoc analyses by revealing subpopulations that may have been overlooked during expert-driven gating.

In addition, MARVIN will enable the study of temporal changes in cellular populations across different blood sampling time points: before vaccination and at each subsequent blood sampling. This will support highly detailed analyses at the individual patient level or across patient subgroups (such as vaccine responders and non-responders), helping to capture a more comprehensive and dynamic view of the immune response across all cell populations.

Importantly, any discoveries made by our model will be further validated through follow-up biological experiments, including targeted cytometric or transcriptomic analyses, in order to achieve more precise characterization of the identified subpopulations.

### 5.2.2 MRD Quantification

Beyond its integration into a specific clinical study, our model has a broader field of applicability. It could be used as part of routine workflows for cell-type annotation in clinical or research settings, or for the identification of subpopulations in such contexts.

Furthermore, as discussed in the results Section 4.5, the detection of minimal residual disease (MRD) is critical for diagnosis, treatment planning, and relapse management in patients with hematological cancers.

Our model not only allows for precise characterization of leukemia subtypes, but also enables the detection of rare subpopulations and subtle anomalies in the patient's immune profile. It could therefore serve as a reliable MRD detection tool.

Detecting these rare cancerous cells is essential, as it can directly impact relapse prevention. If left undetected, relapse can have serious clinical consequences, often leading to death. For this reason, we envision positioning our model both as a benchmark for cell annotation in cytometry and as a reliable tool for MRD detection.

To support this, we propose an experiment inspired by real-world clinical scenarios where standard cytometry analysis failed to detect residual disease. In such cases, MRD was only revealed through more advanced and resource-intensive techniques, such as NGS or PCR/qPCR (van Dongen et al., 1999). One could envision a dataset comprising three types of patients: double negatives (undetected by both cytometry and advanced techniques), discordant cases (negative in cytometry but positive in NGS/PCR), and double positives. Among these, the discordant group is particularly compelling, as it represents cases where our model could provide the biggest clinical impact.

If our approach, applied solely to standard cytometry data, is able to detect MRD in patients where manual gating fails and would typically require further in-depth analyses, it would strongly demonstrate its added value. Such an outcome would highlight the model's ability to uncover clinically relevant signals that might otherwise remain hidden in routine analysis pipelines.

# Bibliography


Abdelaal, T., van Unen, V., Höllt, T., Koning, F., Reinders, M. J., & Mahfouz, A. (2019). Predicting cell populations in single cell mass cytometry data. *Cytometry*, *95*(7), 769–781. https://doi.org/10.1002/cyto.a.23738

Aghaeepour, N., Nikolic, R., Hoos, H. H., & Brinkman, R. R. (2011). Rapid cell population identification in flow cytometry data. *Cytometry. Part A : the journal of the International Society for Analytical Cytology*, *79*(1), 6–13. https://doi.org/10.1002/cyto.a.21007

Bandura, D. R., Baranov, V. I., Ornatsky, O. I., Antonov, A., Kinach, R., Lou, X., Pavlov, S., Vorobiev, S., Dick, J. E., & Tanner, S. D. (2009). Mass cytometry: Technique for real time single cell multitarget immunoassay based on inductively coupled plasma time-of-flight mass spectrometry [Publisher: American Chemical Society]. *Analytical Chemistry*, *81*(16), 6813–6822. https://doi.org/10.1021/ac901049w

Bendall, S. C., Nolan, G. P., Roederer, M., & Chattopadhyay, P. K. (2012). A deep profiler's guide to cytometry. *Trends in Immunology*, *33*(7), 323–332. https://doi.org/10.1016/j.it.2012.02.010

Bendall, S. C., Simonds, E. F., Qiu, P., Amir, E.-a. D., Krutzik, P. O., Finck, R., Bruggner, R. V., Melamed, R., Trejo, A., Ornatsky, O. I., Balderas, R. S., Plevritis, S. K., Sachs, K., Pe'er, D., Tanner, S. D., & Nolan, G. P. (2011). Single-cell mass cytometry of differential immune and drug responses across a human hematopoietic continuum. *Science (New York, N.y.)*, *332*(6030), 687–696. https://doi.org/10.1126/science.1198704

Bini, L., Mojarrad, F. N., Matthes, T., & Marchand-Maillet, S. (2024, February 28). HemaGraph: Breaking barriers in hematologic single cell classification with graph attention. https://doi.org/10.48550/arXiv.2402.18611

Blampey, Q., Bercovici, N., Dutertre, C.-A., Pic, I., André, F., Ribeiro, J. M., & Cournède, P.-H. (2023, April 21). A biology-driven deep generative model for cell-type annotation in cytometry. https://doi.org/10.1093/bib/bbad260

Buckland, M., & Gey, F. (1994). The relationship between recall and precision. [Publisher: Wiley-Blackwell]. *Journal of the American Society for Information Science*, *45*(1), 12–19. https://doi.org/10.1002/(SICI)1097-4571(199401)45:1<12::AID-ASI2>3.0.CO;2-L

Cardenas, A., Fadadu, R. P., & Koppelman, G. H. (2023). Epigenome-wide association studies of allergic disease and the environment. *The Journal of allergy and clinical immunology*, *152*(3), 582–590. https://doi.org/10.1016/j.jaci.2023.05.020

Chinthrajah, R. S., Purington, N., Andorf, S., Long, A., O'Laughlin, K. L., Lyu, S. C., Manohar, M., Boyd, S. D., Tibshirani, R., Maecker, H., Plaut, M., Mukai, K., Tsai, M., Desai, M., Galli, S. J., & Nadeau, K. C. (2019). Sustained outcomes in a large double-blind, placebo-controlled, randomized phase 2 study of peanut immunotherapy. *Lancet (London, England)*, *394*(10207), 1437–1449. https://doi.org/10.1016/S0140-6736(19)31793-3

Dilokthanakul, N., Mediano, P. A. M., Garnelo, M., Lee, M. C. H., Salimbeni, H., Arulkumaran, K., & Shanahan, M. (2017, January 13). Deep unsupervised clustering with gaussian mixture variational autoencoders. https://doi.org/10.48550/arXiv.1611.02648

