Simulation of biomaterials research experiments with generative models

Veretennikov, Stepan

Award date:
2021

EINDHOVEN UNIVERSITY OF TECHNOLOGY

Department of Mathematics and Computer Science
Data Mining Research Group

Simulation of biomaterials research experiments with generative models

Master Thesis

Stepan Veretennikov

Supervisor:
dr. Vlado Menkovski

Assessment committee:
dr. Vlado Menkovski
dr. Burcu Gumuscu Sefunc
dr. Wouter Meulemans

Eindhoven, October 2021

Abstract

Studying the behavior of living cells in response to the topography of a material's surface is a significant part of biomaterials research. High-throughput screening experiments, in which cells are exposed to a large collection of surface topographies, accumulate a wealth of data on individual-level cell-topography interactions, presenting a challenge to unravel the underlying relationship with the end goal to improve surface topography design. A number of existing approaches to the analysis of screening data use machine learning to model the relationship between surface topography properties and cell response characteristics in either a regression or a classification setting. These approaches, however, do not account for uncertainty, intrinsic to biological experiments. Furthermore, they mostly target a limited number of distinct well-predicted cell features, while a cell response as a whole belongs to a high-dimensional space, hence cannot be predicted by a plain regression model.

In this thesis, we consider the task of modeling the cell-surface topography relationship from the perspective of the generative approach in machine learning. The observed cell response is regarded as an outcome of a generative process, subject to the experiment input conditions, represented by the surface topography. We investigate whether the generative modeling approach can be used to derive a data-driven simulation model of the experiment, such that experiment outcomes are *generated* conditioned on a given topography. The task of topography design is considered inverse to experiment simulation, where topographies that are likely to result in a given (desired) cell response need to be generated. We develop a deep generative simulation model for the case when a cell response is represented by individual cell images, and an image representation of topographies is used. The proposed model serves simultaneously as a simulation model, i.e. allows to generate cell images conditioned on a given topography image, and as a tool for topography design, allowing to generate topography images for a given cell image. The latent space of the proposed model is assumed to be composed of independent latent subspaces, corresponding to specific cell features, which provides for interpretability of the learned representation and allows for cell feature value-conditioned topography design.

We evaluate our model on two datasets, including a synthetic and a real-world dataset. In both cases, a disentangled interpretable latent space is derived for cell images, which allows to generate new cell images with predefined cell properties. Furthermore, the experimental results on the synthetic dataset show that the model is able to learn the embedded relationship between topographies and cells in a probabilistic manner using the disentangled latent space, and based only on the image training data. As a result, the synthetic-case model performs as intended in both application scenarios, generating reliable topography-conditioned cell images and cell-conditioned topography images. The proposed approach exposed a number of limitations due to the assumptions made, however, it provided a valuable example of application of generative modeling in biomaterials research.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Biomaterials engineering

The field of biomaterials engineering is concerned with development of materials that are able to interact with living tissue in a useful and predictable way. Applications of biomaterials vary from hip or dental implants, to heart valves, stents and sutures [50]. Depending on the application, the purpose of a biomaterial is to be compatible with the surrounding tissue, to induce certain behavior of the tissue and to facilitate integration of a medical device into the body. For example, to build bone implants, biomaterials that stimulate integration of an implant into bone tissue are chosen. Hence, for each application, biomaterials inducing specific cell response are sought.

To design biomaterials that are able to induce certain biological response of living cells, the field of biomaterials engineering studies cell behavior in response to different materials. A special case of that research direction is investigating how cells, e.g. mesenchymal stromal cells (MSCs), behave upon exposure to different surfaces produced using the same material. The question investigated is how a 'surface topography', i.e. a surface pattern, impacts the cell response of interest, such as certain biomarker expression or cell shape characteristics.

The goal of such studies is to explore the existing cell behavior-surface topography relationship and, ultimately, to find the optimal surface topography for a given application requiring specific cell response. To solve this task, physical experiments are conducted, in which living cells are exposed to a large collection of different surface topographies. The collected data is subsequently analyzed to reveal the cell-surface topography relationship, which enables one to select or design new surface topographies stimulating the desired cell behavior.

Problem statement

In fact, the task of finding an optimal surface topography given an application consists of two challenges: 1. **Data analysis** to find the "hit" [50] surface topographies which have led to the desired cell response, and to explore the cell response-surface topography relationship; 2. **Topography design**, which involves either selecting perspective topographies from an already existing library or designing new topographies based on the analysis.

Notably, these challenges may be considered as steps in a loop of experiments aiming to converge to the optimal surface topography for a given application. In [52] the authors discuss the concept of an autonomous system for biomaterials discovery, which would be able to analyze the data obtained from physical experiments and to design new materials *in silico*, in an automated fashion, using machine learning (ML) techniques. In this concept, the process of finding the optimal surface topography given an application can be viewed in the from of the following algorithm:

1: Create an initial library of surface topographies
2: **while** cells with desired properties are not found **do**
3: Run physical experiments with cells
4: Analyze the screening data
5: Design new topographies
6: ...
7: **end while**

Hence, the **research problem** is formulated as: Can we leverage machine learning to improve and automate 1. the analysis of the accumulated screening data and 2. the design of new surface topographies, expected to enhance the desired cell response in the next series of experiments?

Machine learning perspective

A number of existing approaches use machine learning to model the relationship between topographies and cells, as further discussed in Chapter 2. Essentially, the modeling approach aims to discover the relationship between two high-dimensional spaces based on the screening data: the space P of parameters defining a surface topography and the space X of cell responses, as shown in Figure 1.1; the screening data is then a collection of pairs $\{p^i, x^i\}_{i=1}^n$. The space X can be, for instance, represented by the space of cell images, however, it may also be regarded in a broader sense and include any observed measures of biological outcome, such as gene expression profiles or other measurements, not captured by cell images.

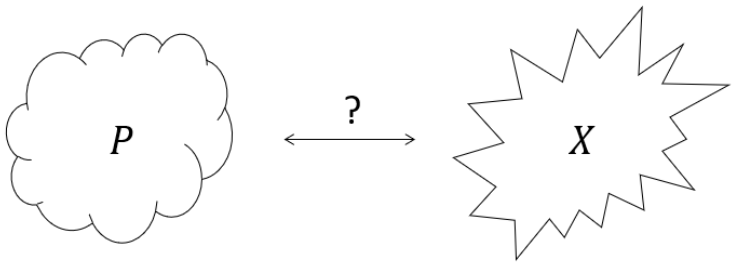

Figure 1.1: P - input parameters space (topography design); X - outcome space (cell response).

From this perspective, modeling of the cell-surface topography interaction is the task to model the mapping $P \to X$, that is, what cell response $x \in X$ can be observed given a surface topography $p \in P$. Whereas the task of topography design is an inverse one, where the mapping $X \to P$ is needed: given a desired cell response x, which surface topographies p could lead to it. Hence, the research problem is reduced to finding a way to model the twofold relationship between the spaces P and X given screening data.

The current approaches to modeling of the cell-surface topography interaction usually involve either regression or classification models that predict some measures of a cell response, such as nucleus form factor or cell orientation [21], based on topography properties. In other words, given some parameter representation of a topography, $p = (p_1, p_2, ..., p_n)$ [53], and the available training data $\{p^i, x^i\}_{i=1}^n$, machine learning models are trained to predict the cell features of interest $f_i(x)$, instead of directly predicting x, as illustrated in Figure 1.2. Then to optimize the cell response $f_i(x)$ in the next iteration of physical experiments, certain topography properties p_j need to be adjusted in a way that suggests improvement of $f_i(x)$ according to the model.

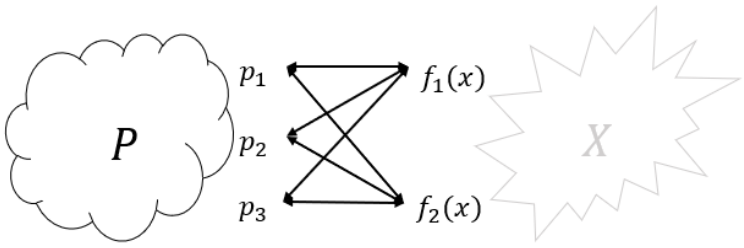

Figure 1.2: Modeling cell-surface topography interaction: current approaches.

The implications of the current approach are that, firstly, such models are point estimate models, which means that they predict a single (expected) value of the cell feature f_i in response to a topography p. Secondly, in the current approach machine learning is essentially used as a tool to analyze data and learn correlations. As a result, only some evident dependencies between topography design parameters and distinct cell features are captured in several independent models.

This observation raises a question of whether it is possible to leverage the advances in machine learning to overcome these limitations and to develop a model capable of learning the complex cell response-surface topography relationship, prone to a high level of uncertainty.

Research questions

As opposed to developing point estimate models, the question investigated in the present work is whether machine learning, and specifically deep generative models (2.2), can be used to develop a data-driven *simulation model* of the cell-surface topography experiment, i.e. a model able to mimic the mapping $P \rightarrow X$ based on screening data. Importantly, such a model should be able to generate a variety of possible cell outcomes x in response to a given topography p, similarly to a real experiment. An illustration is provided in Figure 1.3.

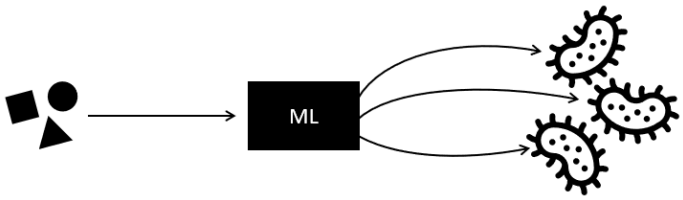

Figure 1.3: *In silico* experiment model

Furthermore, this work investigates whether deep generative models could be also used as tool for biomaterials discovery, i.e. for *in silico* topography design. In this scenario, a model should generate a variety of surface topographies p that could lead to a given cell response x, as shown in Figure 1.4. Desirably, either an observation $x \in X$ could be provided as an input to the model in this scenario, or a value of a cell property of interest $f_i(x)$, corresponding to an observation x.

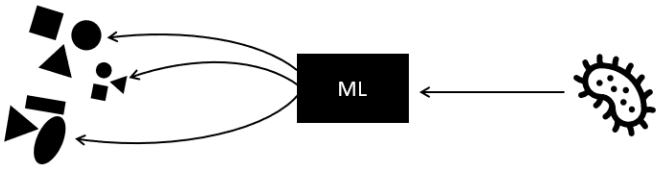

Figure 1.4: *In silico* topography design model

A simulation model of the cell-surface topography experiment could be useful in biomaterials research for several reasons. Firstly, it could be beneficial for an analyst to observe a full cell response x as an output of the model instead of observing a number of predictions of distinct cell features $f_i(x)$. Secondly, only a fraction of the topography design space P can be covered by physical experiments in reasonable time, while an *in silico* experiment model, once trained, can be run multiple times consecutively. As a result, a simulation model of the experiment could help accelerate biomaterials discovery. Furthermore, a simulation model, as formulated above, accounts for uncertainty intrinsic to physical experiments, when a topography leads to a variety of cell responses, and an observed cell response could originate from a variety of topographies.

In the scope of the present work only the case when the outcome space X is represented by the space of *individual cell images* is considered, thereby excluding visually unobserved measures of a cell response. Similarly, an image representation of topographies is used. This decision is motivated by the fact that a visual representation of the outcomes enables one to judge how realistic the generated samples are, and whether they respect the real-world constraints. In the general case, however, an extended cell response representation can be considered, which includes other important measures of cell behavior. Accordingly, alternative topography space representations can be used, such as numerical parameterizations.

The idea behind investigating the applicability of the generative approach to simulation of the experiment is that, firstly, deep generative models are known to be suitable for tasks on high-dimensional datasets, and, most importantly, they allow to incorporate uncertainty. Instead of predicting point estimates of the expected value of cell features, or predicting a single 'typical' cell image in response to a topography, a generative model could be formulated to output a probability distribution $p(x|p)$ of cell images x conditioned on a topography p. Vice versa, a cell image x depicting a cell with certain visual characteristics could potentially originate from a probability distribution $p(p|x)$ of topographies given x. Hence, having a deep generative model for the cell-surface topography relationship allows to reduce the tasks of experiment simulation and of *in silico* topography design to simple queries, where cell or topography images are generated by sampling from the respective distributions $p(x|p)$ and $p(p|x)$.

The **research questions** of the present work are:

1. Can generative models be used to simulate the cell-surface topography experiment?[1] (*in silico* experiment model)

   In silico experiment query: generate different cell images x given a surface topography p.

2. Can generative models be used for *in silico* cell response-conditioned topography design?

   In silico topography design query: generate different topographies p that could lead to a given cell image x.

Proposed approach

In the present work a deep generative model is proposed that is aimed to simultaneously serve as a simulation model of the cell-surface topography experiment and as a tool for *in silico* topography design. The model is designed to generate individual cell images for a given topography image to mimic the physical experiment. Furthermore, the model could be used for topography design by generating topography images for a given cell image, possibly with desired cell properties, or for a given value of the cell feature of interest.

The proposed model is built upon a Variational Autoencoder (VAE) [29] architecture and exploits several techniques to derive a disentangled, or factorized, latent space, particularly inspired by Domain-Invariant Variational Autoencoder (DIVA) [22]. The latent space of cell images is divided into independent latent subspaces, corresponding to cell features of interest, and an additional noise space, corresponding to residual variation in cell images, not explained by these cell features. The key idea behind the proposed approach is to model the twofold cell response-surface topography relationship $P \leftrightarrow X$ in the latent spaces instead of the original spaces. The connection between the two datasets is implemented via shared latent subspaces. The latent subspaces of the cell model corresponding to cell features are assumed to also be part of the latent space in the topography model, with the difference that in the topography model these subspaces represent the factors of variation in topographies that influence the respective cell features. Hence, the proposed model constitutes a two-sided VAE for two related datasets that allows for cross-conditional generation: cell images given a topography image and topography images given a cell image.

Contributions

The contributions of the present work are as follows:

- We introduced a generative modeling perspective on biomaterials research experiments, which addresses the outlined challenges of 1. high dimensionality of the input and output data spaces and 2. high uncertainty of the underlying relationship, inherent to such experiments.

[1] The question of whether machine learning can be used to simulate cell response is in particular mentioned in [53] as an outstanding question in biomaterials research.

- We proposed a deep generative simulation model of the cell-surface topography experiment for the case where both the cell response and the surface topography have an image representation. The model is capable to simulate cell images in response to a given topography image and to simulate topography images in response to a given cell image or a cell feature value.
- We test the proposed architecture on two datasets, including a synthetic and a real-world dataset. In both cases, a factorized and interpretable latent representation of the cell image dataset was derived. Furthermore, the experimental results on the synthetic dataset show that the model is capable to unravel the existing relationship between topographies and cells using the training image data.
- The proposed model constitutes a two-sided variational autoencoder that connects two different datasets, with possibly existing relationship between them, using the concept of a shared latent space, which, to the best of our knowledge, has not been previously considered in the literature.

Thesis outline

In Chapter 2 the current framework of cell-surface topography interaction studies is described, and the current approaches to screening data analysis and to topography design are overviewed. Furthermore, in Chapter 2 theoretical foundations of generative modeling and approaches to deriving disentangled latent representations in generative models are discussed. In Chapter 3 the proposed approach for simulation of the experiment and for *in silico* topography design is described in detail. In Chapter 4 the two datasets used to verify the proposed model are described, and the experimental results for these datasets are provided in Chapter 5. Finally, in Chapter 6 the limitations of the proposed approach and directions for future work are discussed.

Chapter 2

Background

2.1 Studies on cell-surface topography interaction

The idea of surface topographies impacting cell behavior has been verified in the literature with different cell types and with a wide range of cell responses of interest. A number of studies have shown that the topography of the surface in contact with human mesenchymal stem cells (hMSCs) influences cell morphology [50], [20], [35], [3], [21], [4]; expression of certain biomarkers [50], [20], [41], [54], [34]; cell proliferation [50], [41], [4]; metabolic activity [3], [4]. Particularly, it has been demonstrated that certain surface topographies can stimulate differentiation of bone marrow-derived mesenchymal stromal cells (bmMSCs) into bone tissue [20]. Furthermore, surface topography determined secretion levels of different components (cytokines) by bmMSCs and by kidney perivascular mesenchymal stromal cells (kPSCs), thereby controlling their function [35].

2.1.1 Experimental framework

To study the cell-surface topography relationship at scale, the following experimental framework is adopted [50] [21]:

- Surface topography designs are generated by placing random combinations of primitive geometric shapes (topographical features, e.g. displayed in Figure 2.1), in a grid, which constitutes a single topography, examples of which are shown in Figure 2.2;
- The generated surface topographies are produced on a chip, TopoChip [50] (or a plate, TopoWellPlate [3]), using the same material. A single TopoChip contains 2176 unique surface topographies;
- A chip with surface topographies is populated with the cell material and is later stained with fluorescent dyes to highlight the components of interest (cell nucleus, cytoskeleton etc.);
- Finally, the chip undergoes a screening, yielding images that capture the resulting cell response on different topographies. Additionally, properties of individual cells or aggregated cell properties per topography are calculated using special software, CellProfiler [50].

Figure 2.1: Topographical features [50]: examples.

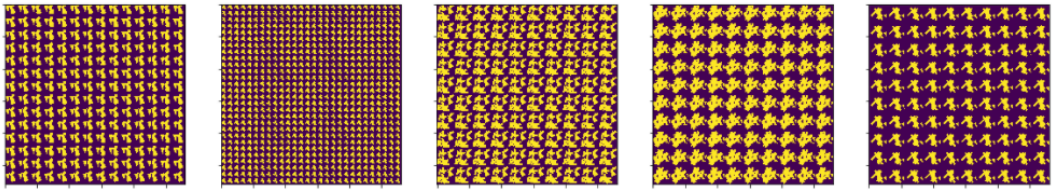

Figure 2.2: Surface topographies [50]: examples.

The outlined procedure provides a collection of observations, where a surface topography description is paired with a corresponding image of cells on top of that topography. Such data captures the relationship between the surface topography properties and the resulting cell properties, which enables one to perform analysis and to search for correlations.

2.1.2 Approaches to screening data analysis and topography design

The challenges of screening data analysis and topography design, outlined in Introduction 1, have been addressed in the literature and multiple approaches were proposed to find correlations between the properties of surface topographies and various cell responses. In [50] the authors identified the topography properties associated with high cell proliferation ratio in a classification setting, where the class of a surface topography was assigned according to the average share of proliferating cells on that topography across a series of experiments. Similarly, in [41] the topography properties associated with cell proliferation and pluripotency biomarker expression were selected, and a logistic regression model was developed to predict the probability of observing pluripotent cells on a topography 24 hours after seeding. In [54] a classification model was developed to predict the level of ICAM-1 biomarker expression based on a topography description. Furthermore, in [3] binary classifiers were trained to predict the assignment of the observed individual cells to five visually identified cell morphology classes based on topography properties.

Whereas in [21] the authors modeled the relationship between the surface topography properties and the resulting cell morphology properties in a regression setting. Cell morphology properties for each topography were calculated as the median value over values for individual cells on the respective topography. Separate models were developed to predict the value of each of the morphological properties, such as nucleus "roundness", cell solidity and others, based on topography properties. Conversely, in [34] a specific topography was predicted based on the resulting morphological properties of cells placed on it. The idea was to identify the topography that could lead to a certain cell morphology, which in turn is associated with expression of particular biomarkers of interest. Furthermore, to visualize the relationship between the surface topographies and the cell response, hierarchical clustering method was used in [34] to group topographies either on the basis of the induced cell morphological features, or according to the gene expression profiles.

In fact, the mentioned papers studying the cell-surface topography relationship share in common that they aim to develop a model of that relationship while analyzing the screening data. The interaction is modeled in either a regression or in a classification setting, where classes or values of the desired cell response are predicted based on topography properties. Such perspective implies that to design new topographies for the next experiment, specific topography properties need to be adjusted in a way that stimulates the desired cell behavior according to the model.

On the other hand, the authors of [51] consider a different perspective where no model of the cell-surface topography relationship is created. At the analysis step, only the best-hit surface topographies are selected. These topographies are subsequently processed with an evolutionary algorithm to produce next-generation topographies. As a result, optimization of the cell response happens by crossover and mutation of the most 'successful' topographies taken as a whole, rather than by combining 'successful' values of specific topography properties.

Hence, the design of improved surface topographies is the end goal of both approaches, however, in the first approach topography design is regarded as an inverse task to modeling the influence of topographies on cell behavior based on experimental data. The modeling approach is further considered throughout this paper, as it particularly aims to unravel the underlying cell-surface topography relationship, apart from providing the end result.

2.2 Generative models in machine learning

Generative modeling approach in machine learning is usually defined on the contrast with a more prevalent discriminative approach. Considering a regression or a classification task with an observation variable x and a target variable y, a discriminative model aims to predict either a conditional probability distribution $p(y|x)$ of the target variable y given an observation x or just a point estimate of y [39]. In other words, a discriminative model only learns to discriminate between the observations to accurately determine the corresponding values or labels of the target variable, ignoring the underlying distribution of the data.

By contrast, a generative model aims to approximate the joint probability distribution $p(x, y)$ of the data and target variables, which allows to infer the data distribution $p(x|y)$ conditioned on a certain class label or on a value. In fact, generative models can also be used to estimate $p(y|x)$ to perform classification or regression tasks like discriminative models. Unlike discriminative models, however, generative models allow to generate new data samples, similar to those in the dataset, and to estimate the likelihood of a given observation. These properties of generative models make them a very popular tool in a variety of problems with high-dimensional structured data such as sequences, images, audio recordings and others. Notably, generative models are often used in an unsupervised setting, when modeling of the probability distribution of the data $p(x)$ or the ability to generate new data samples is the goal itself [46].

To approximate the probability distribution of observations $p(x)$ from some high-dimensional space X, such as a space of images, generative models often assume that there exists a lower-dimensional representation of the data, a latent representation z with some simple probability distribution in the latent space Z_X. In practice, the goal of training a generative model is to train a decoder, or generator, which is a mapping $d_X : Z_X \to X$. A decoder allows to sample from a high-dimensional and complex distribution $p(x)$ by sampling from a low-dimensional and simple distribution $p(z)$ [46], as shown in Figure 2.3.

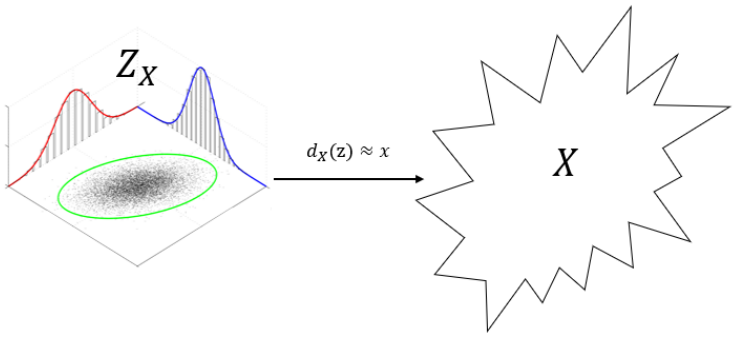

Figure 2.3: A decoder $d_X : Z_X \to X$

The key challenge in training a generative model is to ensure that the distribution of the generated samples $g(z)$ indeed corresponds to the distribution of the data $p(x)$ [46] [12]. The two main approaches in generative modeling, variational autoencoders (VAEs) [29] [43] and generative adversarial networks (GANs) [17], differ in the way they address this challenge. A VAE attempts to invert the decoder with a separate encoder network that maps a sample x from the space X to an approximate posterior distribution $q(z|x)$ in the latent space. Subsequently, a sample from $q(z|x)$ is mapped back to the space X by the decoder, thereby reconstructing the original sample x. Hence, the generated samples in the VAE framework are reconstructions of the original data samples, which is exploited by the training objective to match the distributions $g(z)$ and $p(x)$.