Drescher, H., Weiskirchen, S., & Weiskirchen, R. (2021). Flow cytometry: A blessing and a curse [Number: 11 Publisher: Multidisciplinary Digital Publishing Institute]. *Biomedicines*, *9*(11), 1613. https://doi.org/10.3390/biomedicines9111613

Finak, G., Langweiler, M., Jaimes, M., Malek, M., Taghiyar, J., Korin, Y., Raddassi, K., Devine, L., Obermoser, G., Pekalski, M. L., Pontikos, N., Diaz, A., Heck, S., Villanova, F., Terrazzini, N., Kern, F., Qian, Y., Stanton, R., Wang, K., . . . McCoy, J. P. (2016). Standardizing flow cytometry immunophenotyping analysis from the human ImmunoPhenotyping consortium. *Scientific Reports*, *6*, 20686. https://doi.org/10.1038/srep20686

Grammer, A. C., Fischer, R., Lee, O., Zhang, X., & Lipsky, P. E. (2004). Flow cytometric assessment of the signaling status of human b lymphocytes from normal and autoimmune individuals. *Arthritis Res Ther*, *6*(1), 28. https://doi.org/10.1186/ar1155

Grønbech, C. H., Vording, M. F., Timshel, P. N., Sønderby, C. K., Pers, T. H., & Winther, O. (2020). scVAE: Variational auto-encoders for single-cell gene expression data. *Bioinformatics*, *36*(16), 4415–4422. https://doi.org/10.1093/bioinformatics/btaa293

He, K., Zhang, X., Ren, S., & Sun, J. (2015, December 10). Deep residual learning for image recognition. https://doi.org/10.48550/arXiv.1512.03385

Heuser, M., Freeman, S. D., Ossenkoppele, G. J., Buccisano, F., Hourigan, C. S., Ngai, L. L., Tettero, J. M., Bachas, C., Baer, C., Béné, M.-C., Bücklein, V., Czyz, A., Denys, B., Dillon, R., Feuring-Buske, M., Guzman, M. L., Haferlach, T., Han, L., Herzig, J. K., . . . Cloos, J. (2021). 2021 update on MRD in acute myeloid leukemia: A consensus document from the european LeukemiaNet MRD working party. *Blood*, *138*(26), 2753–2767. https://doi.org/10.1182/blood.2021013626

Ji, D., Nalisnick, E., Qian, Y., Scheuermann, R. H., & Smyth, P. (2018). Bayesian trees for automated cytometry data analysis [ISSN: 2640-3498]. *Proceedings of the 3rd Machine Learning for Healthcare Conference*, 465–483. Retrieved April 10, 2025, from https://proceedings.mlr.press/v85/ji18a.html

Kalina, T. (2020). Reproducibility of flow cytometry through standardization: Opportunities and challenges [_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cyto.a.23901]. *Cytometry Part A*, *97*(2), 137–147. https://doi.org/10.1002/cyto.a.23901

Kaushik, A., Dunham, D., He, Z., Manohar, M., Desai, M., Nadeau, K. C., & Andorf, S. (2021). *CyAnno* : A semi-automated approach for cell type annotation of mass cytometry datasets (J. Xu, Ed.). *Bioinformatics*, *37*(22), 4164–4171. https://doi.org/10.1093/bioinformatics/btab409

Kingma, D. P., & Ba, J. (2017, January 30). Adam: A method for stochastic optimization. https://doi.org/10.48550/arXiv.1412.6980

Kingma, D. P., Rezende, D. J., Mohamed, S., & Welling, M. (2014, October 31). Semi-supervised learning with deep generative models. https://doi.org/10.48550/arXiv.1406.5298

Kingma, D. P., & Welling, M. (2022, December 10). Auto-encoding variational bayes. https://doi.org/10.48550/arXiv.1312.6114

Lee, H.-C., Kosoy, R., Becker, C. E., Dudley, J. T., & Kidd, B. A. (2017). Automated cell type discovery and classification through knowledge transfer. *Bioinformatics*, *33*(11), 1689–1695. https://doi.org/10.1093/bioinformatics/btx054

Lemieux, A., Sannier, G., Nicolas, A., Nayrac, M., Delgado, G.-G., Cloutier, R., Brassard, N., Laporte, M., Duchesne, M., Sreng Flores, A. M., Finzi, A., Tastet, O., Dubé, M., & Kaufmann, D. E. (2024). Enhanced detection of antigen-specific t cells by a multiplexed AIM assay. *Cell Reports Methods*, *4*(1), 100690. https://doi.org/10.1016/j.crmeth.2023.100690

Levine, J. H., Simonds, E. F., Bendall, S. C., Davis, K. L., Amir, E.-a. D., Tadmor, M. D., Litvin, O., Fienberg, H. G., Jager, A., Zunder, E. R., Finck, R., Gedman, A. L., Radtke, I., Downing, J. R., Pe'er, D., & Nolan, G. P. (2015). Data-driven phenotypic dissection

of AML reveals progenitor-like cells that correlate with prognosis. *Cell*, *162*(1), 184–197. https://doi.org/10.1016/j.cell.2015.05.047

Li, H., Shaham, U., Stanton, K. P., Yao, Y., Montgomery, R. R., & Kluger, Y. (2017). Gating mass cytometry data by deep learning. *Bioinformatics*, *33*(21), 3423–3430. https://doi.org/10.1093/bioinformatics/btx448

Liu, P., Liu, S., Fang, Y., Xue, X., Zou, J., Tseng, G., & Konnikova, L. (2020). Recent advances in computer-assisted algorithms for cell subtype identification of cytometry data. *Frontiers in Cell and Developmental Biology*, *8*, 234. https://doi.org/10.3389/fcell.2020.00234

Loshchilov, I., & Hutter, F. (2019, January 4). Decoupled weight decay regularization. https://doi.org/10.48550/arXiv.1711.05101

Maaten, L. v. d., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, *9*(86), 2579–2605. Retrieved June 7, 2025, from http://jmlr.org/papers/v9/vandermaaten08a.html

McInnes, L., Healy, J., & Melville, J. (2020, September 18). UMAP: Uniform manifold approximation and projection for dimension reduction. https://doi.org/10.48550/arXiv.1802.03426

McKinnon, K. M. (2018). Flow cytometry: An overview. *Current Protocols in Immunology*, *120*(1). https://doi.org/10.1002/cpim.40

Nguyen, H. H., Nguyen, C. N., Dao, X. T., Duong, Q. T., Kim, D. P. T., & Pham, M.-T. (2024, August 24). Variational autoencoder for anomaly detection: A comparative study [version: 1]. https://doi.org/10.48550/arXiv.2408.13561

Nowicka, M., Krieg, C., Crowell, H. L., Weber, L. M., Hartmann, F. J., Guglietta, S., Becher, B., Levesque, M. P., & Robinson, M. D. (2019). CyTOF workflow: Differential discovery in high-throughput high-dimensional cytometry datasets. *F1000Research*, *6*, 748. https://doi.org/10.12688/f1000research.11622.3

Park, L. M., Lannigan, J., & Jaimes, M. C. (2020). OMIP-069: Forty-color full spectrum flow cytometry panel for deep immunophenotyping of major cell subsets in human peripheral blood. *Cytometry*, *97*(10), 1044–1051. https://doi.org/10.1002/cyto.a.24213

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., ... Chintala, S. (2019, December 3). *PyTorch: An imperative style, high-performance deep learning library* [arXiv.org]. Retrieved May 23, 2025, from https://arxiv.org/abs/1912.01703v1

Prasad, V., Das, D., & Bhowmick, B. (2020, May 10). Variational clustering: Leveraging variational autoencoders for image clustering. https://doi.org/10.48550/arXiv.2005.04613

Prince, S. J. (2023). *Understanding deep learning*. The MIT Press. http://udlbook.com

Qiu, P., Simonds, E. F., Bendall, S. C., Gibbs, K. D., Bruggner, R. V., Linderman, M. D., Sachs, K., Nolan, G. P., & Plevritis, S. K. (2011). Extracting a cellular hierarchy from high-dimensional cytometry data with SPADE. *Nature biotechnology*, *29*(10), 886–891. https://doi.org/10.1038/nbt.1991

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. (2021, February 26). Learning transferable visual models from natural language supervision. https://doi.org/10.48550/arXiv.2103.00020

Shu, R. (2016). Gaussian mixture vae. https://ruishu.io/2016/12/25/gmvae/

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*(56), 1929–1958. Retrieved April 7, 2025, from http://jmlr.org/papers/v15/srivastava14a.html

Staats, J., Divekar, A., McCoy, J. P., & Maecker, H. T. (2019). Guidelines for gating flow cytometry data for immunological assays. In J. P. McCoy Jr (Ed.), *Immunophenotyping: Methods*

*and protocols* (pp. 81–104). Springer New York. https://doi.org/10.1007/978-1-4939-9650-6_5

Teney, D., Nicolicioiu, A., Hartmann, V., & Abbasnejad, E. (2025, April 29). Neural redshift: Random networks are not random functions. https://doi.org/10.48550/arXiv.2403.02241

Traag, V., Waltman, L., & Eck, N. J. v. (2019). From louvain to leiden: Guaranteeing well-connected communities. *Scientific Reports*, *9*(1), 5233. https://doi.org/10.1038/s41598-019-41695-z

van Dongen, J. J. M., Macintyre, E. A., Gabert, J. A., Delabesse, E., Rossi, V., Saglio, G., Gottardi, E., Rambaldi, A., Dotti, G., Griesinger, F., Parreira, A., Gameiro, P., Diáz, M. G., Malec, M., Langerak, A. W., San Miguel, J. F., & Biondi, A. (1999). Standardized RT-PCR analysis of fusion gene transcripts from chromosome aberrations in acute leukemia for detection of minimal residual disease [Publisher: Nature Publishing Group]. *Leukemia*, *13*(12), 1901–1928. https://doi.org/10.1038/sj.leu.2401592

Zhai, X., Mustafa, B., Kolesnikov, A., & Beyer, L. (2023, September 27). Sigmoid loss for language image pre-training. https://doi.org/10.48550/arXiv.2303.15343

# Appendix A

# KL divergence between two Gaussian distributions

Considering two distributions

$$q(\mathbf{x}) = \mathcal{N}(\mu_q, \sigma_q^2), \qquad p(\mathbf{x}) = \mathcal{N}(\mu_p, \sigma_p^2),$$

the KL divergence between them writes:

$$\mathrm{KL}(q(\mathbf{x})\|p(\mathbf{x})) = \int q(\mathbf{x}) \log \frac{q(\mathbf{x})}{p(\mathbf{x})} d\mathbf{x}. \tag{A.1}$$

For two Gaussian distributions, $\log \frac{q(\mathrm{x})}{p(\mathrm{x})}$ can be formulated as:

$$\log \frac{q(\mathbf{x})}{p(\mathbf{x})} = \log \left( \frac{\frac{1}{\sqrt{2\pi\sigma_q^2}} \exp\left(-\frac{(\mathrm{x}-\mu_q)^2}{2\sigma_q^2}\right)}{\frac{1}{\sqrt{2\pi\sigma_p^2}} \exp\left(-\frac{(\mathrm{x}-\mu_p)^2}{2\sigma_p^2}\right)} \right) \tag{A.2}$$

$$= \log \frac{\sigma_p}{\sigma_q} + \left( -\frac{(\mathbf{x}-\mu_q)^2}{2\sigma_q^2} + \frac{(\mathbf{x}-\mu_p)^2}{2\sigma_p^2} \right). \tag{A.3}$$

Plugging it back into Eq. A.1,

$$\mathrm{KL}(q(\mathbf{x})\|p(\mathbf{x})) = \int q(\mathbf{x}) \left( \log \frac{\sigma_p}{\sigma_q} + \left( -\frac{(\mathbf{x}-\mu_q)^2}{2\sigma_q^2} + \frac{(\mathbf{x}-\mu_p)^2}{2\sigma_p^2} \right) \right) d\mathbf{x}. \tag{A.4}$$

The first term $\log \frac{\sigma_p}{\sigma_q}$ is constant.