By contrast, a GAN does not use the latent space to match the two distributions [46]. Instead, it employs an auxiliary discriminator network that learns to discern between the generated

samples and the original samples. In turn, the goal of the decoder, or generator, is to mislead the discriminator and generate samples $d_X(z)$ as similar to the original samples as possible, which stimulates the distributions $g(z)$ and $p(x)$ to be close as well.

In the present work the VAE approach will be taken as a basis for a simulation model of the cell-surface topography experiment and will be considered below in more detail. The main feature that makes the VAE approach more applicable in modeling the distribution of the cell response is that it guarantees that each observation x is represented by some region $q(z|x)$ in the latent space. A GAN, however, is known to be susceptible to the mode collapse problem [39] [46], when certain classes of the observed data are not represented in the latent space and thus cannot be generated. Furthermore, owing to the training objective of the VAE, similar cell images are expected to be close in the latent space, while close points in the latent space are expected to represent similar images.

2.2.1 Variational autoencoder

A variational autoencoder (VAE) [29] [43] is a generative model that employs a latent variable model to represent the data distribution. This implies that the distribution of observations $p(x)$ from the original high-dimensional space X can be expressed (2.1) in terms of some latent factor z with a simple distribution in a lower dimensional space Z_X. In (2.1) the probability distribution $p_\theta(x|z)$ is usually parameterized with a neural network with weights θ, which serves as a decoder, for example, $p_\theta(x|z) = N(g(z,\theta), s^2 I)$ [12] is a normal distribution, where g is the decoder neural network, s is a hyperparameter [39] and I is an identity matrix; $p(z)$ is the prior distribution of the latent variable z and is usually assumed to be multivariate normal, i.e. $p(z) = N(0, I)$ [12].

$$p_\theta(x) = \int_{Z_X} p_\theta(x|z)\, p(z)\, dz \tag{2.1}$$

As mentioned above, a VAE has an encoder-decoder architecture, shown in Figure 2.4, where the decoder generates data samples based on the values of the latent variable z, and the encoder embeds data samples x into the latent space.

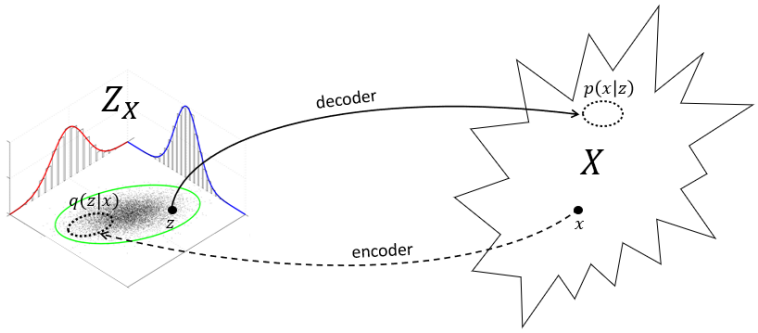

Figure 2.4: An encoder-decoder architecture of a VAE.

The encoder is needed to infer the posterior distribution $p_\theta(z|x)$, which can be expressed as (2.2) by the Bayes' rule. However, the integral in the denominator corresponding to $p_\theta(x)$ (2.1) is intractable [39], and therefore the posterior cannot be calculated directly. Instead, the encoder neural network serves as an approximate posterior $q_\phi(z|x) \approx p_\theta(z|x)$, such that it outputs the distribution parameters of $q_\phi(z|x)$, where ϕ are the weights of the encoder. For example, the approximate posterior can be formulated as $q_\phi(z|x) = N(\mu_\phi(x), \sigma^2_\phi(x) I)$ [12], where the encoder outputs both parameters μ and σ^2 of the normal distribution for a given x.

$$p_\theta(z|x) = \frac{p_\theta(x|z)\, p(z)}{p_\theta(x)} \tag{2.2}$$

The goal in training of a VAE is to maximize the log-likelihood of the training data $\sum_x log\, p_\theta(x)$ given the assumed model (2.1), which is equivalent from the training perspective [39] to maximizing the expected log-likelihood $\mathbb{E}\left[log\, p_\theta(x)\right]$ for an observation, coming from the true data distribution. However, due to intractability of the integral (2.1), training is done by maximizing a lower bound on the log-likelihood, which can be derived as shown below. Notably, the Jensen's inequality is applied since log is a concave function, $q_\phi(z|x) \geq 0$ and $\int q_\phi(z|x)\, dz = 1$.

$$\begin{aligned}
log\, p(x) &= log \int p_\theta(x|z)\, p(z)\, dz \\
&= log \int q_\phi(z|x)\, \frac{p_\theta(x|z)\, p(z)}{q_\phi(z|x)}\, dz \\
&\geq \int q_\phi(z|x)\, log\, \frac{p_\theta(x|z)\, p(z)}{q_\phi(z|x)}\, dz \quad \text{(by the Jensen's inequality)} \\
&= \int q_\phi(z|x)\, log\, p_\theta(x|z)\, dz \; - \int q_\phi(z|x)\, log\, \frac{q_\phi(z|x)}{p(z)}\, dz \\
&= \mathbb{E}_{q_\phi(z|x)}\; log\, p_\theta(x|z) \; - \; KL\left(q_\phi(z|x)\,||\,p(z)\right)
\end{aligned}$$

The derived lower bound (2.3) on the log-likelihood of the data is called the *evidence lower bound* (ELBO). The first term can be seen as the reconstruction error; it reflects the ability of the model to recover an observation x after being passed through the bottleneck of the latent space. Whereas the second term can be seen as a regularization that forces individual approximate posteriors $q_\phi(z|x)$ for different data samples x to be non-deterministic by penalizing them for diverging from the prior distribution $p(z)$ [46]. As a result, the overall "inferred prior" distribution $q(z)$ [32], or "aggregated posterior" [38], combining the individual posteriors resembles the assumed prior $p(z)$ [1], which makes the points in the latent space that correspond to data samples concentrate densely.

$$ELBO = \mathbb{E}_x \left[- \Big(\underbrace{\mathbb{E}_{q_\phi(z|x)}\left[-log\, p_\theta(x|z)\right]}_{\text{Reconstruction error}} + \underbrace{KL\left(q_\phi(z|x)\,||\,p(z)\right)}_{\text{Regularization}} \Big) \right] \tag{2.3}$$

The VAE is trained by maximizing the *expected* ELBO or, equivalently, minimizing the negative expected ELBO, approximated by Monte-Carlo sampling, and using the gradient descent algorithm with backpropagation [39]. One of the challenges in training the VAE is propagating the gradients with respect to parameters ϕ through the approximate posterior $q_\phi(z|x)$ in the reconstruction term, which cannot be computed directly:

$$\nabla_\phi \sum_i \mathbb{E}_{q_\phi(z|x^{(i)})}\; log\, p_\theta(x|z) = \sum_i \sum_j \nabla_\phi\, q_\phi(z^{(ij)}|x^{(i)})\, log\, p_\theta(x^{(i)}|z^{(ij)}) \tag{2.4}$$

To address this, the "reparameterization trick" is used, where the samples z from the posterior $q_\phi(z|x)$ are represented by a deterministic function of the encoder's output, which depends on ϕ. For example, when the encoder outputs parameters of the normal distribution $q_\phi(z|x) = N(\mu(x,\phi), \sigma^2(x,\phi)I)$, the latent variable z is reparameterized as $z = \mu(x,\phi) + \sigma^2(x,\phi) * \varepsilon = g_\phi(\varepsilon, x)$, where $\varepsilon \sim N(0,I) = p(\varepsilon)$, and $g_\phi(\varepsilon, x)$ is differentiable with respect to ϕ. Then, the gradients could propagate through $p(\varepsilon)$ since it does not depend on ϕ:

$$\begin{aligned}
\sum_i \sum_j \nabla_\phi\, q_\phi(z^{(ij)}|x^{(i)})\, log\, p_\theta(x^{(i)}|z^{(ij)}) &= \sum_i \sum_j \nabla_\phi\, p(\varepsilon^{(ij)})\, log\, p_\theta(x^{(i)}|g_\phi(\varepsilon^{(ij)}, x^{(i)})) \\
&= \sum_i \sum_j p(\varepsilon^{(ij)})\, \nabla_\phi\, log\, p_\theta(x^{(i)}|g_\phi(\varepsilon^{(ij)}, x^{(i)}))
\end{aligned} \tag{2.5}$$

2.2.2 Disentangled representation in latent spaces

A distinct attention in generative modeling-related research is given to learning disentangled latent representations of the data. Originally studied in the field of representation learning, a disentangled representation is defined as a representation of the data by a number of features corresponding to distinct observed sources of variation, such that a single feature is invariant to changes in the data related to other features [44]. As opposed to extracting features of the data that are relevant for a specific task, learning a disentangled representation is aimed at characterizing the data independently of the current task and is generally considered more expressive, reliable, interpretable, robust to variation peculiar to natural data, and useful for transfer learning [5] [44].

In the context of generative modeling, the idea of learning a disentangled representation is to derive a latent space, where different dimensions or different latent variables correspond to different factors of variation in the the data. Informally, in a disentangled latent space varying only one latent variable with the rest fixed should result in variation in a single concept in the generated images [10]. To learn a disentangled latent representation, a number of approaches have been proposed which could be divided at a high level into supervised and unsupervised. Unsupervised approaches for disentangled representation learning exploit assumptions about statistical independence of different latent variables, as well as assumptions about invariance of certain factors to specific changes in the input [44]. Whereas supervised approaches employ additional information characterizing observations, which could help the model separate different factors of variation.

Unsupervised disentanglement

In one of the first papers on unsupervised disentanglement in generative models the InfoGAN [10] model was proposed, where the mutual information between latent variables and observations is maximized in addition to a regular GAN objective. The idea of InfoGAN is that information about latent variables corresponding to meaningful factors of variation should improve the knowledge of which images should be generated, and thus a mutual information objective should stimulate the model to learn these factors in an unsupervised manner. Another prominent work [19] proposes a VAE-based unsupervised disentanglement approach, where the KL-divergence term from the regular VAE objective (2.3) is multiplied by a hyperparameter β, as shown in (2.6), which is aimed at constraining the capacity of the latent space, when $\beta > 1$, at a cost of some reduction in reconstruction quality. In turn, the increased penalty on the divergence of the approximate posterior $q_\phi(z|x)$ from the factorized prior $p(z)$ forces the model to split existing independent sources of variation in the data and assign them to separate dimensions of the latent space.

$$\mathbb{E}_{q_\phi(z|x)}\ log\, p_\theta(x|z)\ -\ \beta\, KL\left(q_\phi(z|x)\,||\,p(z)\right) \tag{2.6}$$

In another VAE-based approach [32] two models sharing the same idea were introduced, DIP-VAE-I and DIP-VAE-II, where instead of balancing the two terms of the VAE objective, as proposed in β-VAE, an additional regularization term for the "inferred prior", or combined posterior, $q_\theta(z) = \int q_\theta(z|x)\, p(x)\, dx$ is suggested, which forces $q_\theta(z)$ to not diverge from the assumed prior $p(z)$. The authors argue that penalizing the discrepancy between individual $q_\phi(z|x)$ and $p(z)$ does not guarantee disentanglement for the combined posterior $q(z)$.

Furthermore, a number of works suggest to improve the β-VAE approach. In the β-TCVAE model [9] the authors propose a decomposition of the KL divergence term that appears in the β-VAE training objective (2.6), splitting it into three terms. They argue that the ability of β-VAE to disentangle factors of variation is mainly explained by a single term in the decomposition, which is the penalty imposed on the total correlation between latent dimensions $\beta_{TC}\, KL(q(z)|\prod_i q(z_i))$, where $q(z)$ is the combined posterior. The authors provided an estimator of $q(z)$ and $q(z_i)$ and suggested that fine-tuning of the weight β_{TC} instead of β could yield better disentanglement for the same reconstruction quality. A similar approach to β-TCVAE is used in the FactorVAE [26] model, however, the authors introduce an additional discriminator network that approximates the total correlation term.

The JointVAE [13] model adapts β-VAE to also include discrete factors of variation and models the latent space as a joint distribution of continuous and discrete latent variables. Additionally, in [8] the objective of β-VAE is modified to introduce separate hyperparameters: C that reflects the "capacity" of the latent space the model ought to use, and γ being the penalty for diverging from that capacity (2.7). The authors propose to gradually increase C during training from zero, while keeping a high and constant γ, making the model prioritise the factors of variation to be learned on the basis of their contribution to the log-likelihood of the data. As a result, it helps achieve a refined balance between disentanglement and reconstruction quality.

$$\mathbb{E}_{q_\phi(z|x)}\ log\, p_\theta(x|z)\ -\ \gamma\left|KL\left(q_\phi(z|x)\,||\,p(z)\right) - C\right| \tag{2.7}$$

Several works addressed disentangling of image transformation in an unsupervised manner. For example, in the unsupervised version of Guided VAE [11] a subset of latent variables was directed to learn the parameters of the affine transformation of an image via an auxiliary decoder, while the content of the image was represented by the rest of the latent variables. A similar goal was pursued in Spatial VAE [6] with the difference that disentanglement of the latent variables corresponding to rotation and translation was integrated in the training process of a VAE.

Notably, it was demonstrated in [36] that learning of disentangled representations in a fully unsupervised fashion is impossible. Therefore, to derive meaningful and factorized latent space, supervision in some form or assumptions about the latent space are warranted.

Supervised disentanglement

Supervised approaches utilize labeling of the data, which can be either fully or partly available, constraints on the latent space, or prior knowledge about the sources of variation in the data to derive a disentangled representation. For example, the supervised version of Guided VAE [11] enforces distinct latent variables to be informative for predicting specific attributes, while simultaneously forcing the rest of the latent variables to be uninformative for the same attributes, via an auxiliary classification objective. In contrast, in [25] the authors propose a semi-supervised approach, where supervision is introduced implicitly, in the form of a collection of triplets that establish similarity between observations in terms of some observed factor, provided for a part of the observations in the data set.

In [31] the authors propose a VAE-based supervised disentanglement approach which forces specific units of the learned representation to encode the desired factors of variation in the images. However, instead of labels for the attributes of interest, as in Guided VAE [11], disentanglement is achieved by training a model on batches that include samples varying only in a single or in few selected factors at a time, while the inferred latent variables corresponding to not selected factors are averaged over a batch before being passed to the decoder. Hence, the approach exploits supervision in the form of grouping of images on the basis of which factors remain constant or vary in that group. A similar group-based supervision is used in ML-VAE [7], where the "content" of images, different between groups, and "style", different within a group, are separated. Whereas in [56] consecutive images in a video are paired, and the encoder is prompted to develop a compact representation of transition-related factors of variation, disentangling them from static features of images, which are encoded in separate dimensions.

Another perspective in disentanglement learning is presented in [49], where to derive a disentangled latent space the authors propose to combine probabilistic graphical models with deep generative models by defining the relationship between some latent variables and observations. Whereas the rest of variation in the data is captured by residual, using deep neural networks. For example, a latent variable corresponding to class labels can be directly included in the probabilistic model. Similarly, in [28] a latent variable for class label is explicitly included in the generative model to provide for semi-supervised classification.

DIVA [22] approach considers disentangled representation learning in the task of "domain generalization", where a representation is desired, which is invariant to domain-related variation.

The idea behind domain invariance is that the data of the same nature could originate from different sources, or domains, and thus can be susceptible to insignificant domain-related artefacts, such as background color. Therefore, in order utilize data from different and probably new domains in downstream tasks, such as regression or classification, the model should disentangle domain-related variation in the latent space. In their model, the authors propose to represent the latent space by three independent latent variables, all inferred with separate encoders with unshared parameters: z_y for class label, as in [49] [28], z_d for domain, and z_ε for residual variation in the data. Furthermore, disentanglement is stimulated by adding classifiers $q_{\omega_y}(y|z_y)$ and $q_{\omega_d}(d|z_d)$ into the VAE training objective that are aimed to predict class and domain labels based on respective latent variables.

$$KL\left(q_{\phi_y}(z_y|x)\,||\,p(z_y|y)\right) \tag{2.8}$$

Notably, in contrast to a regular VAE, in DIVA [22] *conditional* priors $p(z_d|d)$ and $p(z_y|y)$ are used for the class and domain latent variables, while at the same time using a conventional unconditional prior $p(z_\varepsilon)$ for residual variation. Such formulation allows for conditional generation of images for a given class and domain and leads to a less restricted grouping of points corresponding to observations with different labels in the latent space, since the KL-term (2.8) does not force conditional priors to be concentrated around a single origin. Furthermore, the idea of conditional prior was also used in the case with a continuous attribute in [59], where a VAE-based model for regression was developed. In this model, a prior distribution conditioned on the target feature $p(z|c)$ helps to shape and order the latent distribution according to its values.

To conclude, supervised approaches allow to derive meaningful and factorized latent representations when there is a notion of which factors of variation need to be separated in the latent space, and when the labels or values of these factors, or another supervisory information, are fully or in part available. Furthermore, by introducing a conditional prior distribution which is conditioned on class labels or attribute values, a latent space or subspace with useful structure can be learned.

Disentanglement metrics

Multiple metrics were proposed to measure the level of disentanglement in the latent space reached by different models, some of which are listed in [36]. In β-VAE [19] a classifier-based metric was proposed, where a classifier is trained to predict which factor was fixed in pairs of images based on the observed difference between latent vectors corresponding to these images, and accuracy serves as a measure of disentanglement. The metric proposed in FactorVAE [26] is an improvement of the above mentioned metric. A a majority-vote classifier is trained to predict which factor was fixed based on indices of the lowest-variance latent dimension, collected for different batches.

In the work on β-TCVAE [9] the metric Mutual Information Gap (MIG) was considered, where, firstly, mutual information as a measure of informativeness of a latent dimension for a given factor is calculated for each latent dimension, and, secondly, the difference between two most informative latent dimensions is calculated and is considered be the measure of disentanglement. A low MIG implies that there are at least two latent dimensions, similar in terms of informativeness for a given factor, and thus a representation is poorly disentangled. The SAP score metric [32] takes the most informative latent dimensions per factor, takes the difference between the prediction scores of these dimensions, and finally the average difference is calculated. Other metrics, including DCI [14] and Modularity [45], have been discussed in the literature as well.

However, the level of disentanglement in different models is frequently assessed using qualitative judgement. One of the approaches is latent space traversals [26], when a single latent varible or a latent dimension is manipulated while keeping the rest fixed. Resulting generated images demonstrate whether a single or multiple factors of variation are changing, which gives an impression of whether a certain latent representation is disentangled.

2.3 Related work on simulations in biomaterials research

The idea of a simulation model of biomaterials research experiments is considered in [53], where the authors discuss the concept of a 'hypothesis-driven' mathematical simulation model, which is built upon prior knowledge about the underlying biological mechanisms that govern cell behavior and does not use data. As opposed to the mathematical modeling perspective, this work considers the concept of a purely data-driven simulation model, based on the generative modeling approach, hence no information about biological mechanisms is embedded in the model. A literature review in search for similar applications of generative models, or data-driven simulation approaches in biomaterials research, however, yielded little results.

At the same time, generative models are frequently used in related biomedical applications, particularly in applications involving screening data and cell images. For instance, in order to perform dimensionality reduction [58], data augmentation [2], to learn useful representation of the cell image data [16] [48] [24] [57]. Furthermore, generative models, mostly GANs, are used in the task of cell image synthesis [40] [30]. However, the task to synthesize cell images, or other measures of a cell response, based on the input conditions of an experiment is poorly covered in the literature as of today.

Chapter 3

Method

In this chapter, a novel generative model is described that is aimed to serve as 1. a simulation model of the cell-surface topography experiment, and 2. as a tool for *in silico* topography design. Notably, it the present work it is assumed that 1. the space X is the space of individual cell images, and 2. P is a space of images of topographies, which can be either an image of a topographical feature (Figure 2.1), or an image of a full topography (Figure 2.2), i.e. a grid of topographical features. A detailed description of the data used in experiments is provided in Chapter 4.

3.1 Motivation

A major challenge in development of a model that is able to mimic the mappings $P \to X$ and $X \to P$ between two image spaces in a probabilistic way is the dimensionality and sparsity of the data. For instance, considering the space X, a $128 * 128$ pixels cell image with a single color channel is a point in the space $[0, 255]^{16384}$. Directly modeling the probability distribution $p(x)$ is unlikely to yield satisfactory results, since, firstly, a 16384-dimensional probability distribution is hard to represent and learn, and, secondly, the probability mass of the subspace that corresponds to images of cells in the space of all possible $128 * 128$ images is negligible. Furthermore, the variation in the observed cell images is likely to be explained by a small number of factors: cell elongation, area, orientation, etc., which makes modeling the probability distribution over pixel values impractical. The task to model the probability distribution of two high-dimensional datasets and the relationship between them is even harder.

To address this challenge, generative models (2.2) employ various techniques to represent probability distributions in high-dimensional spaces and, particularly, latent variable models. This idea implies that both spaces X and P can be represented in lower-dimensional latent spaces Z_X and L_P, respectively. Accordingly, to sample new images of topographies or cells, the decoders $d_X : Z_X \to X$ and $d_P : L_P \to P$ are trained. By extension, the key idea exploited in the present work is to attempt to model the twofold relationship $P \leftrightarrow X$ between topography and cell images in the latent spaces, instead of the original spaces, as illustrated in Figure 3.1. The remaining part of this section is organized as follows: firstly, an overview of the proposed model is provided; secondly, a formal description of the proposed model and the training procedure details are provided; finally, the application scenarios are discussed.

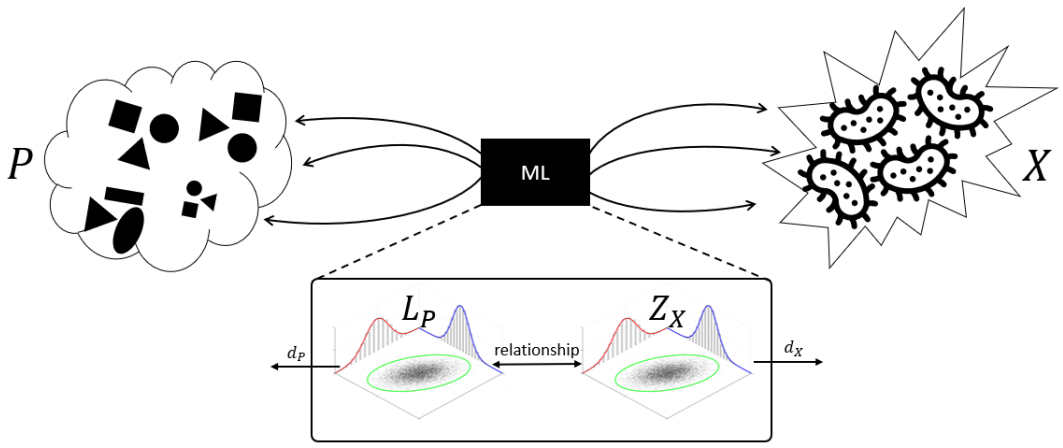

Figure 3.1: Modeling the relationship $P \leftrightarrow X$ in latent spaces.

3.2 Model concept

An assumption made in the proposed model is that topographies could affect cell response, expressed as images, only through some visually discernible, measurable, and independent cell properties, which are of interest for a particular task, e.g. nucleus FormFactor, Perimeter, Solidity [21], and for which all values are available per each cell image. Accordingly, a cell image x is assumed to originate from a combination of these cell properties f_i and residual variation ε, as shown in Figure 3.2. It is also assumed that the noise component ε is not influenced by topographies and is irrelevant for the analysis of the cell-surface topography relationship.

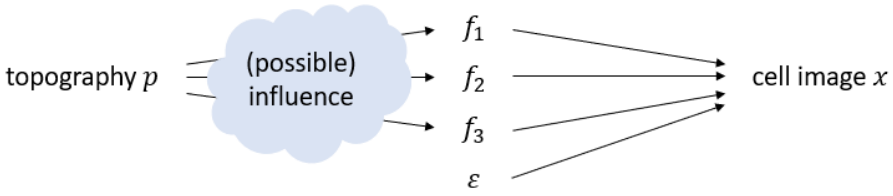

Figure 3.2: Assume p influences x through features f.

Given the assumptions mentioned above, the first component of the proposed approach is a disentangled latent space of the cell image dataset X. It is proposed to learn a fully factorized latent space $Z_X = Z_{f_1} \times Z_{f_2} \times \ldots \times Z_{f_n} \times Z_\varepsilon$, where n is the number of cell features, such that each latent subspace Z_{f_i} encodes the variation in cell images explained only by the feature f_i. While the subspace Z_ε of the latent space Z_X captures the residual variation in cell images that is not explained by the features f_i. An illustration of a factorized latent space with three cell features is provided in Figure 3.3. To generate new cell images a decoder d_X is trained that is able to map a combination of points $z_{f_1} \sim Z_{f_1}, \ldots, z_{f_n} \sim Z_{f_n}, z_\varepsilon \sim Z_\varepsilon$ to a distribution $p(x|z) = p(x|z_{f_1}, \ldots, z_{f_2}, z_\varepsilon)$.

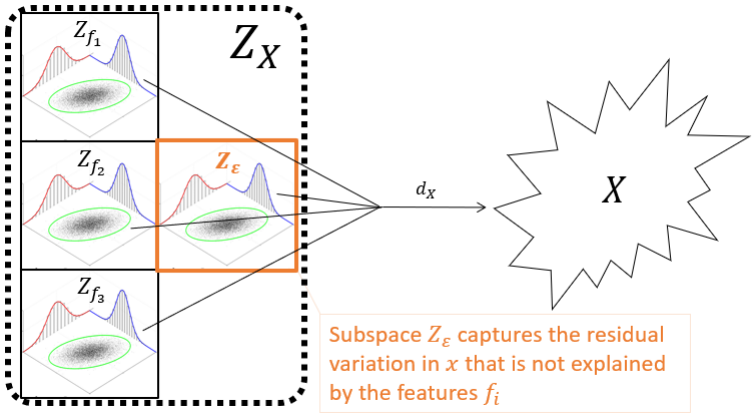

Figure 3.3: Learn a factorized latent space Z_X.

The second core component of the model is the latent space L_P for the dataset P of topographies. The key idea of the present work is that instead of learning a fully independent latent space of topographies and subsequently modeling the relationship between L_P and Z_X, it is proposed to use a shared latent space to represent the cell-surface topography relationship. The idea is to reuse the latent subspaces $Z_{f_1}, \ldots, Z_{f_n}$, jointly referred to as Z_f, as part of the latent space L_P. The remaining part of the latent space L_P is represented by an additional subspace L_ε. Notably, the latent subspace Z_{f_i}, being considered as part of L_P, changes its interpretation and now represents the *influence* of topographies on the cell feature f_i. In turn, the latent subspace L_ε is

introduced to capture the variation in topographies that is not related to their influence on cells. An illustration of the latent space L_P with a shared subspace Z_f is provided below in Figure 3.4.

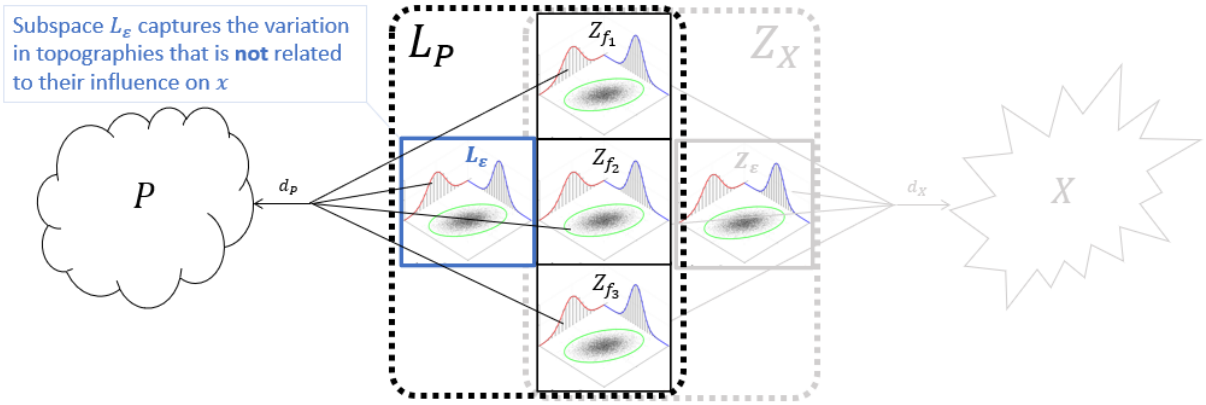

Figure 3.4: Reuse the latent subspaces Z_{f_i} as part of L_P.

Such a formulation implies that the latent space $L_P = Z_{f_1} \times \ldots \times Z_{f_n} \times L_\varepsilon$ is also factorized, which suggests two strong assumptions. Firstly, each factor of variation in topographies that to some extent impacts one of the cell features should be encoded in one of the subspaces Z_{f_i}, while not being captured in the residual subspace L_ε. Secondly, a single factor of variation in topographies can influence only a single cell feature f_i by design and should be encoded in the corresponding subspace Z_{f_i} in that case. Arguably, while not being generally justified, these assumptions do not constrain the ability of such a model to be used in the desired applications, as it is shown in Chapter 5.

3.3 Model description

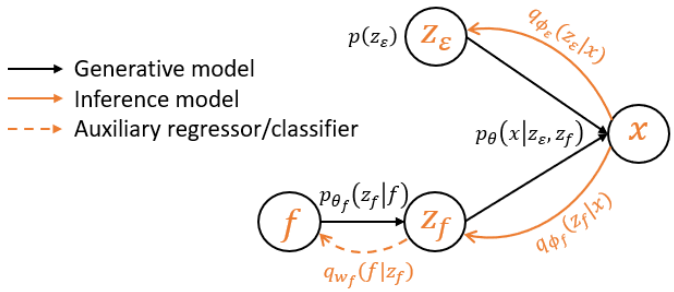

Figure 3.5: Graphical model for the cell image dataset (cell model).

The first component of the full simulation model is a generative model for the cell image dataset X, referred to as the *cell model*. The proposed generative model, as illustrated in Figure 3.5, is based on the VAE [29] (2.2.1) and is inspired by the DIVA [22] approach. Since it is assumed that the latent space is factorized, it is represented by independent latent variables $z_{f_1}, \ldots, z_{f_n}, z_\varepsilon$ with separate encoders $q_{\phi_{f_1}}(z_{f_1}|x), \ldots, q_{\phi_{f_n}}(z_{f_n}|x)$, $q_{\phi_\varepsilon}(z_\varepsilon|x)$ (3.1). To simplify notation, f represents either a single or multiple cell features $f_1, \ldots, f_n$ chosen to characterize a cell image x, and, accordingly, z_f corresponds to either a single or multiple latent variables. The respective parameters ϕ_f, ϕ_ε of the encoders are not shared, as suggested by [22]. The latent variable z_f has a conditional prior distribution $p_{\theta_f}(z_f|f)$ with parameters θ_f, while z_ε has an independent prior

distribution $p(z_\varepsilon)$ (3.2). A single decoder $p_\theta(x|z_\varepsilon, z_f)$ with parameters θ is used to generate cell images. The joint distribution of all variables is provided in (3.3).

$$q_{\phi_f}(z_f|x) = N(\mu_{\phi_f}(x), \sigma^2_{\phi_f}(x)I) \qquad q_{\phi_\varepsilon}(z_\varepsilon|x) = N(\mu_{\phi_\varepsilon}(x), \sigma^2_{\phi_\varepsilon}(x)I) \tag{3.1}$$

$$p_{\theta_f}(z_f|f) = N(\mu_{\theta_f}(f), \sigma^2_{\theta_f}(f)I) \qquad p(z_\varepsilon) = N(0, I) \tag{3.2}$$

$$p(x, z_\varepsilon, z_f, f) = p_\theta(x|z_\varepsilon, z_f)p(z_\varepsilon)p_{\theta_f}(z_f|f)p(f) \tag{3.3}$$

To support disentanglement of the latent variables z_f, z_ε, the supervised disentanglement DIVA approach proposed in [22] is used. Auxiliary regressors or classifiers $q_{\omega_f}(f|z_f)$, depending on whether a particular cell feature f is continuous or categorical, are introduced that aim to predict the values or labels of the corresponding cell features f based on latent representations z_f of cell images x. The KL-terms from the VAE training objective for both posterior distributions are multiplied by hyperparameters β_f, β_ε, as proposed in [19], which would allow to control the capacity of the latent subspaces [8]. The training objective of the described model according to DIVA approach [22] would be as provided in (3.4). It is defined for a pair of a cell image x and a vector f of cell features and needs to be maximized. The last term in $F_1(x, f)$ corresponds to the performance of an auxiliary regressor or classifier $q_{\omega_f}(f|z_f)$, whose importance is regulated by the hyperparameter α_f.

$$\begin{aligned} F_1(x, f) = \mathbb{E}_{q_{\phi_\varepsilon}(z_\varepsilon|x)q_{\phi_f}(z_f|x)}\, log\, p_\theta(x|z_\varepsilon, z_f) \; &- \; \beta_f\, KL\left(q_{\phi_f}(z_f|x)\,||\,p_{\theta_f}(z_f|f)\right) \\ &- \; \beta_\varepsilon\, KL\left(q_{\phi_\varepsilon}(z_\varepsilon|x)\,||\,p(z_\varepsilon)\right) \\ &+ \; \alpha_f\, \mathbb{E}_{q_{\phi_f}(z_f|x)}\, log\, q_{\omega_f}(f|z_f) \end{aligned} \tag{3.4}$$

However, it is further proposed to introduce an auxiliary parameterized normal prior $p_{\theta_{pr}}(z_f)$ (3.6), referred to as the *full prior*, for conditional prior distributions $p_{\theta_f}(z_f|f)$. It is implemented by augmenting the original training objective of the cell model (3.4) with another KL-term, as shown in the final cell-model objective (3.7). The full prior has a zero mean, while the parameters representing the variance by dimensions are made trainable with the condition that their sum equals one. The idea behind the full prior is that 1. it imposes a normal distribution on the marginal distribution $p_{\theta_f}(z_f)$ (3.5), the shape of which is otherwise uncontrolled; 2. it approximates the marginal distribution by learning the variance parameters instead of calculating it directly (3.5), and hence the full prior with learned variance parameters $p_{\theta^*_f}(z_f)$ can be used as a substitute for the marginal distribution. Furthermore, the trainable relative variance of the full prior allows for an elongated shape of the marginal distribution $p_{\theta_f}(z_f)$, i.e. it allows the marginal distribution to be stretched along one axis more, than along another. An elongated shape is assumed to be beneficial, since the means of the posterior distributions $q_{\phi_f}(z_f|x)$ are encouraged to be arranged in the z_f space according to values of f by the conditional prior in the first KL-term (3.4).

$$p_{\theta_f}(z_f) = \int p_{\theta_f}(z_f|f)p(f)df \tag{3.5}$$

$$p_{\theta_{pr}}(z_f) = N(0, \sigma^2_{\theta_{pr}}I), \quad \text{such that} \sum_i (\sigma_{\theta_{pr}})^2_i = 1 \tag{3.6}$$

$$F_1^{pr}(x, f) = F_1(x, f) - \; \beta_{pr}\, KL\left(p_{\theta_f}(z_f|f)\,||\,p_{\theta_{pr}}(z_f)\right) \tag{3.7}$$

The second component of the proposed method is a generative model for topographies, referred to as the *topography model*, that aims both to learn the distribution of topography images in the latent space and to unravel the relationship between the datasets. As mentioned above, it is proposed that the latent spaces of topographies and images have a shared latent subspace Z_f.

This subspace is represented by the latent variable z_f and is intended to encode the influence of topographies on the feature f. Whereas an additional independent latent variable l_ε captures the variation in topographies that is uninformative for cell response prediction. The topography model is illustrated in Figure 3.6, and the combined model is illustrated in Figure 3.7.

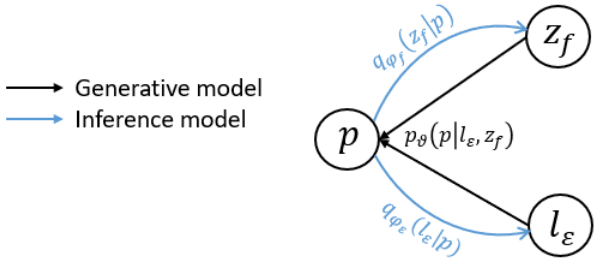

Figure 3.6: Graphical model for the topography image dataset (topography model).

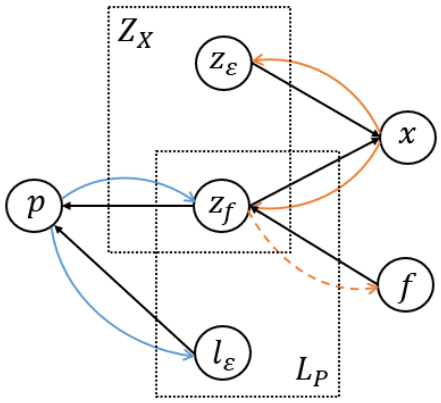

Figure 3.7: Combined model with a shared latent variable z_f.

Similarly to the cell model, the latent variables z_f, l_ε in the topography model have distinct encoders $q_{\varphi_f}(z_f|p)$, $q_{\varphi_\varepsilon}(l_\varepsilon|p)$ (3.8) with unshared parameters, and a single decoder $p_\vartheta(p|l_\varepsilon, z_f)$ maps samples from the factorized latent space to the original space of topographies. The latent variable l_ε has an independent standard normal prior $p(l_\varepsilon) = N(0, I)$, identical to $p(z_\varepsilon)$ (3.2), while for the latent variable z_f the full prior distribution from the cell model $p_{\theta^*_{pr}}(z_f)$ is taken as a prior distribution (3.9). The variance parameters of the full prior are learned during training of the cell model (3.7) and are fixed in the topography model, which is denoted by $*$ sign. The joint distribution of all topography-related variables is given by (3.10). Accordingly, a DIVA approach-based training objective for the topography model, taken separately from the combined model, would be as formulated in (3.11), where the hyperparameters β_{pf}, β_l control the capacity of the latent subspaces z_f, l_ε respectively. The difference from the cell-model objective is that the latent variable z_f is not supervised by an auxiliary regression or classification objective, and that an independent (unconditional) prior is used for z_f.

$$q_{\varphi_f}(z_f|p) = N(\mu_{\varphi_f}(p), \sigma^2_{\varphi_f}(p)I) \qquad q_{\varphi_\varepsilon}(l_\varepsilon|p) = N(\mu_{\varphi_\varepsilon}(p), \sigma^2_{\varphi_\varepsilon}(p)I) \tag{3.8}$$

$$p_{topmodel}(z_f) \equiv p_{\theta^*_{pr}}(z_f) \approx p_{\theta_f}(z_f) \tag{3.9}$$

$$p(p, l_\varepsilon, z_f) = p_\vartheta(p|l_\varepsilon, z_f)p(l_\varepsilon)p_{\theta^*_{pr}}(z_f) \tag{3.10}$$

$$\begin{aligned} F_2(p) = \mathbb{E}_{q_{\varphi_l}(l_\varepsilon|p)q_{\varphi_f}(z_f|p)}\ log\, p_\vartheta(p|l_\varepsilon, z_f) &- \beta_{pf}\, KL\left(q_{\varphi_f}(z_f|p)\,||\,p_{\theta^*_{pr}}(z_f)\right) \\ &- \beta_l\, KL\left(q_{\varphi_l}(l_\varepsilon|p)\,||\,p(l_\varepsilon)\right) \end{aligned} \tag{3.11}$$

However, it should be noted that jointly or consecutively maximizing the cell-model objective $F_1^{pr}(x, f)$ (3.7) and the (draft) topography-model objective $F_2(p)$ (3.11) is not sufficient to learn the cell-surface topography relationship. In fact, training both objectives would result in two independent models with unrelated latent variables denoted as z_f, since there is no component that binds topographies and cells from the training pairs (p^i, x^i) in the combined architecture. Therefore, the question remains, how to enforce z_f to represent the *same* latent variable both for cells and topographies, as shown in Figure 3.8.

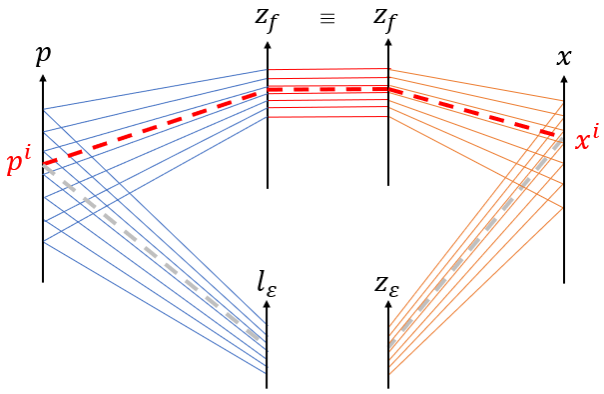

Figure 3.8: The task of enforcing z_f to be the same latent variable for x and p.

To address the outlined issue, a three-step training procedure for the combined model and two auxiliary objective terms in the topography-model are proposed. In the **first step**, it is proposed that the cell model is trained independently of the topography model, since the latent variables z_f and z_ε in the cell model are intended to solely capture the variation in cell images, and the model aims to learn a disentangled latent representation of the cell image dataset. The cell model model is trained by maximizing the formulated above objective $F_1^{pr}(x, f)$ (3.7), or equivalently by minimizing the negative $F_1^{pr}(x, f)$, with respect to all parameters involved. Later the weights of the encoders, decoder, conditional prior, regressor, and of the full prior (3.12) are fixed, which is denoted by $*$, and are not influenced by the subsequent training procedure.

$$q_{\phi_f^*}(z_f|x) \quad q_{\phi_\varepsilon^*}(z_\varepsilon|x) \quad p_{\theta^*}(x|z_\varepsilon, z_f) \quad p_{\theta_f^*}(z_f|f) \quad q_{\omega_f^*}(f|z_f) \quad p_{\theta_{pr}^*}(z_f) \tag{3.12}$$

In the **second step** of the proposed training procedure, only the encoders of the topography model are trained that relate to the z_f latent variables, i.e. $q_{\varphi_f}(z_f|p)$. The goal of these encoders is to capture the influence of topographies on respective cell features through z_f, and the encoders are subsequently used in the task of topography-conditioned cell image generation, i.e. to simulate the experiment. The pursued behavior of the model is as follows: given a topography p, the encoder produces a posterior distribution $q_{\varphi_f}(z_f|p)$; then, samples $z_f \sim q_{\varphi_f}(z_f|p)$ from the posterior, coupled with arbitrary samples $z_\varepsilon \sim p(z_\varepsilon)$ from the prior in the residual subspace, are passed to the cell-model decoder $p_{\theta^*}(x|z_\varepsilon, z_f)$; and the decoder produces cell images with the topography-induced values of the cell feature f.

To train the encoders $q_{\varphi_f}(z_f|p)$, an auxiliary likelihood-based term 2Ax (3.13) is introduced in the training objective; it is discussed in more detail in a later section. The idea behind this term is that cell images $\hat{x}^i \sim p_{\theta^*}(x|z_\varepsilon, z_f)$ generated based on a topography p^i as described above should be similar to the original cell image x^i from a given training pair (p^i, x^i). Notably, the proposed term (3.13) suggests that $z_\varepsilon \sim q_{\phi_\varepsilon^*}(z_\varepsilon|x)$ is sampled from the cell-model posterior during training, as opposed to the actual experiment simulation scenario when z_ε is sampled from the prior. It is hypothesized that sampling z_ε from the posterior should benefit the training process, since it would increase the likelihood of generated cell images. The auxiliary term 2Ax is maximized in combination with the negative KL regularization term $KL\left(q_{\varphi_f}(z_f|p)\,||\,p_{\theta_{pr}^*}(z_f)\right)$ from the draft topography-model objective (3.11). In this way, the posterior is forced to match the full prior by default, allowing for all values of f in generated cell images during simulation. However, the posterior would diverge from the full prior to cover only certain regions of the z_f space, corresponding to the induced cell feature values f, if the influence of topographies on this feature is detected. Hence, the resulting training objective of the second step is formulated as (3.14).

$$\max_{\varphi_f} \quad \mathbb{E}_{q_{\phi_\varepsilon^*}(z_\varepsilon|x)q_{\varphi_f}(z_f|p)}\ log\, p_{\theta^*}(x|z_\varepsilon, z_f) \tag{3.13}$$

$$F_{2Ax}(x,p) = \mathbb{E}_{q_{\phi^*_\varepsilon}(z_\varepsilon|x)q_{\varphi_f}(z_f|p)}\ log\, p_{\theta^*}(x|z_\varepsilon, z_f)\ -\ \beta_{pf}\, KL\left(q_{\varphi_f}(z_f|p)\,||\,p_{\theta^*_{pr}}(z_f)\right) \tag{3.14}$$

In the **third step** of the training procedure, the remaining components of the topography model are trained: the encoder $q_{\varphi_l}(l_\varepsilon|p)$ and the decoder $p_\vartheta(p|l_\varepsilon, z_f)$, while the weights of the previously mentioned encoder $q_{\varphi^*_f}(z_f|p)$ are fixed. In this step, the topography model is trained both for the task of topography reconstruction, to serve as a VAE for topographies, and for the task of cell-conditioned topography design, which is the second main application of the combined model. In the latter application, the pursued behavior of the model is inverse to experiment simulation: given a cell image x, the cell-model encoder produces a posterior distribution $q_{\phi^*_f}(z_f|x)$, samples from which are passed to the topography-model decoder $p_\vartheta(p|l_\varepsilon, z_f)$ along with arbitrary $l_\varepsilon \sim p(l_\varepsilon)$; and the decoder produces topography images that could result in the given cell image.