The second term gives

$$\int q(\mathbf{x}) \left( -\frac{(\mathbf{x}-\mu_q)^2}{2\sigma_q^2} \right) d\mathbf{x} = -\frac{1}{2\sigma_q^2} \cdot \sigma_q^2 = -\frac{1}{2}, \tag{A.5}$$

because $\mathbb{E}_{q(\mathrm{x})}\left[(\mathbf{x}-\mu_q)^2\right] = \sigma_q^2$. The third term evaluates to

$$\int q(\mathbf{x}) \left( \frac{(\mathbf{x}-\mu_p)^2}{2\sigma_p^2} \right) d\mathbf{x} = \frac{1}{2\sigma_p^2} \mathbb{E}_{q(\mathrm{x})}\left[(\mathbf{x}-\mu_p)^2\right]. \tag{A.6}$$

The expectation can be developed:

$$\mathbb{E}_{q(\mathbf{x})}\left[(\mathbf{x}-\mu_p)^2\right] = \mathbb{E}_{q(\mathbf{x})}\left[((\mathbf{x}-\mu_q)+(\mu_q-\mu_p))^2\right] \tag{A.7}$$

$$= \underbrace{\mathbb{E}_{q(\mathbf{x})}\left[(\mathbf{x}-\mu_q)^2\right]}_{=\sigma_q^2} + 2(\mu_q-\mu_p)\underbrace{\mathbb{E}_{q(\mathbf{x})}\left[(\mathbf{x}-\mu_q)\right]}_{=0} + (\mu_q-\mu_p)^2 \tag{A.8}$$

$$= \sigma_q^2 + (\mu_q-\mu_p)^2. \tag{A.9}$$

So in fine,

**KL divergence between two Gaussian distributions**

$$\mathrm{KL}(q(\mathbf{x})\|p(\mathbf{x})) = \frac{1}{2}\log\frac{\sigma_p^2}{\sigma_q^2} + \frac{(\sigma_q^2+(\mu_q-\mu_p)^2)}{2\sigma_p^2} - \frac{1}{2}.$$

# Appendix B

# Annotation Performance

## B.1 Per-cell-type metrics on the benchmark datasets, in comparison with Scyan

This appendix provides a more detailed quantification of the classification performance of our model vs. Scyan. Rather than relying solely on global metrics, we report per-cell-type metrics, along with the use of confusion matrices to highlight which cell types are most frequently misclassified and to better understand error patterns across the models.

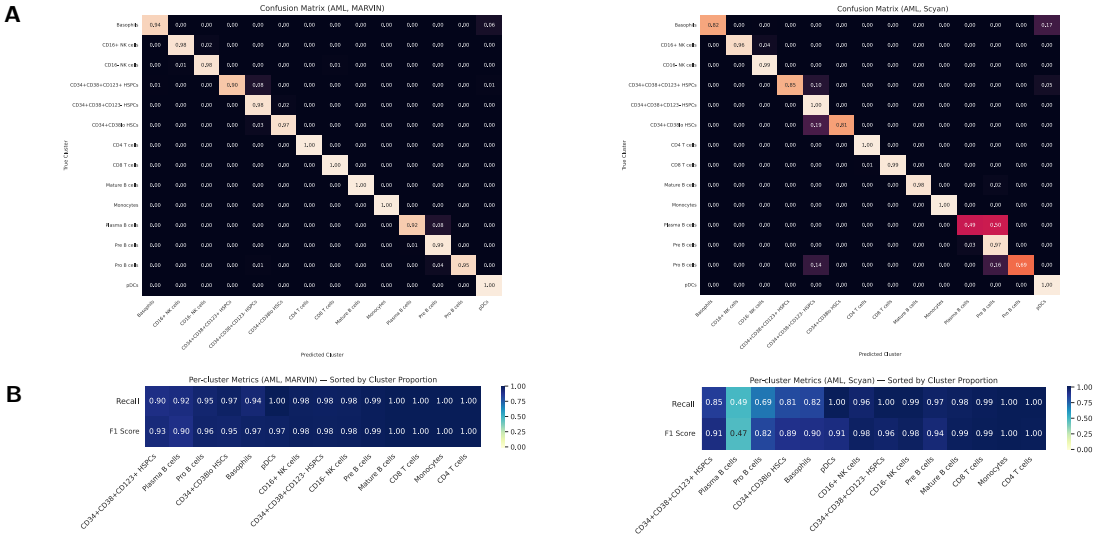



**Figure B.1: Per-cell-type metrics on AML Dataset, MARVIN vs Scyan. A.** Confusion matrices of annotations compared to manual gating. **B.** Recall and F1-score per-cell-type.

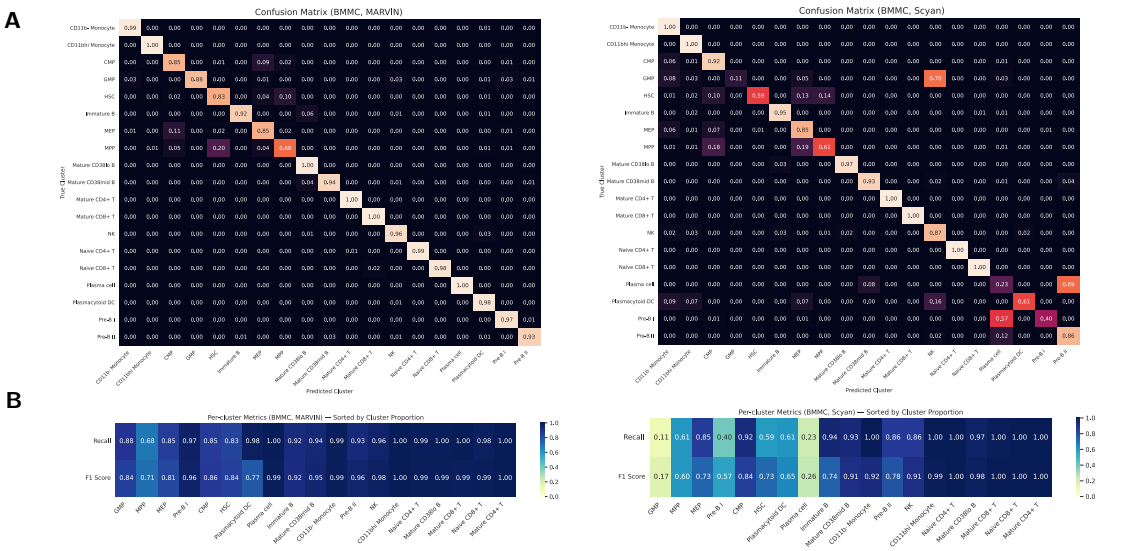



**Figure B.2: Per-cell-type metrics on BMMC Dataset, MARVIN vs Scyan. A.** Confusion matrices of annotations compared to manual gating. **B.** Recall and F1-score per-cell-type.