The topography reconstruction objective is secured by the likelihood term in the initial objective (3.11). Whereas to achieve the desired behavior with respect to cell-conditioned topography generation, another likelihood-based auxiliary objective is introduced, 2B (3.15). Similarly to the previous step, it is based on the idea that topographies generated based on a given cell image should be similar to those from the training pairs. However, in contrast to the auxiliary term 2Ax, the residual latent variable of the topography model $l_\varepsilon \sim p(l_\varepsilon)$ is sampled from the prior distribution during training; the motivation for that is discussed in a later section. Finally, the KL term pertaining to the residual space l_ε from (3.11) is added in this step. The resulting training objective is formulated in (3.16), where η regulates the importance of the auxiliary term.

$$\max_{\vartheta} \quad \mathbb{E}_{p(l_\varepsilon)q_{\phi^*_f}(z_f|x)}\ log\, p_\vartheta(p|l_\varepsilon, z_f) \tag{3.15}$$

$$\begin{aligned} F_{orig,2B}(x,p) = \mathbb{E}_{q_{\varphi_l}(l_\varepsilon|p)q_{\varphi^*_f}(z_f|p)}\ log\, p_\vartheta(p|l_\varepsilon, z_f)\ +\ \eta\ \mathbb{E}_{p(l_\varepsilon)q_{\phi^*_f}(z_f|x)}\ log\, p_\vartheta(p|l_\varepsilon, z_f) \\ -\ \beta_l\, KL\left(q_{\varphi_l}(l_\varepsilon|p)\,||\,p(l_\varepsilon)\right) \end{aligned} \tag{3.16}$$

3.3.1 Training procedure summary

The training procedure of the full model is formulated as follows:

1. All the components of the cell model are trained in the first step. It is done by maximizing the objective $F_1^{pr}(x,f)$ (3.17) with respect to all involved parameters: $\phi_\varepsilon, \phi_f, \theta, \theta_f, \theta_{pr}$. The hyperparameters include $\beta_\varepsilon, \beta_f, \beta_{pr}, \alpha_f$. Subsequently, the weights of all the components of the cell model are fixed (3.18), which is denoted by $*$ sign.

$$\begin{aligned} F_1^{pr}(x,f) = \mathbb{E}_{q_{\phi_\varepsilon}(z_\varepsilon|x)q_{\phi_f}(z_f|x)}\ log\, p_\theta(x|z_\varepsilon, z_f)\ -\ \beta_\varepsilon\, KL\left(q_{\phi_\varepsilon}(z_\varepsilon|x)\,||\,p(z_\varepsilon)\right) \\ -\ \beta_f\, KL\left(q_{\phi_f}(z_f|x)\,||\,p_{\theta_f}(z_f|f)\right) \\ -\ \beta_{pr}\, KL\left(p_{\theta_f}(z_f|f)\,||\,p_{\theta_{pr}}(z_f)\right) \\ +\ \alpha_f\ \mathbb{E}_{q_{\phi_f}(z_f|x)}\ log\, q_{\omega_f}(f|z_f) \end{aligned} \tag{3.17}$$

$$q_{\phi^*_f}(z_f|x) \quad q_{\phi^*_\varepsilon}(z_\varepsilon|x) \quad p_{\theta^*}(x|z_\varepsilon, z_f) \quad p_{\theta^*_f}(z_f|f) \quad q_{\omega^*_f}(f|z_f) \quad p_{\theta^*_{pr}}(z_f) \tag{3.18}$$

2. In the second step, the topography-model encoder $q_{\varphi_f}(z_f|p)$ corresponding to the shared latent variable z_f is trained by maximizing the objective $F_{2Ax}(x,p)$ (3.19) with respect to parameters φ_f. The only hyperparameter in this step is β_{pf}. Subsequently, the weights of the encoder are fixed: $q_{\varphi^*_f}(z_f|p)$.

$$F_{2Ax}(x,p) = \mathbb{E}_{q_{\phi^*_\varepsilon}(z_\varepsilon|x)q_{\varphi_f}(z_f|p)}\ log\, p_{\theta^*}(x|z_\varepsilon, z_f)\ -\ \beta_{pf}\, KL\left(q_{\varphi_f}(z_f|p)\,||\,p_{\theta^*_{pr}}(z_f)\right) \tag{3.19}$$

3. Finally, the remaining components of the topography model $q_{\varphi_l}(l_\varepsilon|p)$, $p_\vartheta(p|l_\varepsilon, z_f)$ are trained by maximizing the objective $F_{orig,2B}(x,p)$ (3.20) with respect to parameters φ_l, ϑ. The hyperparameters include η, β_l.

$$F_{orig,2B}(x,p) = \mathbb{E}_{q_{\varphi_l}(l_\varepsilon|p)q_{\varphi_f^*}(z_f|p)}\ log\, p_\vartheta(p|l_\varepsilon, z_f) + \eta\, \mathbb{E}_{p(l_\varepsilon)q_{\phi_f^*}(z_f|x)}\ log\, p_\vartheta(p|l_\varepsilon, z_f) - \beta_l\, KL\left(q_{\varphi_l}(l_\varepsilon|p)\,||\,p(l_\varepsilon)\right) \tag{3.20}$$

3.4 Reasoning behind the training objectives

Choosing a prior distribution for z_f in the topography model

A conditional prior $p_{\theta_f}(z_f|f)$ in the cell model was used to help the model spread the means of the posterior distributions in the latent space according to the values of the cell feature f, such that distant regions of the latent space would correspond to high and low values of f respectively. If, alternatively, a standard normal prior $p(z_f) = N(0, I)$ was used, all z_f points would be forced to concentrate around a single origin (zero), which is unnecessary. However, using a conditional prior in the topography model might not be as reasonable. The values of features f are defined for cells, but not for topographies, and a single topography may correspond to cell images with drastically different values of a given cell feature f in case there is no influence of topographies on this feature. Consequently, minimization of a KL-divergence with a conditional prior (3.21) would encourage the posterior $q_{\varphi_f}(z_f|p)$ to cover the whole latent space, which could include regions with no support, if the shape of the marginal distribution $p_{\theta_f}(z_f)$ is not controlled, as in the initial cell-model objective (3.4), i.e. before the introduction of the full prior.

$$KL\left(q_{\varphi_f}(z_f|p)\,||\,p_{\theta_f}(z_f|f)\right) \tag{3.21}$$

To illustrate the problem, consider the training data $\{p^i, x^i\}_{i=1}^n$ and suppose that some factor of variation in topographies influences the cell feature f. Notably, a single topography p^i may correspond to a number of cells x_j^i with the cell feature values f_j^i. Suppose that the topography p^i from the training data is paired with a range of cells having high values f_j^i. Then minimizing the KL-divergence (3.21) will encourage the posterior $q_{\varphi_f}(z_f|p^i)$ to cover all of the individual conditional priors $p_{\theta_f}(z_f|f_j^i)$ lying in the part of the latent space that corresponds to high values of f. An illustration of this situation is provided in Figure 3.9. However, in the opposite case, when there is no influence of topographies on the cell feature f, a full range of f values could be observed for a single topography p, for instance, both high and low values of f. Then the encoder would be prompted to produce a posterior distribution that covers distant regions of the marginal distribution $p_{\theta_f}(z_f)$, in which different conditional priors lie. As a result, since no constraints are imposed on the marginal distribution $p_{\theta_f}(z_f)$ in the current example, the posterior could overlap with no-support regions of the latent space, as shown in Figure 3.10.

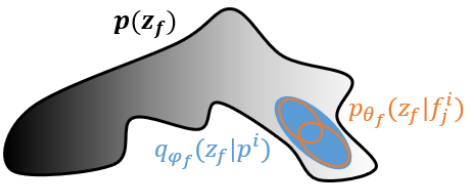

Figure 3.9: Minimizing (3.21) when p influences f.

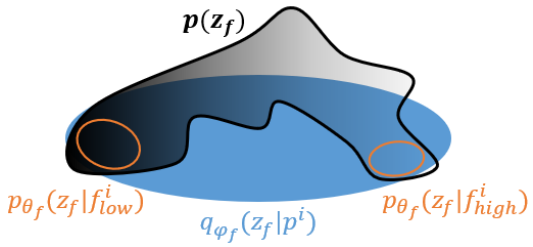

Figure 3.10: Minimizing (3.21) when p does not influence f.

Hence, using a conditional prior $p_{\theta_f}(z_f|f)$ in the topography model is undesirable *in case* the marginal distribution of z_f is unknown and uncontrolled. In other words, either the marginal distribution should be shaped to have a 'convex' support to allow for the use of a conditional prior in the topography model, or an alternative prior should be used, for example, the marginal distribution $p_{\theta_f}(z_f)$ itself. However, to use the marginal distribution $p_{\theta_f}(z_f)$ itself as the prior in the topography model, it needs to be at least calculated, as shown in (3.5), which involves integrating over f values. To circumvent this, it was proposed to use the full prior distribution $p_{\theta_{pr}}(z_f)$ (3.6) in the cell-model training objective (3.7): as discussed above, the full prior $p_{\theta_{pr}}(z_f)$ with learned variance parameters approximates the marginal distribution $p_{\theta_f}(z_f)$ without calculating it directly and thus can be used as a prior distribution of z_f in the topography model. Furthermore, after imposing the full prior in the cell model, the marginal distribution $p_{\theta_f}(z_f)$ is expected to have a 'convex' support of a normal distribution, which would eliminate the problem of a conditional prior in the topography model outlined above. Now a posterior $q_{\varphi_f}(z_f|p)$ covering any combination of conditional priors $p_{\theta_f}(z_f|f)$ as a result of minimizing (3.21) is expected to stay inside the marginal distribution. Hence, once the full prior is introduced in the cell model, a conditional prior $p_{\theta_f}(z_f|f)$ can also be used in the topography model. In this paper the decision is to use the full prior $p_{\theta_{pr}}(z_f)$ as the prior distribution for z_f in the topography model, while the term (3.21) is considered later from a different angle, as a distance-based auxiliary objective.

Enforcing a shared latent space

As mentioned above, it is important to ensure that z_f represents the same latent variable in the cell model and in the topography model, which can only be achieved with an auxiliary objective or objectives that use the training pairs (p^i, x^i). Arguably, an auxiliary objective needs to be determined based on the desired behavior of the model. Considering the first goal to encode the influence of topographies on cells, the desired behavior is as follows: if topographies influence the cell feature f, the topography encoder $q_{\varphi_f}(z_f|p)$ should map a topography p^i inducing high values of the cell feature f to a posterior distribution covering the region of the latent space z_f that corresponds to high values of f, and vice versa, as shown in Figure 3.11. In other words, the posterior distribution $q_{\varphi_f}(z_f|p^i)$ should overlap with the posterior distribution $q_{\phi_f^*}(z_f|x^i)$ and with a conditional prior $p_{\theta_f^*}(z_f|f^i)$, where (p^i, x^i) is a training pair, and f^i is the cell feature value for x^i. As a result, by sampling $z_f \sim q_{\varphi_f}(z_f|p^i)$ from the topography-model posterior and $z_\varepsilon \sim p(z_\varepsilon)$ from the cell-model prior, the cell-model decoder $p_{\theta^*}(x|z_\varepsilon, z_f)$ would be able to generate cell images with high values of the cell feature f.

In the opposite case, when topographies do not influence the cell feature f, the model should reflect the fact that *all* feature values are possible for a given topography. In terms of the latent representation, the posterior distribution $q_{\varphi_f}(z_f|p^i)$ for a given topography p^i should match the marginal distribution of z_f, which is approximated by $p_{\theta_{pr}^*}(z_f)$ (3.9). Accordingly, by sampling $z_f \sim q_{\varphi_f}(z_f|p^i) \approx p_{\theta_{pr}^*}(z_f)$ and $z_\varepsilon \sim p(z_\varepsilon)$, the cell-model decoder would be able to generate cell images with any value of f. An illustration for this case is provided in Figure 3.12.

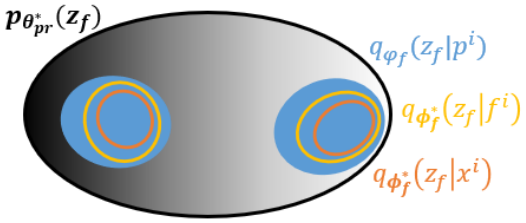

Figure 3.11: Desired behavior: p influences f.

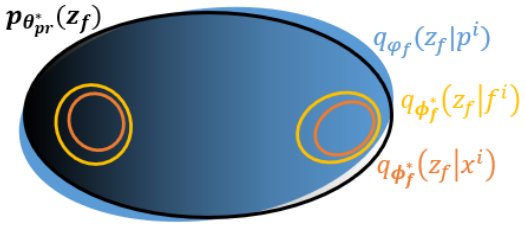

Figure 3.12: Desired behavior: p does not influence f.

The second goal is to capture the irrelevant for the cell response factors of variation by the latent variable l_ε. The problem arises from the fact that both encoders $q_{\varphi_\varepsilon}(l_\varepsilon|p)$ and $q_{\varphi_f}(z_f|p)$ are not supervised in capturing the "informative" or the "residual" factors of variation in the initial objective (3.11). Furthermore, the decoder $p_\vartheta(p|l_\varepsilon, z_f)$ has no mechanism to discern between the two latent variables z_f, l_ε. A disentangled latent representation of topographies is important, however, since the task of reconstructing a topography image given a topography image, as pursued by the training objective $F_2(p)$ (3.11), is not of primary concern for the topography model. Rather, its main application is to generate topographies that could lead to a given a cell image, which is relevant for *in silico* topography design (3.5). Consider the desired behavior of the model with respect to topography image generation. In the case when topographies influence the cell feature f, the informative and uninformative for the cell feature f factors of variation in topographies are encoded by z_f and l_ε respectively. Suppose a cell image x^i with a high value f^i is given. Consequently, by sampling $z_f \sim q_{\phi_f^*}(z_f|x^i)$ from the cell-model posterior, or $z_f \sim p_{\theta_f^*}(z_f|f^i)$ from the cell-model conditional prior, and by sampling $l_\varepsilon \sim p(l_\varepsilon)$ from the topography-model prior, the topography-model decoder $p_\vartheta(p|l_\varepsilon, z_f)$ should be able to generate different topography images that share in common the factor that leads to high values of f.

Conversely, in the case when topographies do not influence the cell feature f, the latent variable l_ε should encode *all* factors of variation in topographies, such that by sampling $z_f \sim q_{\phi_f^*}(z_f|x^i)$, or $z_f \sim p_{\theta_f^*}(z_f|f^i)$, and $l_\varepsilon \sim p(l_\varepsilon)$ the topography-model decoder is able to generate topographies that vary in all factors of variation observed in topography images. Notably, in the task of topography generation given a cell image, irrespective of whether topographies influence cell behavior, it is not expected that $q_{\phi_f^*}(z_f|x^i) \approx p_{\theta_{pr}^*}(z_f)$, since the cell model is trained in advance independently of the topography model, and its weights are fixed. The distribution of z_f is likely to be highly structured according to values of f, and the posteriors $q_{\phi_f^*}(z_f|x^i)$ are expected to deviate significantly from the marginal distribution, as shown in both Figures 3.11, 3.12. Therefore, it is the task of the topography-model decoder $p_\vartheta(p|l_\varepsilon, z_f)$ to learn to ignore the latent variable z_f in case of no influence of topographies on f.

In fact, the first goal of encoding the influence of topographies on cells pertains only to training of the encoder $q_{\varphi_f}(z_f|p)$. Whereas the second goal of learning to generate topographies conditioned on cell images is only affected by the decoder $p_\vartheta(p|l_\varepsilon, z_f)$ and by the second encoder $q_{\varphi_\varepsilon}(l_\varepsilon|p)$, which should learn the residual variation in topographies that is deemed uninformative for cell response f by the first encoder $q_{\varphi_f}(z_f|p)$. Clearly, different auxiliary objectives are needed to invoke the desired behavior of the model with respect to these goals.

Distance-based auxiliary objectives

Two perspectives on forcing the $q_{\varphi_f}(z_f|p)$ to learn the influence of topographies on cells can be considered. The first perspective is to minimize some distance measure D, e.g. KL-divergence, between the two posteriors, $q_{\varphi_f}(z_f|p)$ and $q_{\phi_f^*}(z_f|x)$ (3.22) with respect to parameters of the former, which can be implemented by including the negative D term in the original training objective $F_2(p)$ (3.11). An illustration of the idea behind this approach is provided in Figure 3.13. This perspective is directly motivated by the desired behavior of the model, described above: the encoder $q_{\varphi_f}(z_f|p)$ would be forced to discern between topographies on the basis of their impact on f, if the topographies in fact influence cells based on data. Similarly, the distance between the topography-model posterior $q_{\varphi_f}(z_f|p)$ and the cell-model conditional prior $p_{\theta_f^*}(z_f|f)$ (3.23) can be minimized, which has the same motivation, as shown in Figure 3.14. Incidentally, an example of the latter objective is using a conditional prior in the topography model (3.21) instead of the full prior, as discussed previously, where the KL-divergence is used as a measure of distance between two distributions.

$$q_{\varphi_f}(z_f|p) \approx q_{\phi_f^*}(z_f|x) \quad \rightarrow \quad \min_{\varphi_f} D(q_{\varphi_f}(z_f|p), q_{\phi_f^*}(z_f|x)) \tag{3.22}$$

$$q_{\varphi_f}(z_f|p) \approx p_{\theta_f^*}(z_f|f) \quad \rightarrow \quad \min_{\varphi_f} D(q_{\varphi_f}(z_f|p), p_{\theta_f^*}(z_f|f)) \tag{3.23}$$

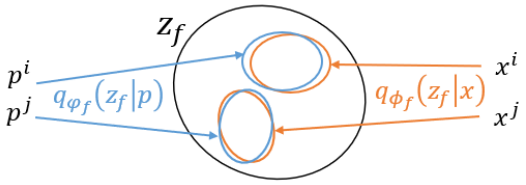

Figure 3.13: Distance-based auxiliary objective (1A).

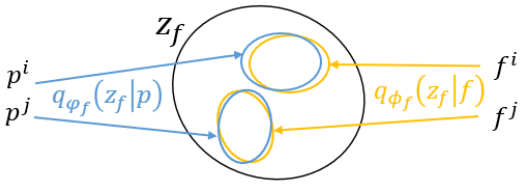

Figure 3.14: Distance-based auxiliary objective (1B).

Although intuitive, both auxiliary objectives (3.22) (3.23) could have a negative implication for the training process. The original training objective (3.11) of the topography model includes the term $KL\left(q_{\varphi_f}(z_f|p)\,||\,p_{\theta^*_{pr}}(z_f)\right)$ that forces the posterior to be close to the full prior, which is an approximate marginal distribution of z_f learned during training of the cell model. The KL-term would be in conflict with any of the proposed above auxiliary objectives, since together, original and auxiliary, they would restrict the same posterior distribution $q_{\varphi_f}(z_f|p)$ in a contradictory way. At the same time, if the KL-term is substituted with one of the auxiliary objectives, the model would no longer be instructed explicitly that by default all cell feature values f are possible for a given topography, unless the correlation between topographies and the cell feature f is found. In other words, the posterior distribution $q_{\varphi_f}(z_f|p)$ should by default match the full prior $p_{\theta^*_{pr}}(z_f)$, as shown in Figure 3.12, unless the influence of topographies on f is detected. An auxiliary objective $KL\left(q_{\varphi_f}(z_f|p)\,||\,q_{\phi^*_f}(z_f|x)\right)$ (3.22) or $KL\left(q_{\varphi_f}(z_f|p)\,||\,p_{\theta^*_f}(z_f|f)\right)$ (3.23) used as a substitute would transfer the same idea to the model in an implicit way: if the distance between distributions cannot be minimized, the posterior $q_{\varphi_f}(z_f|p)$ is likely to be wide, as shown in Figure 3.15. However, the posterior would not be instructed explicitly to match the full prior $p_{\theta^*_{pr}}(z_f)$ in that case. As a result, the cell-model decoder $p_{\theta^*}(x|z_\varepsilon, z_f)$ would potentially be unable to generate cells with a full range of f values by sampling $z_f \sim q_{\varphi_f}(z_f|p)$.

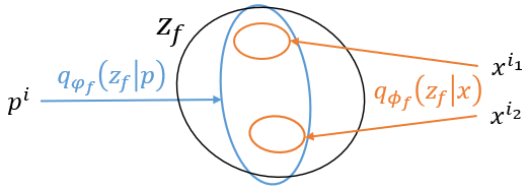

Figure 3.15: Distance-based auxiliary objective (1A): no influence case.

Likelihood-based auxiliary objectives

Suppose that $KL\left(q_{\varphi_f}(z_f|p)\,||\,p_{\theta^*_{pr}}(z_f)\right)$ term in the original training objective of the topography model (3.11) remains to enforce the full prior $p_{\theta^*_{pr}}(z_f)$ as the default output of the encoder. However, it is still necessary to encourage the encoder $q_{\varphi_f}(z_f|p)$ to capture the influence of topographies on cells. An alternative view on the problem is that the encoder should operate in a way that maximizes the likelihood of the cell images from the dataset. The idea is that a cell image $\hat{x}^i$ generated by the cell-model decoder $p_{\theta^*}(x|z_\varepsilon, z_f)$ when provided with a sample $z_f \sim q_{\varphi_f}(z_f|p^i)$ from the topography-conditioned posterior should be similar to the original cell image x^i from a training pair (p^i, x^i) with respect to affected cell features f. An illustration of that perspective is shown in Figure 3.16. Maximizing this auxiliary objective (3.24) in combination with the negative KL-term in (3.11) encourages exactly the desired behavior of the topography-model encoder: the KL-term forces the posterior $q_{\varphi_f}(z_f|p)$ to match the full prior $p_{\theta^*_{pr}}(z_f)$ by default, however, the

posterior is allowed to deviate from the full prior in a way that increases the likelihood of the simulated cell images.

$$\max_{\varphi_f} \quad \mathbb{E}_{p(z_\varepsilon)q_{\varphi_f}(z_f|p)} \; log\, p_{\theta^*}(x|z_\varepsilon, z_f) \tag{3.24}$$

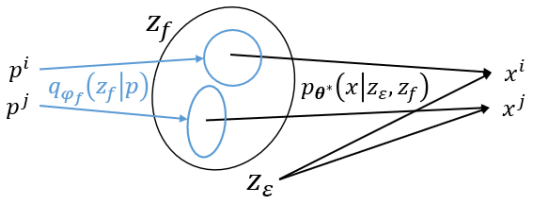

Figure 3.16: Likelihood-based auxiliary objective (2A).

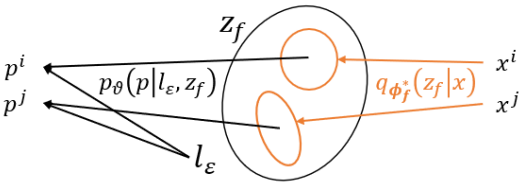

Figure 3.17: Likelihood-based auxiliary objective (2B).

A similar likelihood-based auxiliary objective can be derived to address the second challenge of disentanglement of the topography-model latent space, as illustrated in Figure 3.17 and formulated in (3.25). The idea of this objective is inverse to (3.24): by sampling $z_f \sim q_{\phi_f^*}(z_f|x^i)$ from the cell-model posterior corresponding to a cell image x^i, the topography model decoder $p_\vartheta(p|l_\varepsilon, z_f)$ aims improve the similarity of generated topographies to the topography p^i from the training pair (p^i, x^i) with respect to those factors of variation in topographies that led to the cell response f. To do that, the topography-model decoder $p_\vartheta(p|l_\varepsilon, z_f)$ aims to utilize the information about the inferred cell response encoded in $z_f \sim q_{\phi_f^*}(z_f|x^i)$, while at the same time combining z_f with an arbitrary sample $l_\varepsilon \sim p(l_\varepsilon)$ from the prior, since the goal is to generate *different* topographies that could have led to a given cell image, which share in common only the factors that invoked certain cell response f. It should be mentioned that the conditional prior $p_{\theta_f^*}(z_f|f)$ can be used instead of the posterior $q_{\phi_f^*}(z_f|x)$ in the auxiliary objective 2B (3.25) with a similar motivation, however it does not provide any hypothetical benefits as compared with the proposed auxiliary objective.

$$\max_{\vartheta} \quad \mathbb{E}_{p(l_\varepsilon)q_{\phi_f^*}(z_f|x)} \; log\, p_\vartheta(p|l_\varepsilon, z_f) \tag{3.25}$$

Noteworthy, the likelihood term from the original training objective (3.11), provided below in (3.26), and both likelihood-based auxiliary objectives (3.24) (3.25) are not mutually exclusive or redundant. The goal of the term from the original objective (3.26) is to train both encoders $q_{\varphi_l}(l_\varepsilon|p)$, $q_{\varphi_f}(z_f|p)$ and the decoder $p_\vartheta(p|l_\varepsilon, z_f)$ to reconstruct a topography image irregardless of the connection between the datasets P and X, i.e. it makes no distinction between the latent variables z_f and l_ε. The auxiliary objective 2A (3.24) aims to train only the encoder $q_{\varphi_f}(z_f|p)$, since the weights of the cell-model decoder $p_{\theta^*}(x|z_\varepsilon, z_f)$ are fixed. Its goal is to force the encoder to learn the influence of topographies on cell images via the latent variable z_f corresponding to a cell feature f. Whereas the auxiliary objective 2B (3.25) aims to train the decoder $p_\vartheta(p|l_\varepsilon, z_f)$; it forces the decoder to extract information from the latent variable z_f, sampled from the cell-model posterior $q_{\phi_f^*}(z_f|x)$, that can be useful in generating topographies associated with a given cell response. Furthermore, the combination of the two auxiliary objectives 2A (3.24), 2B (3.25) cannot substitute the original likelihood-term (3.26), since only in it the residual encoder $q_{\varphi_l}(l_\varepsilon|p)$ is trained.

$$\max_{\varphi_f, \varphi_l, \vartheta} \quad \mathbb{E}_{q_{\varphi_l}(l_\varepsilon|p)q_{\varphi_f}(z_f|p)} \; log\, p_\vartheta(p|l_\varepsilon, z_f) \tag{3.26}$$

Alternative training schemes for the likelihood-based auxiliary objectives

Once all the training objectives are defined, the training procedure of the topography model is to be discussed, a summary of which is provided in the section (3.3.1). It is proposed to train the encoder $q_{\varphi_f}(z_f|p)$ prior to training of the other two components of the topography model: the encoder $q_{\varphi_l}(l_\varepsilon|p)$, and the decoder $p_\vartheta(p|l_\varepsilon, z_f)$. The idea behind this decision is that the goal of the encoder $q_{\varphi_f}(z_f|p)$ is to only capture the information about topographies p^i that improves the knowledge of which cell images x^i could be observed in response. Whereas the uninformative for cell response f variation in topographies, but necessary for topography generation, should be captured by the latent variable l_ε as a residual. It is not desired to train the encoder $q_{\varphi_f}(z_f|p)$ in the tasks of topography reconstruction (3.26) and topography generation for a given cell image (3.25). In fact, $q_{\varphi_f}(z_f|p)$ is only intended to be trained in the task of cell image generation conditioned on a topography, i.e. simulation of the experiment. To enforce the desired behavior of the encoder, the following maximization objective is formulated (3.27). Importantly, the KL-term from the original objective (3.11) imposing the full prior $p_{\theta^*_{pr}}(z_f)$ as the 'default' posterior distribution is introduced specifically in this step, since it only affects the encoder under consideration.

$$F_{2A}(x,p) = \mathbb{E}_{p(z_\varepsilon)q_{\varphi_f}(z_f|p)}\ log\, p_{\theta^*}(x|z_\varepsilon, z_f) \;-\; \beta_{pf}\, KL\left(q_{\varphi_f}(z_f|p)\,||\,p_{\theta^*_{pr}}(z_f)\right) \tag{3.27}$$

It should be mentioned that training of the encoder $q_{\varphi_f}(z_f|p)$ using the proposed objective (3.27) involves a decision that has not been discussed. In the current setting the prior distribution $p(z_\varepsilon)$ is used to sample z_ε for cell image generation. However, an alternative approach would be to sample $z_\varepsilon \sim q_{\phi^*_\varepsilon}(z_\varepsilon|x)$ from the cell-model posterior. The two alternative training schemes are illustrated in Figures 3.18, 3.19, where the trained encoder is shown in red.

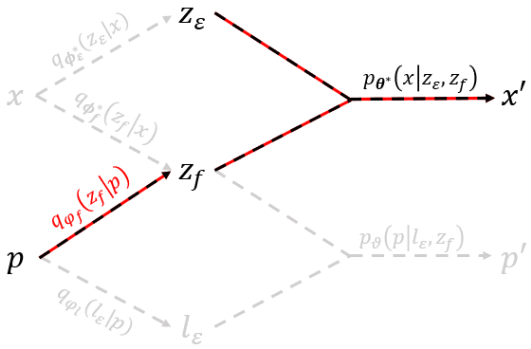

Figure 3.18: Conditional cell image generation objective (2A): sample from $p(z_\varepsilon)$.

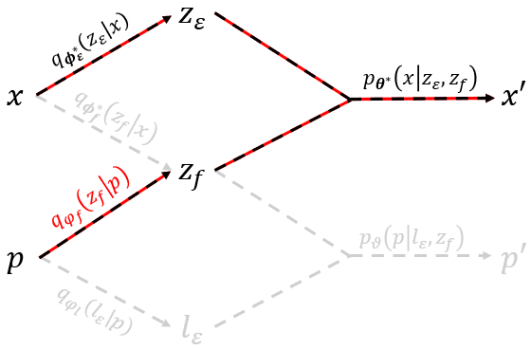

Figure 3.19: Conditional cell image generation objective (2Ax): sample from $q_{\phi^*_\varepsilon}(z_\varepsilon|x)$.

Arguably, since the weights of the cell-model encoder $q_{\phi^*_\varepsilon}(z_\varepsilon|x)$ and of the cell-model decoder $p_{\theta^*}(x|z_\varepsilon, z_f)$ are fixed, sampling from the posterior $q_{\phi^*_\varepsilon}(z_\varepsilon|x)$ would not affect the flow of gradients of the objective with respect to the weights of the target encoder $q_{\varphi_f}(z_f|p)$ and, thus, cannot compromise the training process. At the same time sampling $z_\varepsilon \sim q_{\phi^*_\varepsilon}(z_\varepsilon|x)$ from the posterior would increase the likelihood of the generated cell images, since the latent variable z_ε encodes additional "noise" variation in cell images. For instance, z_ε could encode information about cell orientation. Then, the generated samples would have the same orientation as the cell image from the training pair (p^i, x^i). On the contrary, when sampling from the prior $p(z_\varepsilon)$ the cell-model decoder would generate cell images with arbitrary orientation, which would negatively affect the likelihood of such images, even if the encoder $q_{\varphi_f}(z_f|p)$ transfers the information about the induced cell feature value f through z_f successfully. Therefore, the previously proposed objective (3.27) is modified, such that the prior distribution $p(z_\varepsilon)$ is substituted for $q_{\phi^*_\varepsilon}(z_\varepsilon|x)$, as shown in (3.28). After optimization of the current objective, the weights of the encoder are fixed: $q_{\varphi^*_f}(z_f|p)$.

$$F_{2Ax}(x,p) = \mathbb{E}_{q_{\phi^*_\varepsilon}(z_\varepsilon|x)q_{\varphi_f}(z_f|p)}\ log\, p_{\theta^*}(x|z_\varepsilon, z_f) \;-\; \beta_{pf}\, KL\left(q_{\varphi_f}(z_f|p)\,||\,p_{\theta^*_{pr}}(z_f)\right) \tag{3.28}$$

Further, given that the encoder $q_{\varphi^*_f}(z_f|p)$ is trained as a result of maximizing the objective (3.28) and its weights are fixed, it remains to train the second encoder $q_{\varphi_l}(l_\varepsilon|p)$ and the decoder $p_\vartheta(p|l_\varepsilon, z_f)$ using the two previously introduced objectives: original (3.26), and auxiliary 2B (3.25). As discussed above, the original objective (3.26) pertains to the task of topography reconstruction and aims to train both the encoder $q_{\varphi_l}(l_\varepsilon|p)$ and the decoder $p_\vartheta(p|l_\varepsilon, z_f)$. Whereas the auxiliary objective 2B (3.25) addresses the task of topography generation for a given cell image and, in the provided formulation, aims to train only the decoder. However, the objective 2B can be potentially modified similarly to the objective 2A (3.25), such that l_ε would be sampled from the posterior $q_{\varphi_l}(l_\varepsilon|p)$ instead of the prior $p(l_\varepsilon)$. As a result, both $p_\vartheta(p|l_\varepsilon, z_f)$ and $q_{\varphi_l}(l_\varepsilon|p)$ would be involved in the training process. Therefore, the remaining questions are 1. whether the two objectives (3.26), 2B (3.25) should be trained in combination or consecutively, and 2. whether l_ε in the auxiliary objective should be sampled from $p(l_\varepsilon)$ (2B, Figure 3.20), or from $q_{\varphi_l}(l_\varepsilon|p)$ (2Bp, Figure 3.21). These issues are discussed in the following paragraph.

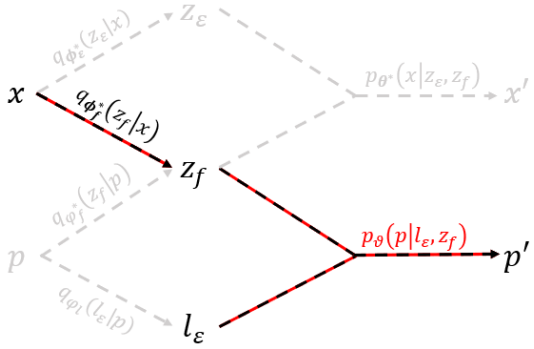

Figure 3.20: Conditional topography generation objective (2B): $l_\varepsilon \sim p(l_\varepsilon)$.

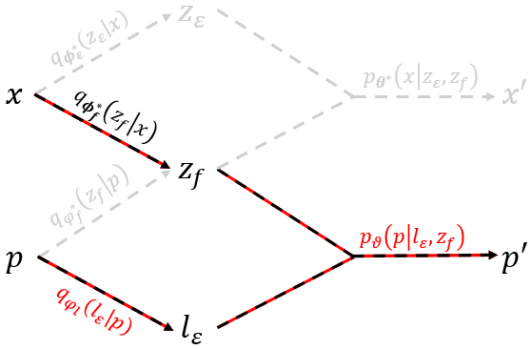

Figure 3.21: Conditional topography generation objective (2Bp): $l_\varepsilon \sim q_{\varphi_l}(l_\varepsilon|p)$.

Consider the case when the two objectives (3.26), 2B (3.25) are trained consecutively with the intention to fix the weights of either the encoder $q_{\varphi_l}(l_\varepsilon|p)$ or the decoder $p_\vartheta(p|l_\varepsilon, z_f)$ after the first step. Suppose that the original objective (3.26) is trained first and consider its training scheme, as shown in Figure 3.22. In this case the task is only to maximize the reconstruction quality for a given topography. Since the fixed encoder $q_{\varphi^*_f}(z_f|p)$ captures only the information related to cell response, the second encoder $q_{\varphi_l}(l_\varepsilon|p)$ is likely to learn to extract *all* the information about a given topography in order to increase the likelihood, unless l_ε is heavily restricted in capacity. Accordingly, the decoder is likely to ignore the latent variable z_f, while paying attention only to l_ε, which would result in a plain VAE model. Alternatively, suppose that 2B (3.25) is trained first and consider its training scheme, as shown in Figure 3.20. This objective is aimed to only train the decoder $p_\vartheta(p|l_\varepsilon, z_f)$ with the intention to fix its weights later. Since l_ε is sampled from the prior $p(l_\varepsilon)$, it would not carry any information about topographies, and the decoder is likely to ignore it. Therefore, it would be impossible to subsequently train the encoder $q_{\varphi_l}(l_\varepsilon|p)$ with the fixed decoder. Hence, the training the objectives (3.26), 2B (3.25) consecutively in any order is not expected to result in the desired behavior of the model.

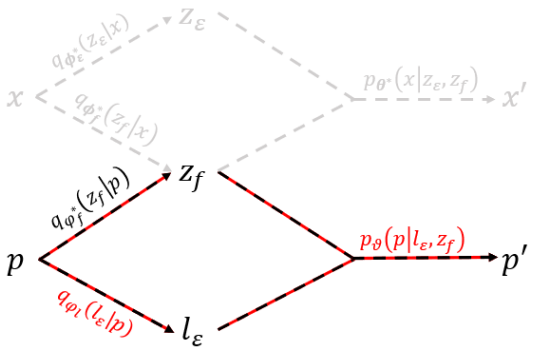

Figure 3.22: Topography reconstruction objective with fixed weights of the encoder $q_{\varphi^*_f}(z_f|p)$.

To circumvent the problem that only the decoder is trained by the auxiliary objective 2B (3.25), the modified objective 2Bp can be considered, where l_ε is sampled from the posterior $q_{\varphi_l}(l_\varepsilon|p)$ instead of the prior $p(l_\varepsilon)$, as reflected by the training scheme in Figure 3.21. Suppose that the modified objective 2Bp is trained first. Then the situation would be equivalent from the training perspective to optimizing the original objective, as shown in Figure 3.22, and would again result in a plain VAE with l_ε capturing all the information about topographies. Hence, it is concluded that 1. the objectives (3.26) and 2B (or 2Bp) should not be trained consecutively, and 2. sampling from the posterior $q_{\varphi_l}(l_\varepsilon|p)$ (2Bp) distribution instead of the prior $p(l_\varepsilon)$ (2B) devalues the utility of the auxiliary objective.

Based on the provided reasoning, it is proposed to train the model for the tasks of topography reconstruction (Figure 3.22) and conditional topography generation (Figure 3.20) in a single training objective, as formulated in (3.29) below, simultaneously training the decoder $p_\vartheta(p|l_\varepsilon, z_f)$ and the encoder $q_{\varphi_l}(l_\varepsilon|p)$. The goal of the model at this step is to find a balance between the quality of topography reconstruction and the ability to capture the cell-topography relationship. The first term aims only to reconstruct a given topography image based on its latent representation (l_ε, z_f), while the second term forces the decoder to utilize the information about the induced cell feature values, captured by z_f. Thereby, the second term can be seen as a regularization that restricts the model from developing a plain VAE model, in case topographies influence the cell response based on data. To control the balance between the terms, the hyperparameter η is introduced. Furthermore, the KL-term related to the latent variable l_ε from the original objective (3.11) is added at this step.

$$F_{orig,2B}(x,p) = \overbrace{\mathbb{E}_{q_{\varphi_l}(l_\varepsilon|p)q_{\varphi^*_f}(z_f|p)}\ log\, p_\vartheta(p|l_\varepsilon, z_f)}^{\text{Topography reconstruction}} + \eta \overbrace{\mathbb{E}_{p(l_\varepsilon)q_{\phi^*_f}(z_f|x)}\ log\, p_\vartheta(p|l_\varepsilon, z_f)}^{\text{Conditional topography generation}} - \beta_l\, KL\left(q_{\varphi_l}(l_\varepsilon|p)\,||\,p(l_\varepsilon)\right) \tag{3.29}$$

3.5 Application scenarios

The key application scenarios of the introduced model include 1. simulation of the cell-surface topography experiment (*in silico* experiment) and 2. *in silico* topography design. Simulation of the experiment implies generating different cell images x that could be observed according to the model on a given surface topography p, represented by an image. Whereas *in silico* topography design is the inverse task, when different topographies p should be generated that could potentially lead to a given (desired) cell response. In the second scenario, the cell response can be expressed either as a cell image x, or as a value of the cell feature of interest f.

1. *In silico* experiment.

To generate cell images x for a given topography p, the topography is firstly processed with the topography-model encoder $q_{\varphi_f^*}(z_f|p)$, which outputs the parameters of the posterior distribution in the Z_f latent subspace. Secondly, samples $z_f \sim q_{\varphi_f^*}(z_f|p)$ from the posterior and $z_\varepsilon \sim p(z_\varepsilon)$ from the cell-model prior are passed on to the cell-model decoder $p_{\theta^*}(x|z_\varepsilon, z_f)$, which maps them to a distribution, or a 'region', in the space of cell images $p(x|z)$, as shown in Figure 3.23. Typically, a single cell image is generated from the decoder's output. In order to generate visually different cell images, which share in common the characteristics induced by a given topography, additional samples z_f, z_ε should be taken and processed with the decoder.

2.1. *In silico* topography design.

To generate topography images p for a given cell image x, the model is exploited in the opposite direction, as demonstrated in Figure 3.24. The cell image x is mapped by the encoder $q_{\phi_f^*}(z_f|x)$ to a posterior distribution of z_f. Subsequently, samples $z_f \sim q_{\phi_f^*}(z_f|x)$ from the posterior and $l_\varepsilon \sim p(l_\varepsilon)$ from the topography-model prior are used by the topography-model decoder $p_{\vartheta^*}(p|l_\varepsilon, z_f)$ to produce a topography image. Similarly to the first application scenario, to obtain visually different topographies that could lead to a given cell response, different samples z_f, l_ε are taken.

2.2. *In silico* topography design based on a cell feature value.

As an extension to the previous application scenario, the model can take a value of the cell feature f, as a representation of the cell response (Figure 3.25). In this case, samples $z_f \sim p_{\theta_f^*}(z_f|f)$ from the conditional prior are used by the decoder. Notably, in the presence of multiple cell features, z_f for irrelevant cell features is sampled from the full prior distribution $p_{\theta_{pr}^*}(z_f)$.

(*) Predicting cell feature distribution based on a topography.

It should be noted that the proposed model also allows for the base application scenario, considered in the literature, where the value of a cell feature is predicted based on a topography. To achieve this, samples $z_f \sim q_{\varphi_f^*}(z_f|p)$ from the posterior distribution, corresponding to a given topography p, are passed to the auxiliary regressor (or classifier) $q_{\omega_f}(f|z_f)$. Furthermore, by sampling z_f many times, it is possible to derive a distribution over f values, which allows to evaluate the level of uncertainty of the model regarding the predicted (expected) f value for a given topography p.

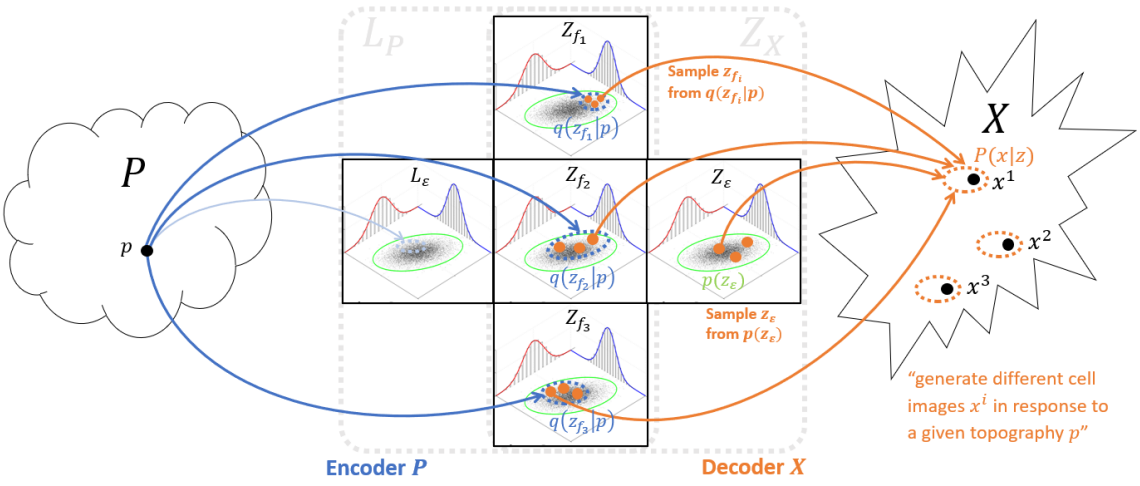

Figure 3.23: Model: *in silico* experiment.

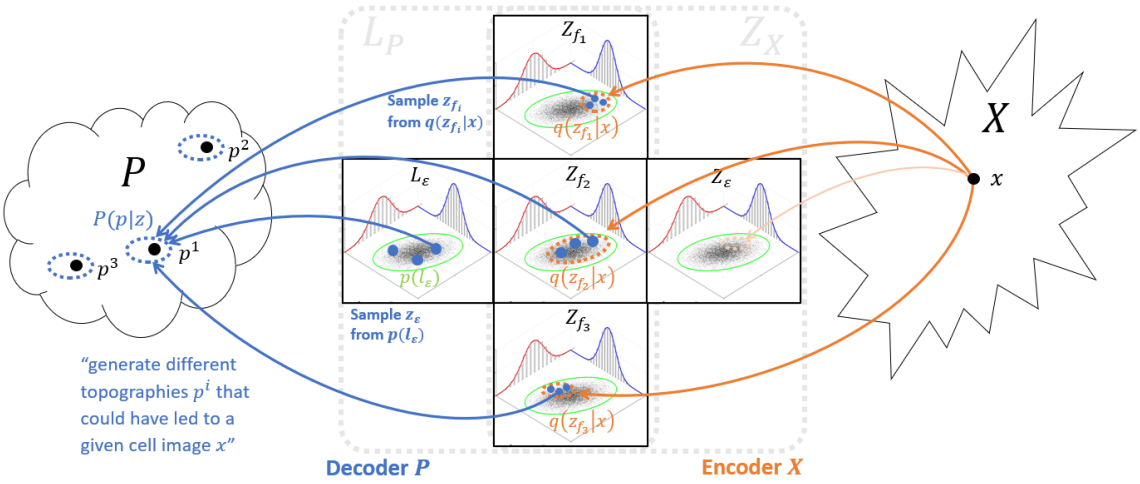

Figure 3.24: Model: *in silico* topography design.

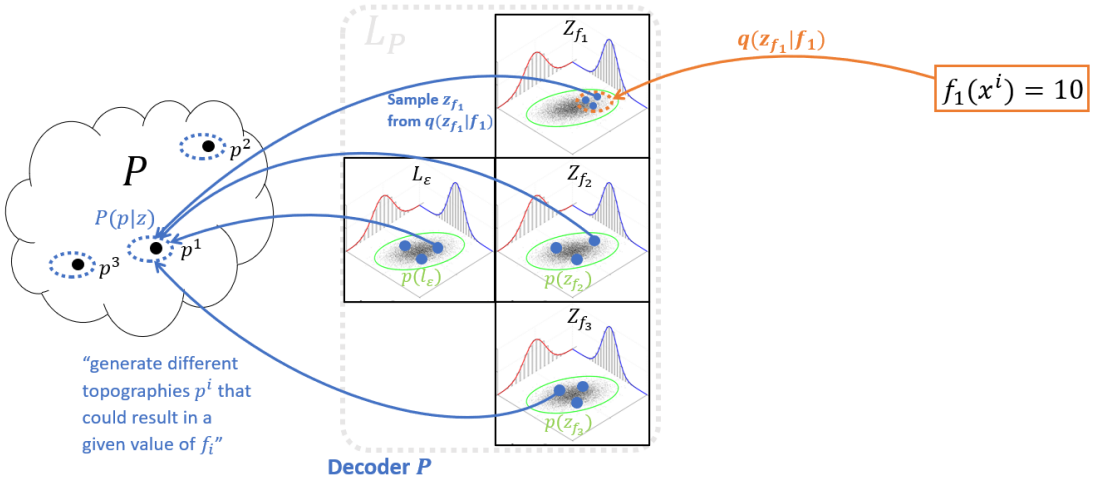

Figure 3.25: Model: *in silico* topography design given f.