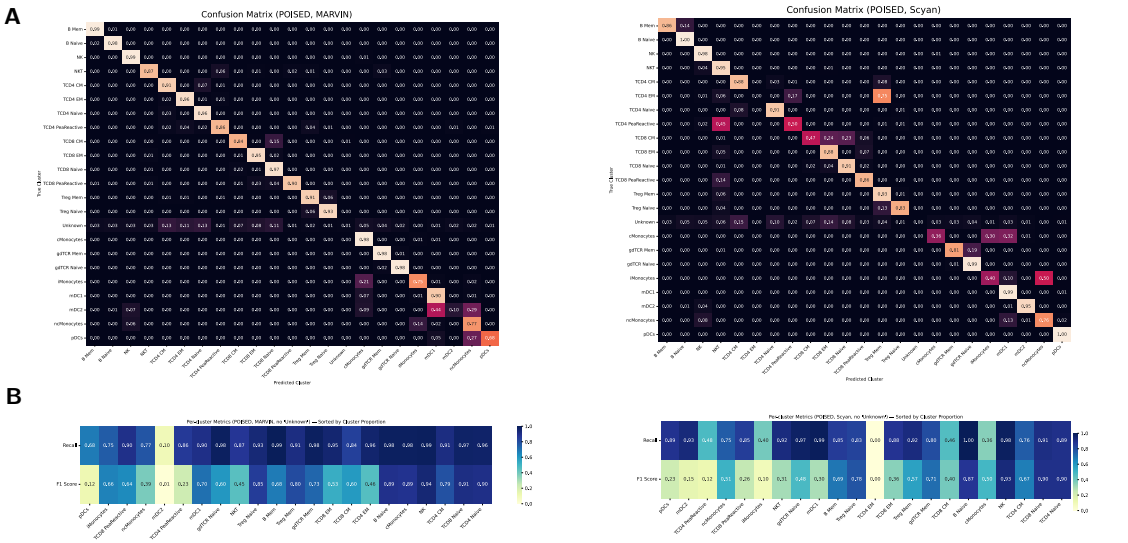
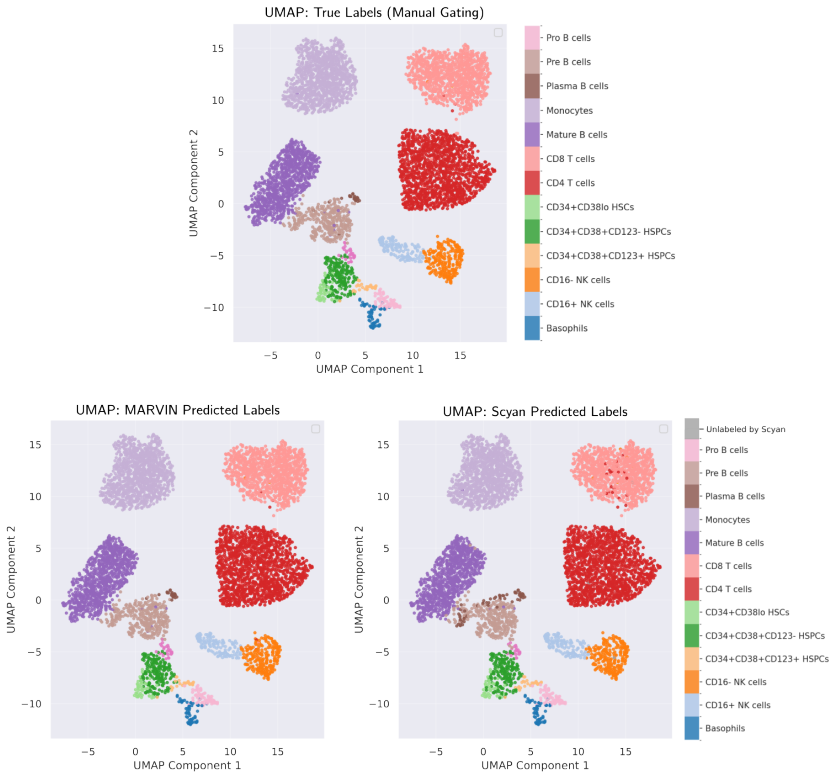


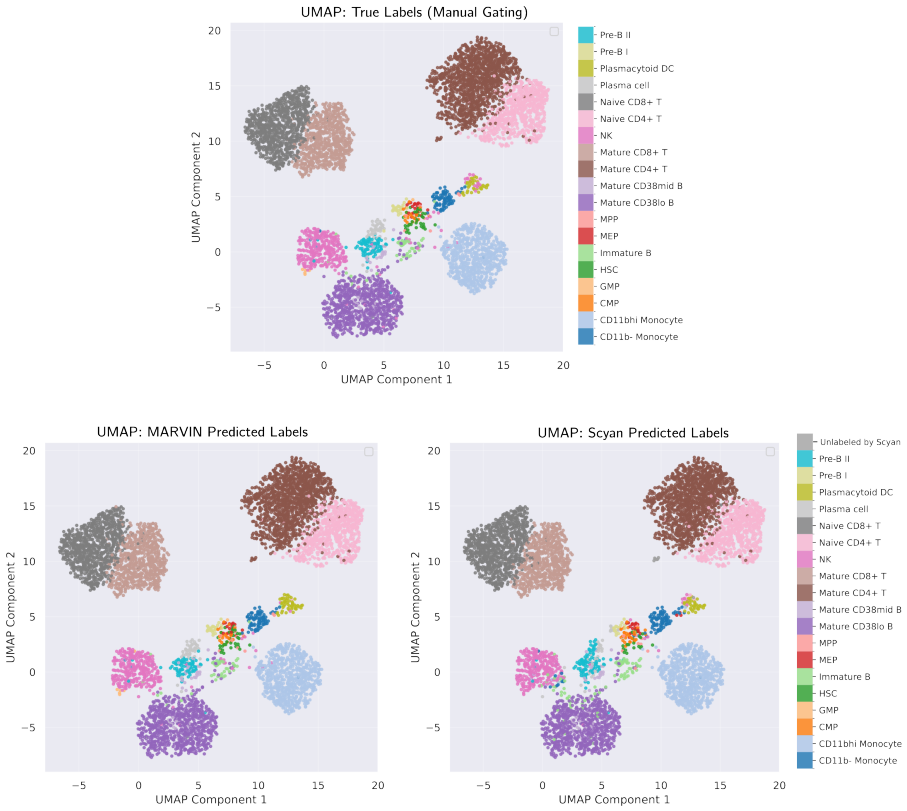

**Figure B.3: Per-cell-type metrics on POISED Dataset, MARVIN vs Scyan. A.** Confusion matrices of annotations compared to manual gating. **B.** Recall and F1-score per-cell-type.

## B.2 UMAP Display of benchmark datasets, with annotations

These representations of the cell annotations in input space, produced by both models and compared to manual gating, provide a visual assessment of their annotation performance. They also help identify which cell types, due to their proximity in the data space, are phenotypically similar and therefore more challenging to classify.




**(A) AML dataset.**




**(B) BMMC dataset.**

**Figure B.4: UMAP (McInnes et al., 2020) visualization of two benchmark datasets.** Each panel shows 3000 cells colored by cell-type annotations from manual gating, MARVIN, and Scyan.

## B.3 Additional Figures for the Latent Representations

### B.3.1 Variational Posterior vs Latent Prior

Analyzing the differences between the variational posterior $q_\phi(\mathbf{z} \mid \mathbf{x}, c)$ and the latent prior $p_\beta(\mathbf{z} \mid c)$, which are jointly optimized in the ELBO to remain aligned, offers valuable insights. The prior $p_\beta(\mathbf{z} \mid c)$ represents the model's global belief about the distribution of latent representations for each cell type, learned across the entire dataset. In contrast, $q_\phi(\mathbf{z} \mid \mathbf{x}, c)$ encodes the latent representation of specific input batches (e.g., a single patient's cells or a subset of the data).

By comparing the posterior and the prior, we can assess how a particular batch or patient deviates from the model's overall expectations. This can help identify shifts in latent representations that reflect differences in phenotype or activation state. Similarly, comparing the inferred categorical distribution $q_\omega(c \mid \mathbf{x})$ to the prior class distribution $p_\pi(c)$ allows us to detect changes in cell population proportions. Together, these analyses enable a better understanding of both global structure and sample-specific deviations, which may carry important biological or clinical meaning.

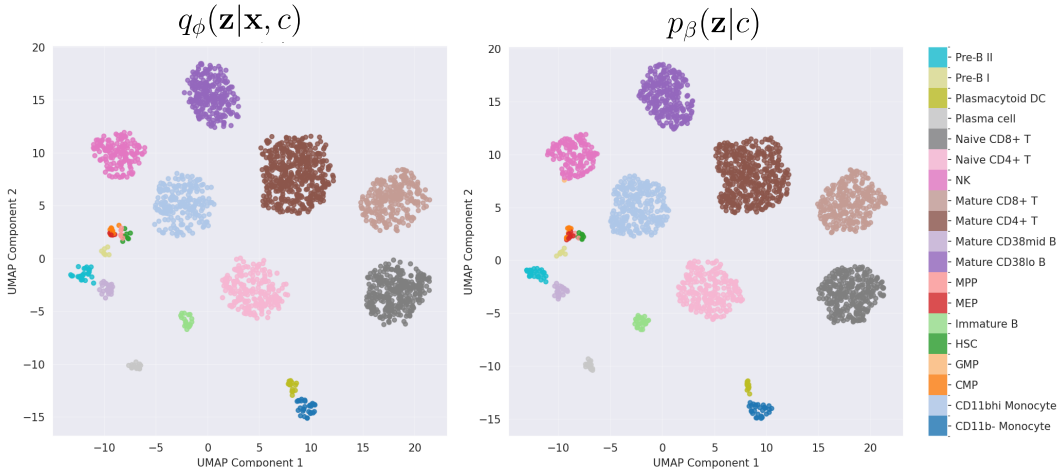



**Figure B.5: UMAP representation of the latent prior $p_\beta(\mathbf{z}|c)$ and the variational posterior $q_\phi(\mathbf{z}|c, \mathbf{x})$, for the BMMC dataset.** A total of 2000 cells were sampled from $c \sim p_\pi(c)$ and used to generate latent samples from the prior $p_\beta(\mathbf{z}|c)$. For the variational posterior, a batch $\mathbf{x}$ of 2000 cells was classified using $c \sim q_\omega(c|\mathbf{x})$, and both $\mathbf{x}$ and $c$ were used to obtain samples from $q_\phi(\mathbf{z}|c, \mathbf{x})$. The UMAP was fitted on the latent prior samples and the same transformation was then applied to the variational posterior samples to allow for direct comparison.

### B.3.2 Pairplot

Since the latent space can be high-dimensional (up to $D = 64$ for more complex datasets), relying solely on dimensionality reduction techniques such as UMAP for interpretation may sometimes oversimplify the structure of the learned representations. To complement this, we propose using a pairplot of the variational posterior $q_\phi(\mathbf{z} \mid \mathbf{x}, c)$ to analyze pairwise relationships between individual latent dimensions.

This approach allows us to directly visualize how well the latent space separates different cell types across each pair of latent variables, without introducing nonlinear projection artifacts. It can help

reveal axes along which specific populations are well separated, and others along which there may still be entanglement. (Similar analyses can also be conducted on the prior $p_\beta(\mathbf{z} \mid c)$).