Chapter 4

Data

In this chapter two datasets are described, on which the proposed approach was tested. The first dataset, ToyCell, is a synthetic dataset created as part of the present work. It consists of artificial cell-resembling images and artificial topography-resembling images, all with 128*128 pixel resolution and three color channels. The ALP screening dataset [20] [55] is a real-world dataset containing images of human mesenchymal stem cells (hMSCs) and images of algorithmically generated topographies, produced on a TopoChip [50], to which the cells were exposed. The preprocessed images of both topographies and individual cells in the ALP screening dataset have 64*64 pixel resolution and three color channels.

4.1 ToyCell synthetic dataset

The ToyCell dataset comprises 50,000 synthetic images of cells and 50,000 synthetic images of topographies. Examples of cell and topography images are provided in Figures 4.1, 4.2 respectively.

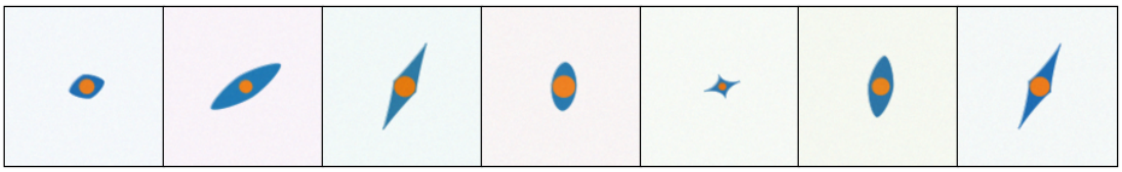

Figure 4.1: ToyCell dataset: cell image examples

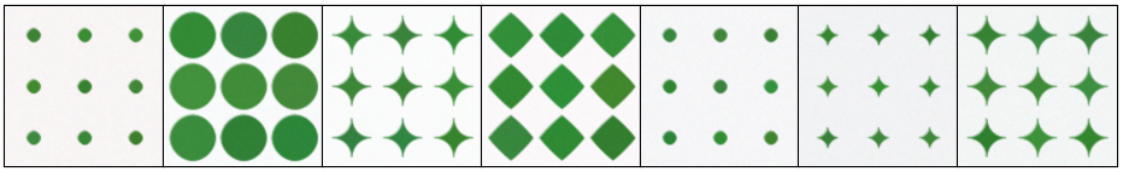

Figure 4.2: ToyCell dataset: topography image examples

A synthetic cell image contains an outer blue shape intended to represent a cell cytoskeleton, and an inner orange shape that represents cell nucleus. A cell image is defined by four features: roundness (f_1), elongation (f_2), nucleus size (f_3) and rotation angle (f_4). The contour of the cytoskeleton is defined by the formula (4.1), and the cell nucleus is a circle defined using the formula (4.2). A cell image generated for given values of f_1, f_2, f_3 is rotated counterclock-wise from the vertical position by f_4 degrees. At creation all features were randomly drawn from the respective uniform distributions with ranges as shown in Table 4.1. Notably, the parameter k in the nucleus size range approximately equals the maximum possible radius of the nucleus circle, such that it remains inside the cytoskeleton shape and depends on the minimum roundness value[1].

$$y = \pm\, f_2 \left|1 - x^{f_1}\right|^{1/f_1}, \quad x \in [-1, 1] \tag{4.1}$$

$$y = \pm \left({f_3}^2 - x^2\right)^{1/2}, \quad x \in [-f_3, f_3],\ f_3 < 1 \tag{4.2}$$

[1] $k_{max} = \max_x \sqrt{2x^2}$ s.t. $(2x^{f_1})^{1/f_1} \leq 1 \ \Rightarrow k_{max} = 2^{\frac{1}{2}-\frac{1}{f_1}} \approx 0.354$ for $f_1 = 0.5$

Variable	Range
Roundness (f_1)	[0.5, 2]
Elongation (f_2)	[1, 5]
Nucleus size (f_3)	[0.4k, 0.9k], k=0.35
Rotation angle (f_4)	[0, 179] degrees

Table 4.1: ToyCell dataset: ranges of the cell image design parameters.

According to the proposed approach described in Chapter 3 a cell image originates from a combination of visually discernable, measurable and independent cell features and residual variation, and topographies may influence cells through these features. For the synthetic dataset the first three features f_1, f_2, f_3 are chosen to be the cell features of interest, while the rotation angle f_4 is assumed to be an irrelevant noise feature. Furthermore, at creation of the dataset the colors of the cytoskeleton, nucleus and the background color are slightly randomized and a Gaussian filter ($\sigma = 0.6$) is applied to images to add noise variation. The contribution of each of the features to the resulting cell image is visualized in Figure 4.3, such that each feature is increased from the minimum to the maximum value from left to right in the corresponding row, while the rest of the features are taken at random.

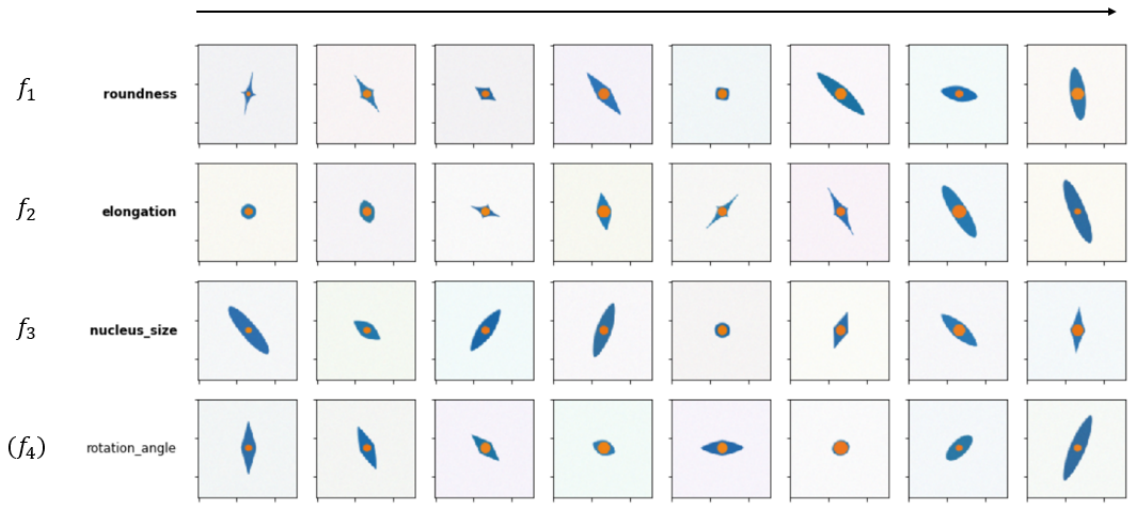

Figure 4.3: ToyCell dataset: cells

A topography image contains nine identical shapes placed in a grid, and only two parameters define a singe topography image: roundness (g_1) and radius (g_2), which are implemented in a similar way to cell images. The contribution of topography features g_1, g_2 to the resulting topography image is visualized in Figure 4.4.

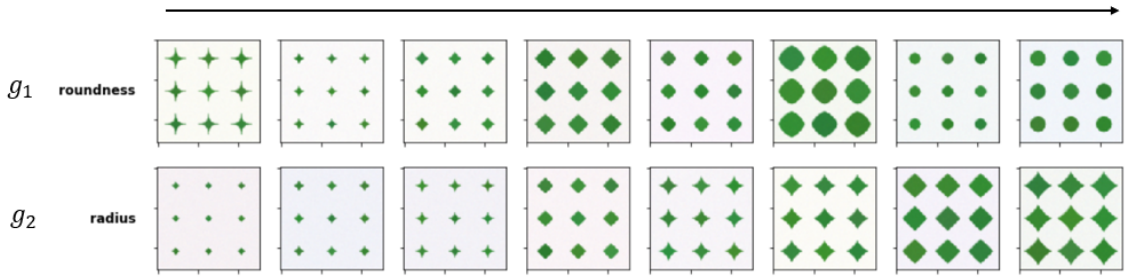

Figure 4.4: ToyCell dataset: topographies

The ranges of the uniform distributions used to generate feature values of the topography images are presented in Table 4.2. Similarly to cell images, the background color and the color of the topography shape are slightly randomized, and a Gaussian filter ($\sigma = 0.6$) is applied to images.

Variable	Range
Roundness (g_1)	[0.5, 2]
Radius (g_2)	[0.2, 0.9]

Table 4.2: ToyCell dataset: ranges of the topography image design parameters.

4.1.1 Artificial relationship between topographies and cells.

Both cell and topography images, generated as described above, are initially not related. Therefore, to verify the proposed approach it is necessary to establish some artificial relationship between topography images and cell images and subsequently unravel this relationship using the model. Importantly, to be compatible with the proposed approach, an artificial relationship should imply that topographies influence one or several of the cell features of interest: roundness (f_1), elongation (f_2) and nucleus size (f_3). Accordingly, in this paper it is assumed that the radius of a topography (g_2) positively influences the elongation of a cell (f_2).

In order to create artificial training pairs (topography image, cell image) with positive correlation between g_2 and f_2, the following procedure is used. Firstly, the topography image data table is sorted by g_2 in ascending order, and the cell image data table is sorted by f_2 in ascending order; both tables have size 50,000, equal to the number of images in each dataset. Secondly, a sliding window having width[2] $w = 1000$ and step 1 is propagated through both tables in parallel. At each position, a single row index inside the sliding window is taken at random independently for each of the tables, and the two selected rows form a training pair. An illustration of this procedure is provided in Figure 4.5. As a result, the training data table consists of 50,000 rows, where each row contains values of all features $f_1, f_2, f_3, f_4, g_1, g_2$ and image file names. Examples of training pairs are shown in Figure 4.6.

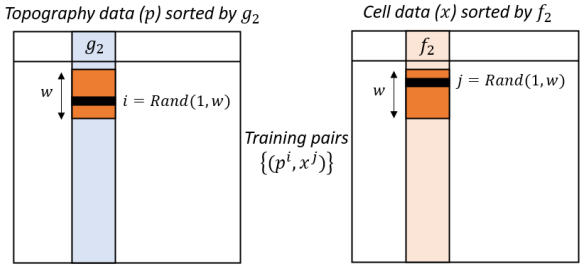

Figure 4.5: Creating artificial training pairs $\{p^i, x^i\}$.

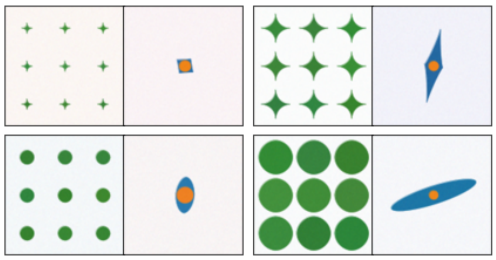

Figure 4.6: Artificial relationship: examples of training pairs.

Additionally, a control-case training data table was created by randomly matching images of topographies and cells. The relationship between g_2 and f_2 in the main and control training data tables is shown in Figures 4.7, 4.8 respectively.

[2]The width w of the sliding window regulates the variance of the dependency.

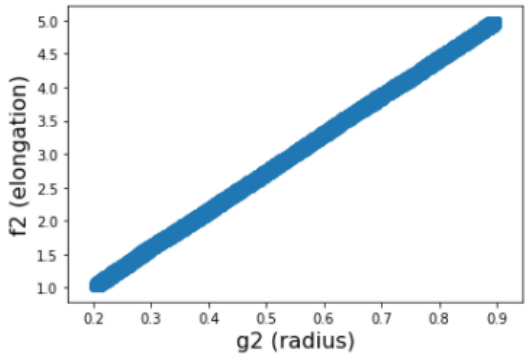

Figure 4.7: Relationship between g_2 and f_2 in the main training data table.

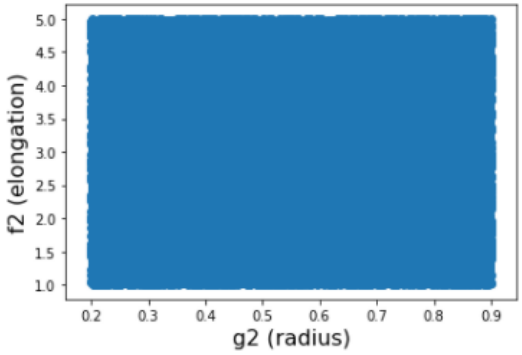

Figure 4.8: Relationship between g_2 and f_2 in the control training data table.

4.2 ALP screening real dataset

The ALP screening dataset [20] [55] contains images of individual human mesenchymal stem cells (hMSCs) exposed to a library of surface topographies produced on a TopoChip [50] and images of topographical features (Figure 2.1) corresponding to these topographies (Figure 2.2). The dataset comprises 38,051 unique cell images captured on 2147 unique topographies out of 2177 in the initial library. The discrepancy in the size of the cell and topography image datasets is explained by the fact that a single topography is duplicated on a chip, and a single instance of a surface topography on a chip yields a median number of 8 cell images. Notably, only the *cytoskeleton layer* of cell images was used in the present work, i.e. cropped cell images have single color channel (intensity). Examples of cell images taken as a basis in further preprocessing are provided in Figure 4.9.

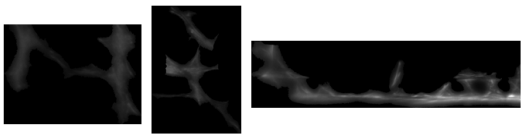

Figure 4.9: Original cropped cell images: examples.

The original cropped cell images have different size, however the relative size is preserved. To preprocess the data, cell images were centered in a square with side length equal to the longest dimension of a cell image in the dataset. Subsequently, single-channel cell images were colored for visualization purposes. Notably, two variants of cell image representation were considered: intensity information-preserving and binary, as shown in Figure 4.10; the binary version was chosen as less complex for a model. Finally, the images were resized to 64*64 pixels.

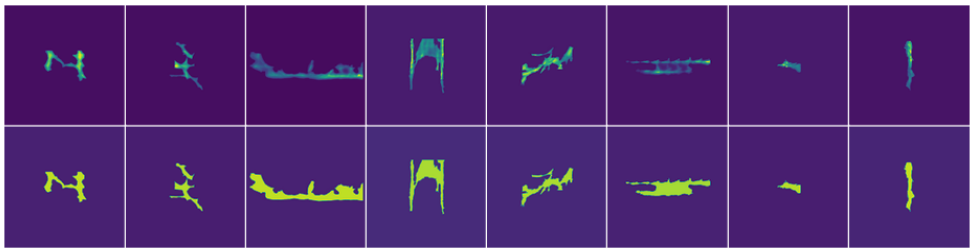

Figure 4.10: ALP screening dataset after preprocessing: cell images examples. Top row: centered and colored with intensity information preserved. Bottom row: centered, converted to binary images and colored (chosen).

Cell images are described by a set of numerical features extracted by CellProfiler [50] software from raw screening images. These include Area, Eccentricity, Solidity, Perimeter, Compactness, Euler Number, Extent, Form Factor, Orientation, Major Axis Length, Minor Axis Length and other features, all defined in [21]. Notably, these features are correlated, therefore they cannot be used all at the same time to represent independent z_f subspaces. A visualization of the dataset with respect to several evident features is provided below in Figure 4.11. In the present work only the Area feature is considered to be a cell feature of interest, hence a single z_f latent variable corresponds to Area, while all the residual variation is to be captured by z_ε.

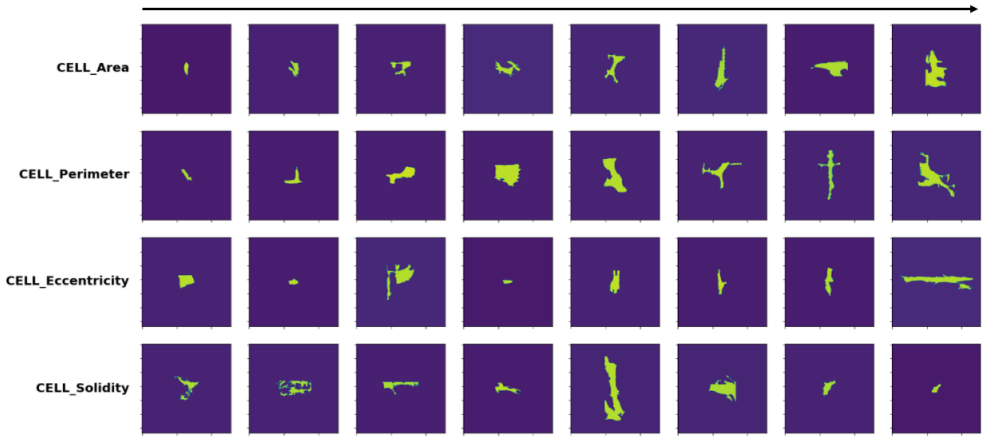

Figure 4.11: ALP screening dataset: cell images by features.

Topography images were created based on the images of topographical features. Topographical features have three size categories: 100*100 μm (micrometers), 200*200 μm and 280*280 μm. A single surface topography has a size of 2800*2800 μm and is formed by repetition of topographical features in a grid: 28 by 28 for features of size 100*100 μm, 14 by 14 for features of size 200*200 μm, and 10 by 10 for features of size 280*280 μm. Since quarters of a full topography image are identical, they can be used to represent topography images instead of full topography images. Examples of quarter-topography images for different size categories are provided in Figure 4.12 in the bottom row using the same color scheme as for cell images.

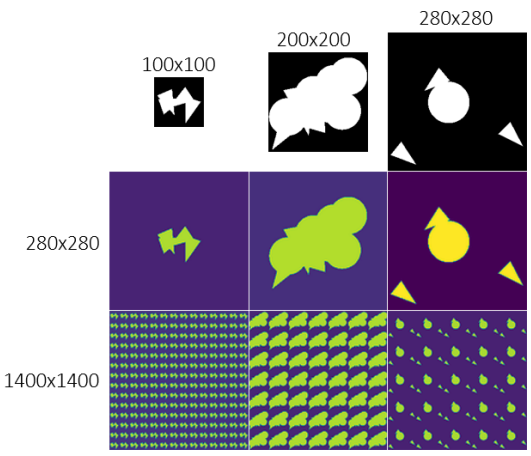

Figure 4.12: Creating topography images: examples for different size categories of topographical features.

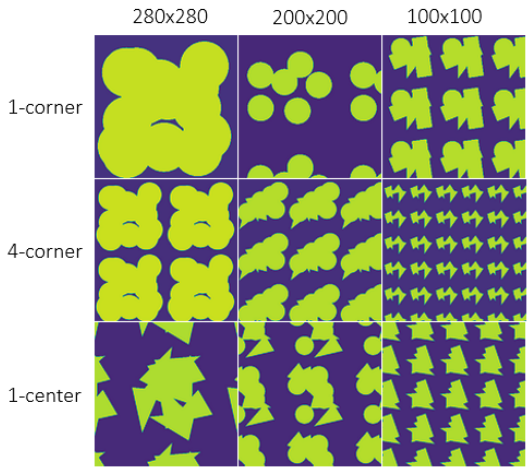

Figure 4.13: Topography representation variants: 1-corner (by the largest-size topographical feature), 4-corner, 1-center, etc.

In comparison to the synthetic dataset, individual topographical features are poorly visible in an image representing a quarter of a single topography, and especially this holds for the smallest-size 100*100 μm features. To emphasize the design properties of topographical features defining a topography it could be useful to zoom in the topography images. However, the information about the space forming on a topography between the repeated instances of a topographical feature might be useful for the model as well, and therefore images with only one instance of a topographical feature (middle row in Figure 4.12) are likely not suitable. A number of alternative representations can be considered, such as those shown in Figure 4.13. Importantly, the relative size of topographical features should be preserved in order for a model to distinguish between the size categories. Therefore, different number of topographical feature instances for different size categories would be seen in any relative size-preserving representation, and possibly non-integer quantity. For example, in "4-corner" representation (middle row in Figure 4.13), the left upper corner of a topography, corresponding to 560*560 μm is taken, such that only four largest-size topographical features are visible, while more topographical feature instances of other sizes are seen on a same-size part of a topography, some of which are seen partially. In the present work, the "4-corner" representation is chosen as the default one.

Topographies are described by a set of numerical features, which are defined in [21] along with cell features. A total number of 38 features are available in the ALP screening dataset. Many of these features are poorly interpretable visually, however, a few can be visualized in the same manner as cell images above. In Figure 4.14 7 features describe the variation in topography images. 'FeatSize' corresponds to the size category of a topographical feature. The features 'LA', 'CA', 'TA' represent the total area occupied by line, circle and triangle primitive shapes relative to the total area of a topographical feature, respectively. Similarly, the features 'DL', 'DC', 'DT' represent the number of such primitive shapes relative to the total area of a topographical feature.

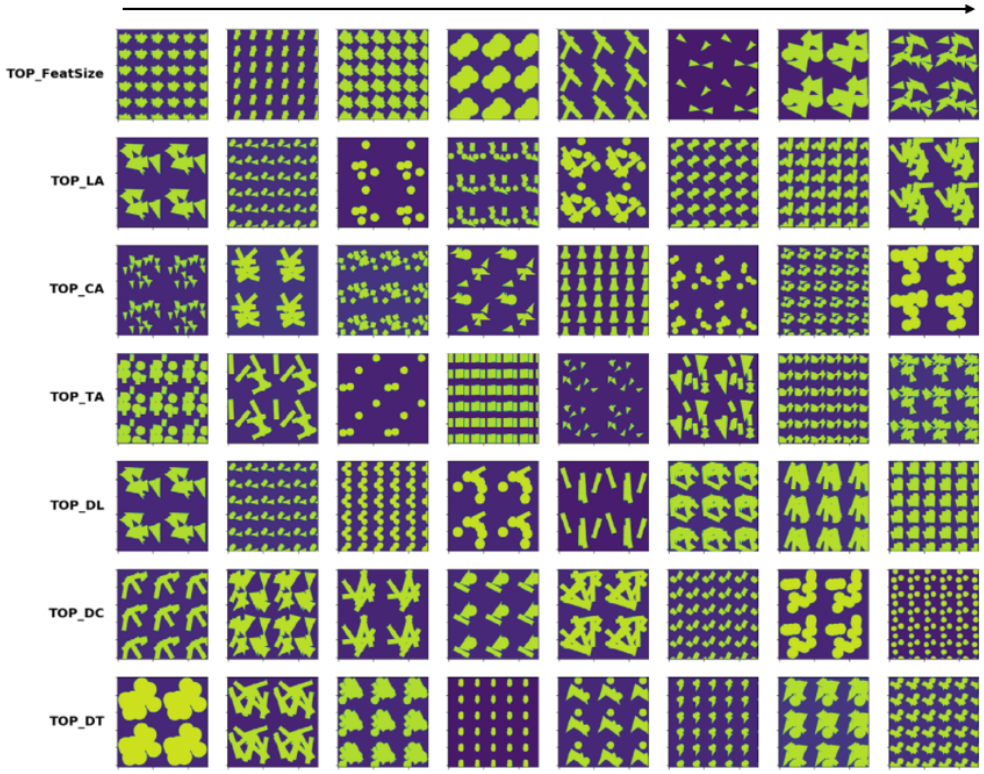

Figure 4.14: ALP screening dataset: topography images by features.

Chapter 5

Experiments and results

In this chapter the experimental results on both datasets are reported. All three steps of the proposed approach, as described in the training procedure summary 3.3.1, are followed successively, and the respective results are provided. Furthermore, the model is tested in application scenarios, as described in 3.5. The architectural choices for both models and the details of the training procedure are provided in Appendix A.

5.1 Experiments on the ToyCell dataset

In the first step, the goal is to derive a disentangled latent representation of cell images, such that three independent latent subspaces z_{f_i} correspond to the cell features of interest: z_{f_1} for roundness (f_1), z_{f_2} for elongation (f_2) and z_{f_3} for nucleus size (f_3), while the noise subspace z_ε should capture the residual variation in cell images, mostly explained by rotation angle (f_4). In the second step, the goal is to train the topography-model encoders $q_{\varphi_f}(z_f|p)$, which should unravel the artificial positive relationship between the topography radius (g_2) feature and cell elongation (f_2). Finally, in the third step the goal is to learn the remaining components of the topography model to allow for topography reconstruction and for conditional topography generation given a cell image.