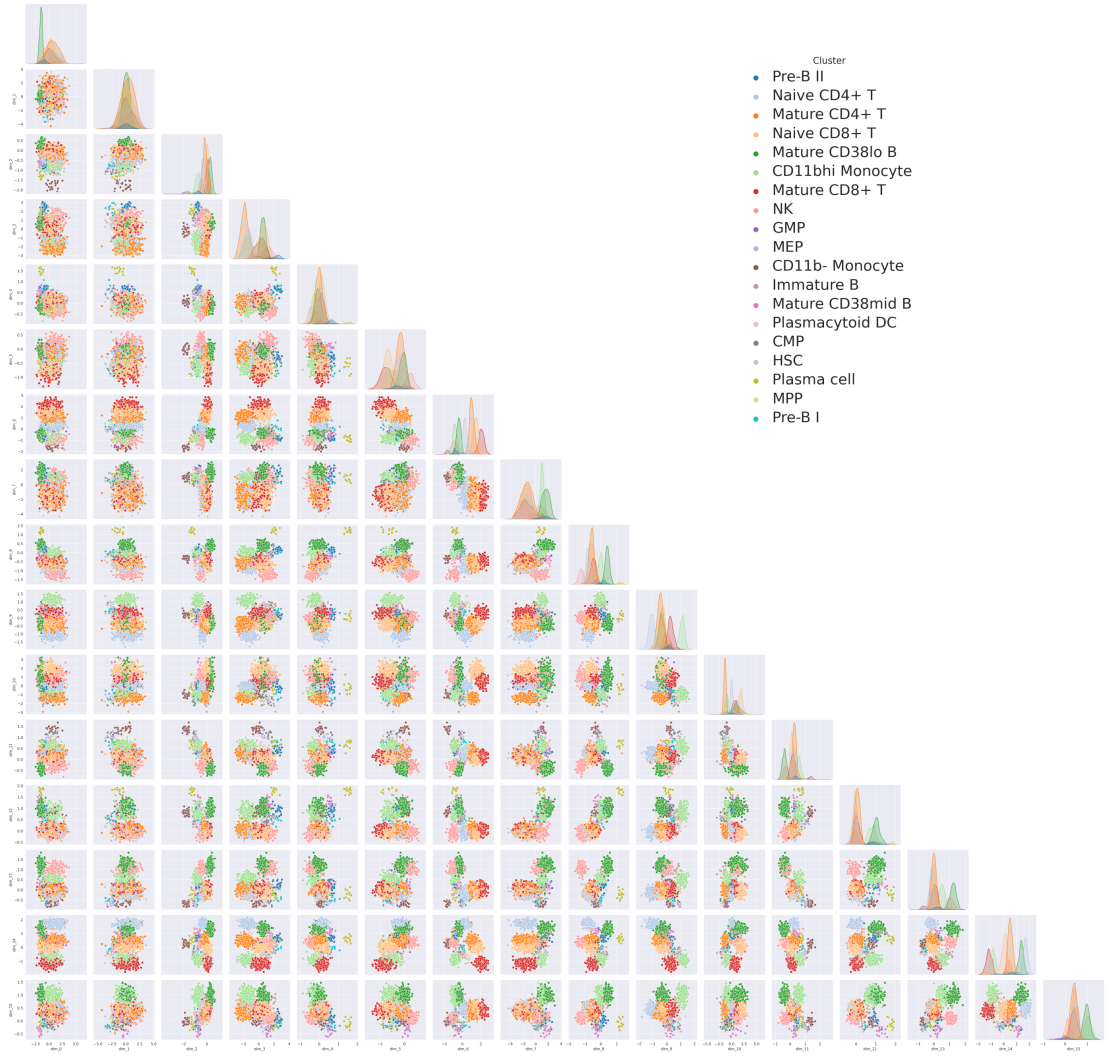



**Figure B.6: Pairplot of the variational distribution $q_\phi(\mathbf{z}|c, \mathbf{x})$, for the BMMC dataset.**

# Appendix C

# Discovery of subpopulations: further analyses

## C.1 Effect of Retaining Ungated Cells on Subpopulation Discovery

This section aims to analyze the subpopulations discovered when Ungated cells are retained in the dataset. These cells account for approximately 24% of the total dataset. It is worth reminding that no information was provided by Chinthrajah et al., 2019 regarding the nature or identity of these unlabeled cells.

The experimental protocol remains the same as in the setting where Ungated cells are removed. Specifically, three additional clusters are allocated for the discovery of new subpopulations, each initialized with a low prior proportion of 0.1% (to reflect the minority aspect of subpopulations). The remaining cell types are assigned their empirical frequencies based on the labeled portion of the dataset. Figure C.1 shows that, in all cases, the discovered subpopulations originate from cells that were initially left ungated by the expert. This can be interpreted by the fact that ungated cells are likely heterogeneous, comprising multiple populations and subpopulations that do not belong to the well-defined cell types specified by manual gating. As a result, the model finds it more beneficial to assign them to the clusters left for subpopulation discovery, rather than isolating the TCD4 Peanut Reactive cells as we had intended. Figure C.2 displays the distribution of cells labeled as Ungated and TCD4 Peanut Reactive by the experts across the discovered clusters. We observe that Ungated cells are diluted across nearly all clusters, with a strong predominance in TCD4 Naive, Central Memory (CM), and TCD8 Naive populations. This observation was also reported by Blampey et al., 2023, the authors of Scyan. However, approximately 34,000 ungated cells were grouped into three distinct subpopulations, as previously discussed, suggesting the potential heterogeneity of the unlabeled cell compartment. A backward analysis would be required to better assess the true identity of these cells that were split into subpopulations.

TCD4 Peanut Reactive cells, on the other hand, were mapped into several known T cell types, including regulatory memory T cells (Tregs), TCD4 Central Memory (CM), Effector Memory (EM), and Naive cells. This behavior is expected: when the model chooses not to isolate these cells into the discovery clusters, because other groups within the ungated compartment are more distinct and thus more "worth" isolating, it instead distributes them among phenotypically similar cell types. This includes CD4+ T lymphocytes (Naive, EM, CM), which share CD4+ expression, as well as regulatory memory T cells. This is particularly plausible in the case of Peanut Reactive

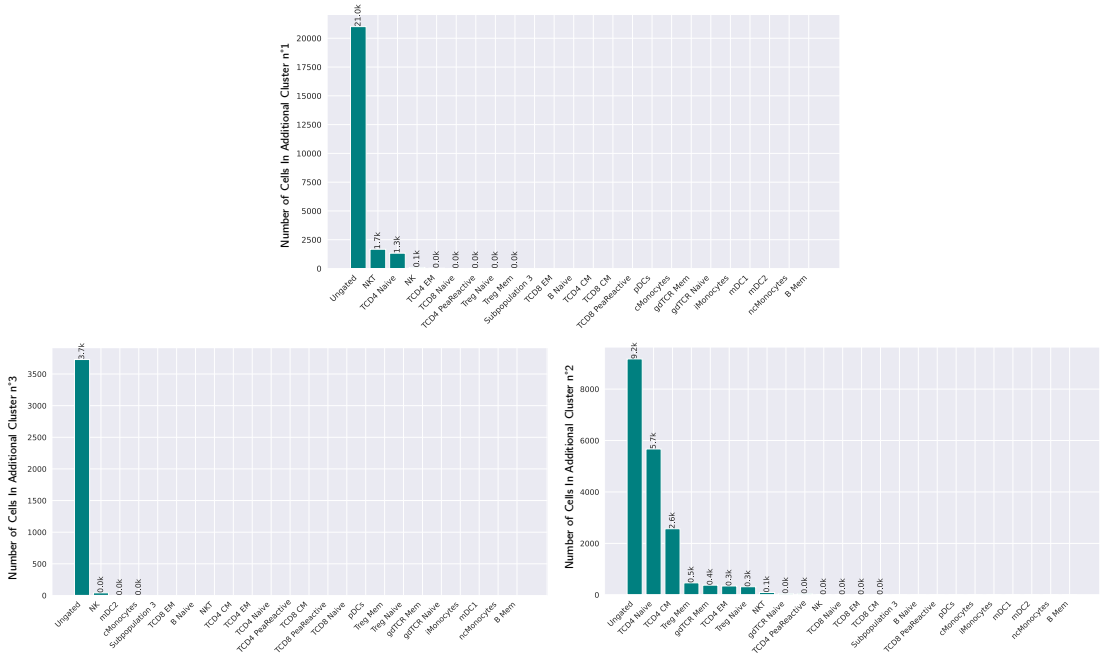
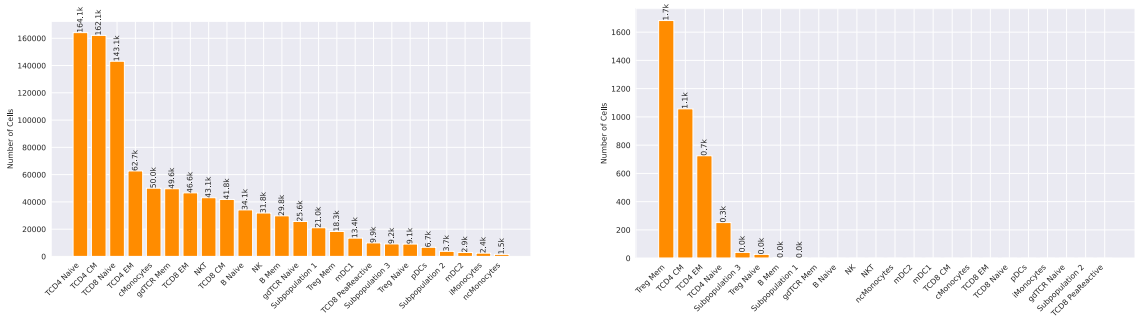



**Figure C.1: The three subpopulations discovered by our model when Ungated cells are retained.**




**(A)** Clusters to which the Ungated cells were mapped by the model

**(B)** Clusters to which the TCD4 PeaReactive cells were mapped by the model

**Figure C.2: Clusters where the Ungated and TCD4 Peanut reactive cells have been mapped.**

cells detected in non-stimulated patients: these cells could be memory T cells specific to peanut antigens, awaiting activation. The fact that Peanut Reactive cells are mapped across several distinct known cell types may also reflect a phenotypic heterogeneity within the expert-gated population itself, possibly depending on activation state or functional status.

When a marker panel is defined, the maximum number of cell populations that can be discovered is inherently limited. For instance, if CD20 or CD19 markers are not included, it becomes impossible to identify B lymphocytes, which are otherwise distinguishable from other cell types based on the expression of these markers (Grammer et al., 2004). With additional information, such as whether debris and doublets have been removed from the ungated cells, it would be possible to estimate the maximal number of identifiable populations and set the number of clusters $K$ accordingly. This could potentially allow the model, by allocating more clusters for subpopulation discovery, to better isolate the TCD4 Peanut Reactive cells of interest.

# Appendix D

# About the Usage of AI

This appendix discusses the use of AI in this master thesis. Obviously, since my deep generative model *is* an artificial intelligence, it is undeniable that this entire work was done with the help of AI (yikes).

Jokes aside, this master's thesis was carried out with the assistance of several AI tools, including ChatGPT, DeepL, and GitHub Copilot. This section serves as a transparency statement regarding the use of these agents, affirming that their involvement was limited to the tasks described below.

ChatGPT and DeepL were used solely to *translate* certain parts of the manuscript, sometimes rephrasing sections, but never to generate original text by themselves. The ideas presented in this thesis are my own; those of my supervisor and co-supervisor, the literature review, and the insightful discussions with some of my friends, but not those of any conversational agent.

Regarding the use of artificial intelligence for coding, GitHub Copilot and ChatGPT have been used primarily to assist in producing aesthetically pleasing figures[1] related to certain experiments, never for the core design of my architecture or model. Thus, MARVIN is the result of my own work and that of my supervisor.

[1] All figures were subsequently refined using Inkscape, which is not an AI tool.