5.1.1 Disentangled latent representation of synthetic cell images

In the synthetic case the number and the dimensionality of the true generative factors of synthetic cell images are known exactly: cell images are explained by four 1-dimensional generative factors f_1, f_2, f_3, f_4 and some negligible variation in colors. Given that fact, the dimensionality of the latent subspaces was chosen a priori and set to 2 for each of the subspaces $z_{f_1}, z_{f_2}, z_{f_3}, z_\varepsilon$. The motivation for such a choice is as follows. For the residual latent subspace z_ε the dimensionality of at least 2 is necessary since cell images cannot be ordered linearly by the values of the rotation angle feature f_4[1]. The problem arises from the fact that a cell image rotated by 90° differs more from the original vertically-aligned cell image (0° rotation angle), than a cell rotated by 179°, which, conversely, almost matches the original image. Therefore, the model needs at least one additional dimension to create a non-linear manifold in the latent space, where similarly rotated cell images are close. Regarding the main latent subspaces z_{f_i} the dimensionality of 1 would suffice to represent variation of 1-dimensional features, however $dim(z_{f_i}) = 2$ was chosen for visualization convenience. It is expected, however, that the model would use only a single dimension to order the centers of the inferred distribution $q_{\phi_f}(z_f|x)$, $p_{\theta_f}(z_f|f)$. Hence, $dim(z_{f_1}) = dim(z_{f_2}) = dim(z_{f_3}) = dim(z_\varepsilon) = 2$.

The hyperparameters in the first step include $\beta_\varepsilon, \beta_{f_i}, \beta_{pr,i}, \alpha_{f_i}$ $(i = 1, 2, 3)$, all of which except for β_ε pertain to the cell features of interest f_1, f_2, f_3 and their respective latent subspaces z_{f_i}. The values of these hyperparameters were selected based on the quantitative and qualitative properties of the latent subspaces $z_{f_1}, z_{f_2}, z_{f_3}$, inferred as a result of optimization of a part of the first-step objective $F_1^{pr}(x, f)$ (3.17) that relates to the cell features-related components: $q_{\phi_f}(z_f|x)$, $p_{\theta_f}(z_f|f)$, $p_{\theta_{pr}}(z_f)$, $q_{\omega_f}(f|z_f)$. The maximization objective used, separate for each cell feature, is provided below in (5.1). In fact, it is a regression objective aiming to predict f based on a sample $z_f \sim q_{\phi_f}(z_f|x)$ from the posterior distribution for a cell image x, but with a regularization forcing, firstly, the posterior to be close to the conditional prior $q_{\phi_f}(z_f|x) \approx p_{\theta_f}(z_f|f)$ and, secondly, the combined posterior to have a normal distribution $p_{\theta_{pr}}(z_f)$ with a trainable relative variance, i.e. $\sum_j (\sigma_{\theta_{pr}})_j^2 = 1$. Notably, the first term in (5.1) stands for regression performance and is

[1] This problem is referred to as "manifold mismatch" in [42]

implemented as the mean squared error (MSE) between the actual and predicted f values, hence it is minimized.

$$\begin{aligned}\alpha_f \, \mathbb{E}_{q_{\phi_f}(z_f|x)} \, log \, q_{\omega_f}(f|z_f) - \, \beta_f \, KL\left(q_{\phi_f}(z_f|x) \,||\, p_{\theta_f}(z_f|f)\right) \\ - \, \beta_{pr} \, KL\left(p_{\theta_f}(z_f|f) \,||\, p_{\theta_{pr}}(z_f)\right)\end{aligned} \tag{5.1}$$

The results for different combinations of $\beta_f, \beta_{pr}, \alpha_f$ on the validation set are provided below in Table 5.1 for a single feature - roundness (f_1). In the table KL stands for the average $KL\left(q_{\phi_f}(z_f|x) \,||\, p_{\theta_f}(z_f|f)\right)$, and KL_{full} stands for the average $KL\left(p_{\theta_f}(z_f|f) \,||\, p_{\theta_{pr}}(z_f)\right)$. The regression performance is denoted by MSE_{qzf}, while MSE_{pzf} stands for performance of an auxiliary regression, where $z_f \sim p_{\theta_f}(z_f|f)$ is sampled from the conditional prior. The latter regression is not directly optimized, however the discrepancy in performance between the main MSE_{qzf} and auxiliary MSE_{pzf} regression objectives reflects how weak the level of pressure imposed by β_f is. Furthermore, MSE_{pzf} characterizes how well the latent space is structured by the values of f, i.e. how far apart the conditional priors for different f values lie in the latent space, which is important for diversity of generated cell images for different given f values. The combinations of hyperparameters were compared after 40 epochs of training with the learning rate 0.01.

Feature	$\boldsymbol{\beta_f}$	$\boldsymbol{\beta_{pr}}$	$\boldsymbol{\alpha_f}$	Epoch	KL	KL_{full}	MSE_{qzf}	(MSE_{pzf})
f_1	1	1	10K	40	0.929	0.458	$5.569 * 10^{-5}$	$72.72 * 10^{-5}$
f_1	1	1	100K	40	1.938	0.556	$0.923 * 10^{-5}$	$59.79 * 10^{-5}$
f_1	5	1	100K	40	0.451	1.770	$2.760 * 10^{-5}$	$9.684 * 10^{-5}$
f_1	10	1	100K	40	0.218	1.953	$4.308 * 10^{-5}$	$8.324 * 10^{-5}$
f_1	20	1	100K	40	0.114	2.052	$5.713 * 10^{-5}$	$7.830 * 10^{-5}$

Table 5.1: Selection of the hyperparameters $\beta_f, \beta_{pr}, \alpha_f$ for the cell feature f_1 (roundness).

The hyperparameter β_{pr} is fixed at 1 with the rest varied, since imposing a normal "full prior" distribution $p_{\theta_{pr}}(z_f)$ on the marginal distribution $p_{\theta_f}(z_f)$ (3.5) is a side goal, mostly relevant for the topography model. It can be seen that the regression performance MSE_{qzf} improves when α_f is increased. Whereas increasing β_f improves KL and reduces the regression performance MSE_{qzf}, while also reducing the discrepancy between MSE_{qzf} and MSE_{pzf}. In Figures 5.1, 5.2 the inferred latent subspace is visualized for $\beta_f = 1$ and 10, respectively; bold dots represent centers of the respective distributions $q_{\phi_f}(z_f|x)$, $p_{\theta_f}(z_f|f)$ for a set of observations. In Figure 5.1 ($\beta_f = 1$, $\alpha_f = 100000$) a significant mismatch between the posterior $q_{\phi_f}(z_f|x)$ and the conditional prior $p_{\theta_f}(z_f|f)$ for a given observation $(x, f_1(x))$ is seen, which is reflected by high KL in Table 5.1. Whereas in Figure 5.2 ($\beta_f = 10$, $\alpha_f = 100000$), the respective distributions are relatively close. Increasing β_f further is deemed unnecessary, since it negatively influences the regression performance MSE_{qzf}. The following values of hyperparameters were selected for all three cell features f_1, f_2, f_3: $\beta_f = 10$, $\beta_{pr} = 1$, $\alpha_f = 100000$. The respective objectives for different cell features were trained until convergence with the learning rate 0.001. The results on the validation set are provided in Table 5.2. The trained components $q_{\phi_f}(z_f|x)$, $p_{\theta_f}(z_f|f)$, $p_{\theta_{pr}}(z_f)$, $q_{\omega_f}(f|z_f)$ for all three features were used as initialization of the respective components in the full cell model. Notably, the z_f latent subspaces are not correlated, as shown in Figure 5.3.

Feature	$\boldsymbol{\beta_f}$	$\boldsymbol{\beta_{pr}}$	$\boldsymbol{\alpha_f}$	Epoch	KL	KL_{full}	MSE_{qzf}	(MSE_{pzf})
f_1	10	1	100K	77	0.195	2.020	$3.995 * 10^{-5}$	$7.540 * 10^{-5}$
f_2	10	1	100K	56	0.182	2.048	$3.754 * 10^{-5}$	$7.177 * 10^{-5}$
f_3	10	1	100K	67	0.191	2.018	$3.891 * 10^{-5}$	$7.391 * 10^{-5}$

Table 5.2: Selected $\beta_f, \beta_{pr}, \alpha_f$ for f_1, f_2, f_3.

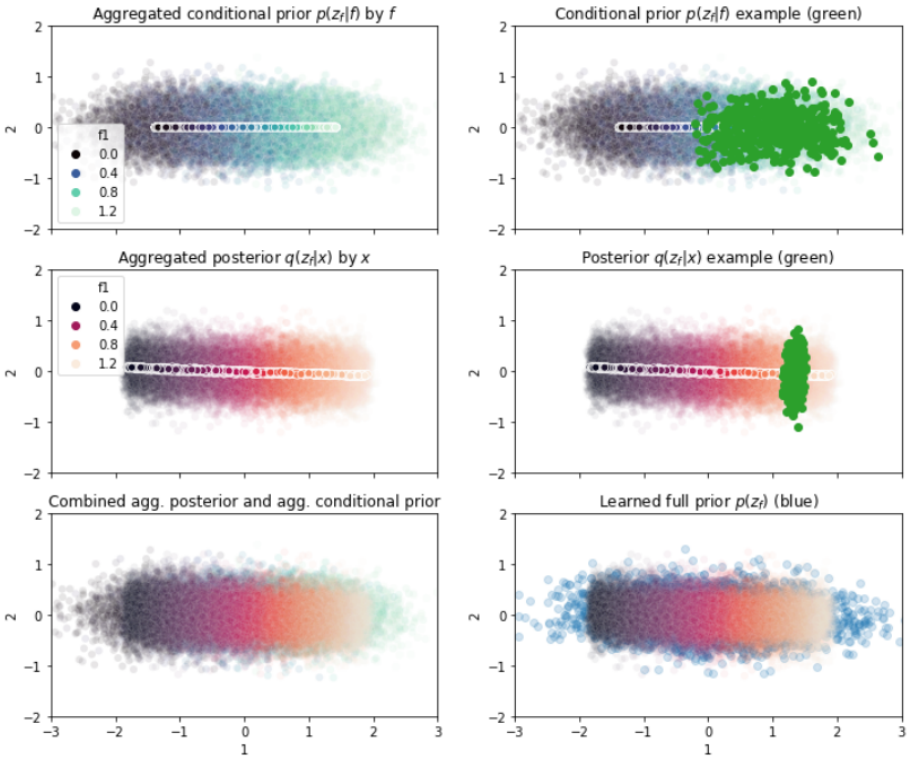

Figure 5.1: Visualization of the z_{f_1} latent subspace ($\beta_f = 1$, $\beta_{pr} = 1$, $\alpha_f = 100000$).

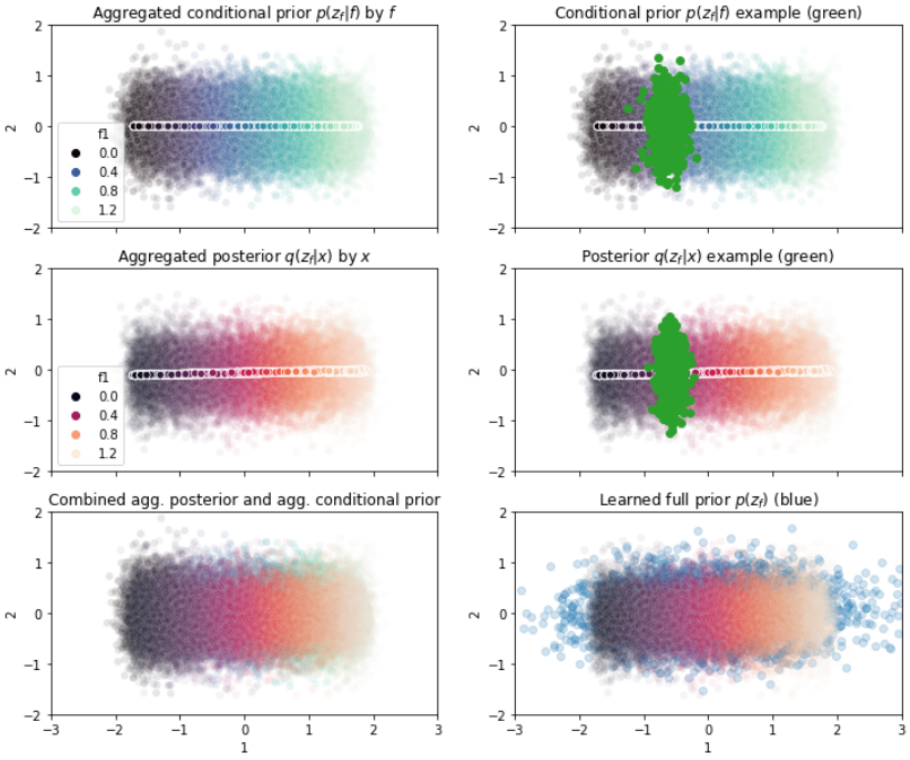

Figure 5.2: Visualization of the z_{f_1} latent subspace ($\beta_f = 10$, $\beta_{pr} = 1$, $\alpha_f = 100000$)

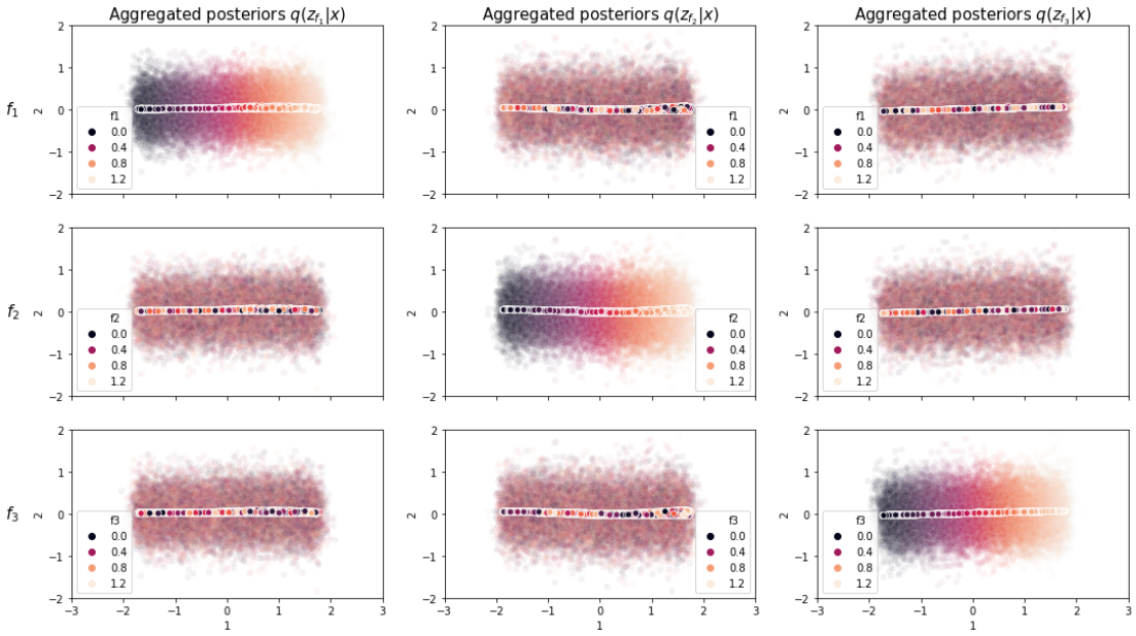

Figure 5.3: Visualization of aggregated posteriors z_{f_i} colored by the respective cell features.

A challenging part in training of the cell model is to select a suitable capacity of the residual latent subspace z_ε. The challenge originates from the fact that all three cell features-linked latent variables z_f are heavily constrained by a regression objective with a high α_{f_i} and a heavy KL-regularization with a high β_{f_i}. The idea of these constraints is that the model is allowed to use z_f variables to only encode the variation in cell images explained by the features f_i. On the contrary, the residual latent variable z_ε, the capacity of which is controlled by the single coefficient β_ε, is not constrained by an auxiliary regression objective and can potentially be used by the model to encode any factors of variation in the data. Therefore, the challenge is to restrict the capacity of z_ε sufficiently, such that the model only uses it to encode the residual factors of variation, i.e. the rotation angle f_4 in the case of the synthetic dataset. If β_ε is too low, the model would use z_ε to encode as much variation as possible, thus ignoring some or all of the z_f latent variables. This situation is undesired even if the model achieves a better reconstruction quality, since the key idea is to derive a disentangled latent representation of cell images. If β_ε is too high, the model would not be able to encode the rotation angle factor. Notably, the dimensionality of z_ε is another leverage to control the capacity of the residual subspace, however, the smallest reasonable latent space dimensionality of 2 was already chosen for z_ε based on prior knowledge, as explained above.

Another complication is that β_ε should be chosen for given values of $\beta_{f_i}, \beta_{pr,i}, \alpha_{f_i}$ $(i = 1, 2, 3)$ for all cell features. If β_ε is too high with respect to the chosen values of z_f-related hyperparameters for some feature f, then the auxiliary regression objective would become a too weak constraint on that z_f subspace, and the model would use z_f to encode the residual factors apart from f. Simultaneous adjustment of all cell-model hyperparameters could be tedious with multiple cell features. To simplify the training process, the z_f-related components trained during optimization of the objective (5.1), as described above, were fixed during training of the residual components. Hence, the cell model objective (3.17) was trained with respect to the weights of $p_\theta(x|z_\varepsilon, z_f)$, $q_{\phi_\varepsilon}(z_\varepsilon|x)$, while the weights of $q_{\phi_f^*}(z_f|x)$, $p_{\theta_f^*}(z_f|f)$, $q_{\omega_f^*}(f|z_f)$, $p_{\theta_{pr}^*}(z_f)$ were fixed.

To fine-tune β_ε, the following factors were taken into account: the value of KL_ε, which denotes the average $KL\left(q_{\phi_\varepsilon}(z_\varepsilon|x)\,||\,p(z_\varepsilon)\right)$, the shape of the aggregated posterior distribution $q_{\phi_\varepsilon^*}(z_\varepsilon|x)$ and the quality of cell image reconstruction, including the reconstruction error numerical score. Furthermore, the fact of whether the space z_ε was structured by the cell features f_1, f_2, f_3 was considered: by design the space z_ε should be structured only according to the residual rotation

angle f_4 feature; otherwise, the capacity of z_ε might be excessive. All models were trained based on a single pre-trained baseline model[2] for 15 epochs with the learning rate 0.001. The quantitative results on the validation set are presented below in Table 5.3.

$\boldsymbol{\beta_\varepsilon}$	KL_ε	Reconstruction error
100	7.566	115302.2
200	5.674	114926.9
300	4.827	114334.1
390	4.335	114345.5
400	1.156	133709.5
500	0.779	133906.7
1000	0.005	134594.6

Table 5.3: Selection of the hyperparameter β_ε.

It was found that the drop in reconstruction error at $\beta_\varepsilon < 400$ is coupled with reduction in the quality of disentanglement and with the space z_ε becoming less structured by the rotation angle f_4 feature. At the same time, the values of $\beta_\varepsilon > 500$ compress the space z_ε excessively, which results both in worse visual reconstruction quality and worse disentanglement. The inferred latent subspaces for $\beta_\varepsilon \in \{300, 400, 500\}$ are visualized in Figures 5.4, 5.5, 5.6, colored by $sin(f_4)$; bold dots represent centers of the posterior distributions for a set of cell images. Hence, $\beta_\varepsilon = 400$ was selected as the default value.

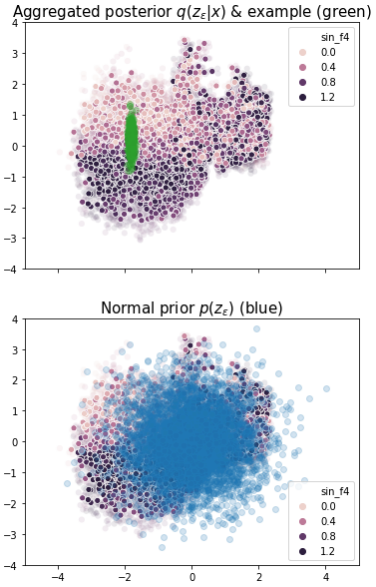

Figure 5.4: z_ε space: $\beta_\varepsilon = 300$

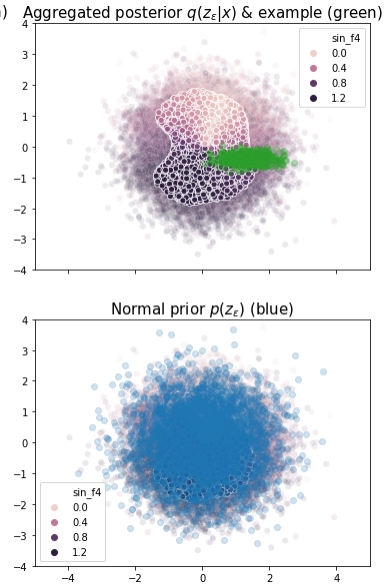

Figure 5.5: z_ε space: $\beta_\varepsilon = 400$

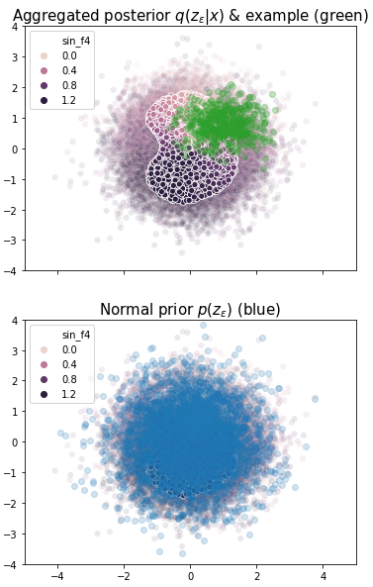

Figure 5.6: z_ε space: $\beta_\varepsilon = 500$

Subsequently, the model was trained until convergence involving a few techniques, such as training alternately the decoder $p_\theta(x|z_\varepsilon, z_f)$ except for the first fully connected layer (with the encoder's weights fixed) and then the encoder $q_{\phi_\varepsilon}(z_\varepsilon|x)$ with the first fully connected layer of the decoder (with the rest decoder's weights fixed). During training of the encoder separately, an alternative method to control the capacity of the latent subspace z_ε was used (2.7) with $\gamma = 300, C = 2$. The resulting subspace z_ε is visualized in Figure 5.7. It can be seen that the subspace is primarily structured according to rotation angle f_4, however the points are also ordered according to elongation f_2, which could imply that the latent representation is not perfectly disentangled.

[2]The baseline model was trained from initialization of the model for approximately 20 epochs with the learning rate decreased from 0.01 to 0.001 after first 3 epochs. Different values of β_ε were used, starting from $\beta_\varepsilon = 1$.

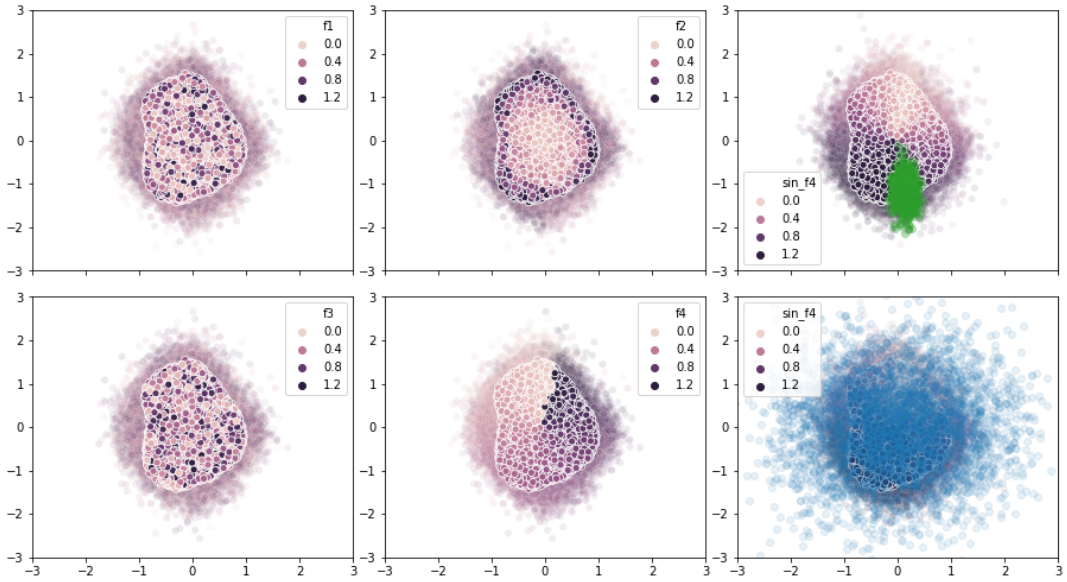

Figure 5.7: Visualization of the z_ε space: aggregated posterior colored by $f_1, f_2, f_3, f_4, sin(f_4)$; posterior $q_{\phi_\varepsilon}(z_\varepsilon|x)$ example in green; prior $p(z_\varepsilon)$ in blue.

A qualitative latent space traversal approach is used to evaluate the level of disentanglement in the latent space. In Figure 5.8 the z_f subspaces corresponding to cell features are traversed using the trained conditional priors $p_{\theta_f}(z_f|f)$. Each (scaled) feature is changed from 0 to 1, and for each given value of a cell feature f a sample $z_f \sim p_{\theta_f}(z_f|f)$ is passed to the decoder to generate a cell image, while the rest z_f samples are fixed; leftmost images correspond to zero values of f_i. It can be seen that the features elongation f_2 and nucleus size f_3 are well captured and disentangled by the model in the z_f subspaces, since only the respective properties of the cell image change. Whereas the space z_{f_1}, corresponding to roundness, is moderately correlated with elongation, i.e. not fully disentangled.

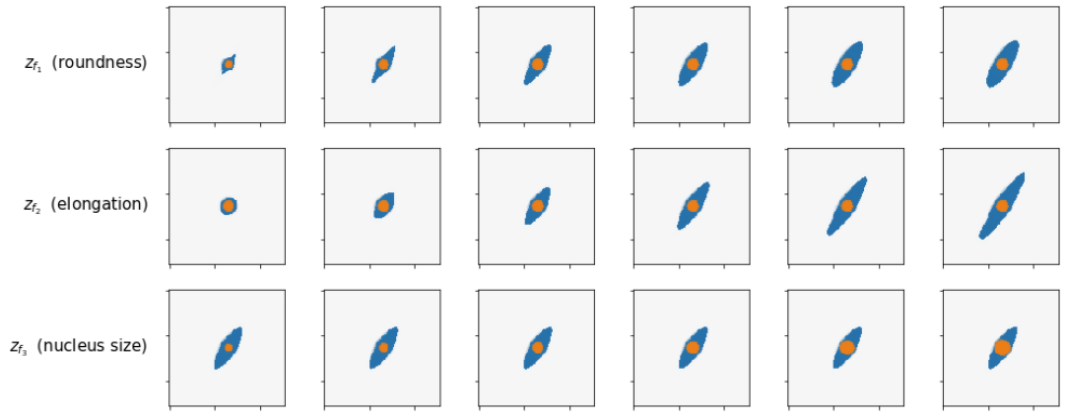

Figure 5.8: Latent space traversal: $z_{f_1}, z_{f_2}, z_{f_3}$.

To evaluate whether z_ε was able to encode the residual variation in cell images, i.e. the rotation angle f_4 feature, a sampling approach was used. A given cell image is firstly reconstructed; then, different samples $z_\varepsilon \sim p(z_\varepsilon)$ are taken while the z_f variables are fixed at the means of the posterior distributions $q_{\phi_f}(z_f|x)$. In Figure 5.9 it is shown that sampling from prior results in rotation of approximately the same generated cell image. Therefore, it can be concluded that z_ε captures the rotation angle f_4 feature.

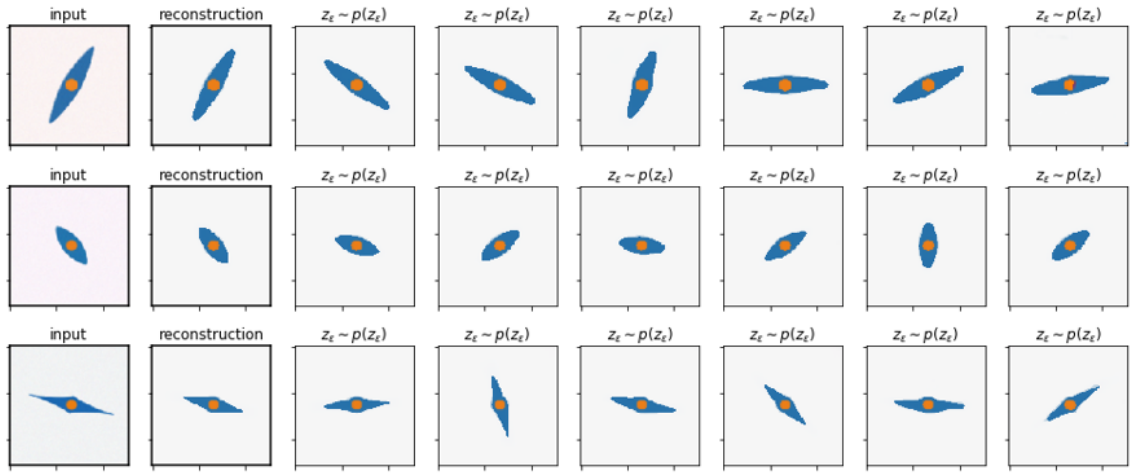

Figure 5.9: Sampling from $p(z_\varepsilon)$.

Overall, the model is able to reconstruct given cell images using a disentangled latent space, as shown in Figure 5.10. However, it was noted that the model has difficulty learning and reproducing the concept of roundness f_1, especially with regard to low-elongation cells.

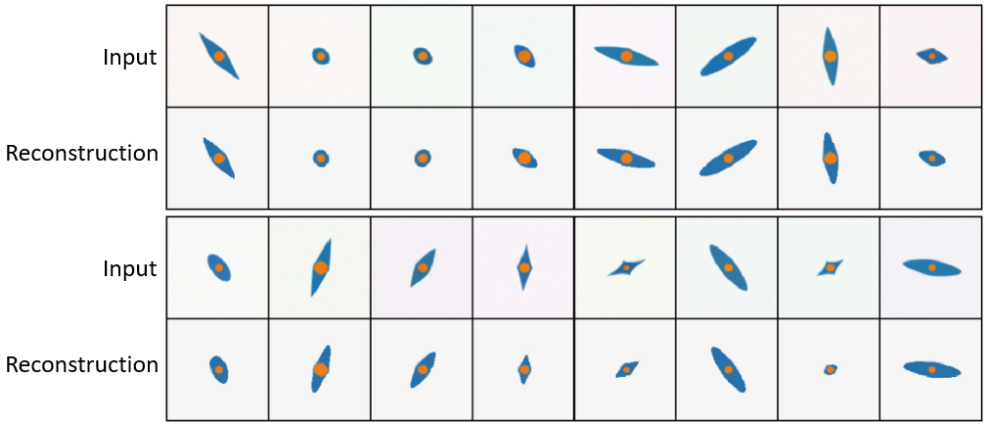

Figure 5.10: Cell image reconstruction.

5.1.2 Modeling the influence of topographies on cells for experiment simulation

In the second step of training (3.19) the goal is to model the possible influence of topographies on cells by training the topography-model encoders for each z_f subspace. The only hyperparameters at this step are $\beta_{pf,i}$ $(i = 1, 2, 3)$, which control the penalty on the posteriors $q_{\varphi_f}(z_f|p)$ for diverging from the full prior distributions $p_{\theta^*_{pr}}(z_f)$, learned in the previous step. The artificial relationship between the topography dataset and the cell dataset, introduced in Chapter 4, implies that radius of topographies g_2 is positively correlated with elongation of cells f_2. Therefore, the hyperparameters β_{pf} are selected in such a way that the posteriors in the subspaces z_{f_1}, z_{f_3} approximately match the respective full priors $p_{\theta^*_{pr,1}}(z_{f_1}), p_{\theta^*_{pr,3}}(z_{f_3})$, while in the subspace z_{f_2}, corresponding to cell elongation, the posteriors are spread according to the radius g_2 of the input topography. To fine-tune the hyperparameters, several models with different β_{pf} values were trained for 6-11 epochs, depending on the validation loss value decline, with the learning rate 0.01. Notably, the same value of β_{pf} for all three z_f subspaces is taken in each model, since in the

general case the relationship between topographies and cells in unknown. The results are provided in Table 5.4, where KL_i stands for the average $KL\left(q_{\varphi_{f_i}}(z_{f_i}|p)\,||\,p_{\theta^*_{pr,i}}(z_{f_i})\right)$.

β_{pf}	Relationship	Epoch	KL_1	KL_2	KL_3	Reconstruction error (cells)
100	$g_2 \rightarrow f_2$	11	0.652	1.480	0.539	134685.5
200	$g_2 \rightarrow f_2$	7	0.323	1.057	0.088	134864.0
300	$g_2 \rightarrow f_2$	8	0.204	0.843	0.020	134970.0
400	$g_2 \rightarrow f_2$	8	0.140	0.695	0.008	135032.0
500	$g_2 \rightarrow f_2$	8	0.109	0.589	0.006	135095.6
1000	$g_2 \rightarrow f_2$	10	0.055	0.304	0.005	135346.0
1000	control case	6	0.005	0.006	0.005	136226.0

Table 5.4: Selection of the hyperparameters $\beta_{pf,i}$.

It can be seen from Table 5.4 that an increase in β_{pf} leads to lower KL_i values in the dataset with artificial relationship, which is expected, since the penalty on posteriors in all three z_f spaces is increased. Furthermore, the value of KL_2, corresponding to the cell elongation subspace z_{f_2}, is significantly larger than KL_1, KL_2 in all cases except for the control case with no relationship. This observation implies that the model is able to discern between topographies on the basis of what cell elongation value they induce. Notably, KL_1 is significantly larger than KL_3 in all cases with $g_2 \rightarrow f_2$ relationship embedded in the dataset, however there was no intended relationship between g_2 and f_1. This observation is attributed to the fact that the roundness latent variable z_{f_1} was not perfectly disentangled, as reported above, and captured elongation-related variance as well. Furthermore, it can be seen that in the control case with random training pairs the model is, as expected, unable to maximize the likelihood of the cell images, which is reflected by a larger reconstruction error as compared with the main case with the same $\beta_{pf} = 1000$ value, and by equally small values of KL_i.

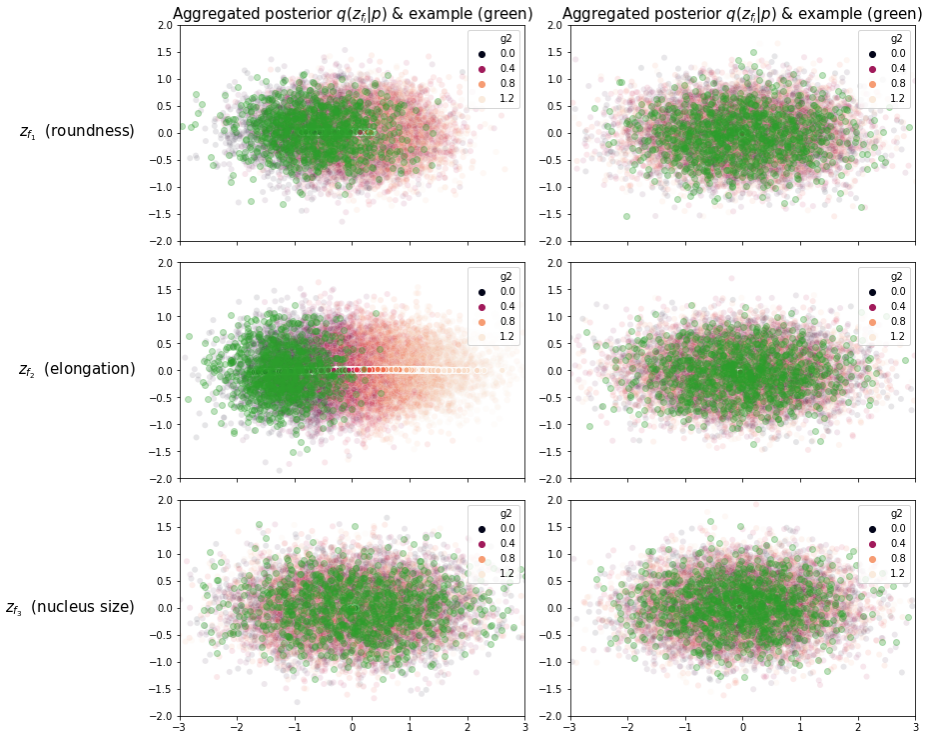

Figure 5.11: Visualization of aggregated posteriors in the topography-model z_{f_i} subspaces, colored by topography radius g_2. Relationship case (left); Control case (right).

The value of $\beta_{pf} = 400$ was chosen as the default value based on the fact that the KL_3 value is close to that of the control case, while KL_2 is large enough to spread the topography-model posteriors in the z_{f_2} space to express the induced variation of cell elongation. The resulting z_f subspaces are visualized in Figure 5.11 for the main and control cases. Posteriors diverging from the full prior are also seen in the z_{f_1} subspace, which is undesired, but is caused by the correlation between z_{f_1} and z_{f_2}.

In silico experiment

Once the topography-model encoders $q_{\varphi_f^*}(z_f|p)$, corresponding to cell features-related subspaces z_f, are trained, the model can be used to simulate the experiment. In the proposed framework, simulation of the experiment implies conditional generation of cell images given a topography image, as described in the Application scenarios 3.5 section. In Figure 5.12 it is shown that a high-radius input topography results in different cell images sharing in common only the property of high elongation. Similarly, a single low-radius input topography leads to a variety of low-elongation cells, which are at the same different in other properties, such as the nucleus size.

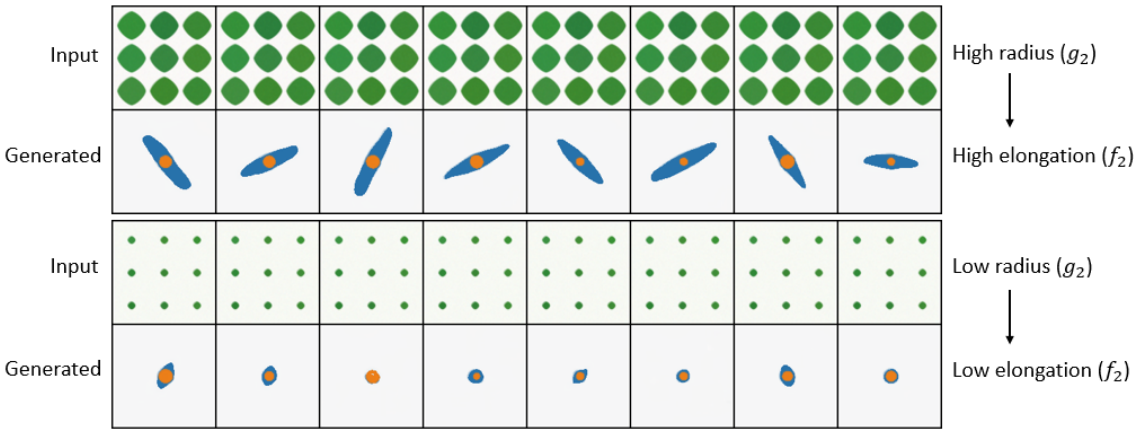

Figure 5.12: Simulation of the experiment: generating cell images conditioned on a topography image.

5.1.3 Modeling the inverse mapping for cell-conditioned topography design

In the last step of training (3.20) the goal is to train the remaining components: the encoder $q_{\varphi_l}(l_\varepsilon|p)$ for residual variation in topographies, not related to their influence on cells, and the topography-model decoder $p_\vartheta(p|l_\varepsilon, z_f)$. The dimensionality of the subspace l_ε was chosen as 2 based on the fact that exactly two true generative factors define a topography image: roundness g_1 and radius g_2. It is desired, however, that the decoder will use l_ε only to extract information about the residual variation in topographies, i.e. roundness g_1, since radius g_2 is assumed to influence cell elongation f_2 and is already captured by the z_{f_2} latent variable, as described in the previous section. The hyperparameters of the last step are η and β_l, where β_l controls the capacity of the l_ε latent subspace, and η controls the balance between the objectives of topography reconstruction and cell-conditioned topography generation. Specifically, η determines the relative importance of the latter objective. In the case when $\eta = 0$, the model only maximizes the reconstruction objective, similar to a VAE. It would likely ignore the z_{f_i} latent variables and would try to encode all the variation in topography images using the l_ε subspace. Alternatively, when η is too large, the model would only focus on expressing the cell-topography relationship, while sacrificing the reconstruction quality.

Balancing the two objectives simultaneously with fine-tuning the capacity of the l_ε space was found particularly challenging and, therefore, a two-phase training strategy was used. During the **first phase** of training η was set to 0, thereby allowing the model to only pursue the reconstruction quality and to use l_ε to encode both factors of variation in the topography image dataset. The capacity of the l_ε subspace was fine-tuned during this phase as well. Noteworthy, instead of adjusting β_l an alternative method (2.7) to control the capacity of a latent space was used, where the parameter γ controls the penalty imposed on the posterior $q_{\varphi_l}(l_\varepsilon|p)$ for diverging from the prior $p(l_\varepsilon)$ more than by a margin C; hence, C represents the allowed capacity. In the **second phase**, the weights of the encoder $q_{\varphi_l^*}(l_\varepsilon|p)$ were fixed and η was increased to stimulate the decoder to use information captured by the z_f latent variables during topography generation. Specifically, the idea is that the decoder should learn to extract information about topography radius g_2 from the z_{f_2} subspace.

First phase of training

Similarly to the training procedure of the cell model, the space capacity-related hyperparameters γ, C were chosen based on the following factors: the quality of topography image reconstruction, the average value of $KL\left(q_{\varphi_l}(l_\varepsilon|p)\,||\,p(l_\varepsilon)\right)$ on the validation set and the shape of the aggregated posterior. The latter factor is important for conditional topography generation, since l_ε is sampled from the prior $p(l_\varepsilon)$ in this application scenario, and it is desirable to generate diverse topographies. The resulting subspace l_ε is visualized in Figure 5.13, where the chosen values are $\gamma = 200$, $C = 3$; bold dots represent centers of the posterior distributions $q_{\varphi_l}(l_\varepsilon|p)$ for a set of observations. The corresponding model was trained for 48 epochs with the learning rate 0.001. It can be seen that the space is structured by both roundness g_1 and radius g_2 features.

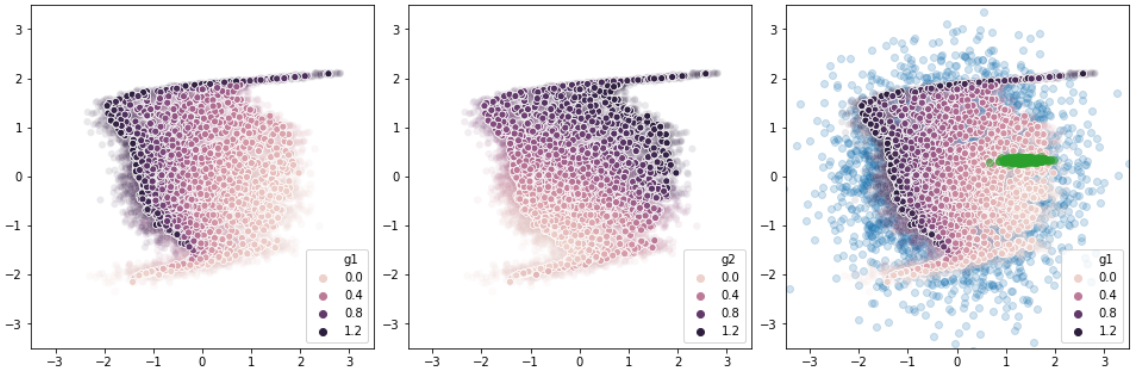

Figure 5.13: Visualization of the l_ε space: aggregated posterior colored by g_1, g_2; posterior $q_{\varphi_l}(l_\varepsilon|p)$ example in green; prior $p(l_\varepsilon)$ in blue.

Figure 5.14 demonstrates the achieved topography reconstruction quality, which can be characterized as decent. At the same time, since $\eta = 0$ in the first phase, the model is yet unable to perform cell-conditioned topography generation, which is shown in Figure 5.15: given a high-elongation cell image, the model generates topographies with arbitrary radius.

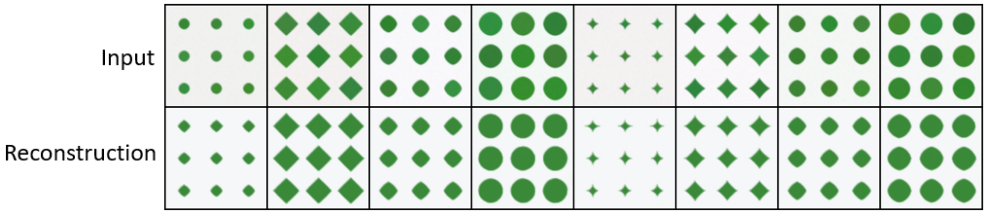

Figure 5.14: Topography image reconstruction (after the first phase of training).

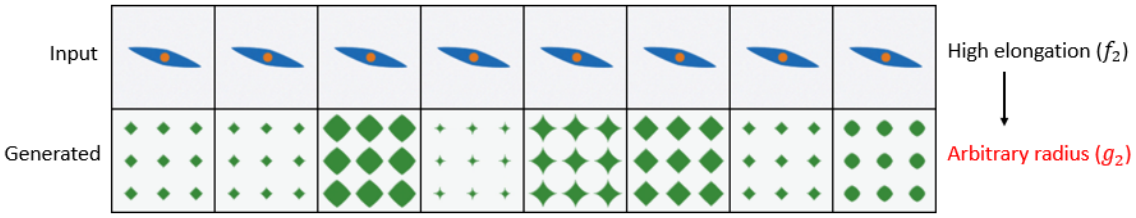

Figure 5.15: Cell-conditioned topography design (after the first phase of training).

Second phase of training

In the second phase the weights of the encoder were fixed $q_{\varphi_l^*}(l_\varepsilon|p)$, and different values of η were tested. In general, when $\eta > 0$, the topography reconstruction error increases, while the cell-conditioned topography generation objective decreases. The goal is therefore to find η that allows the model to express the existing relationship between topographies and cells, without sacrificing much of the reconstruction quality. The model was trained from the 49th to the 99th epoch with the chosen value $\eta = 0.05$ and the learning rate 0.001. In Figure 5.16 the results for the task of topography image reconstruction are shown, where one can see that the quality of reconstruction moderately decreased.

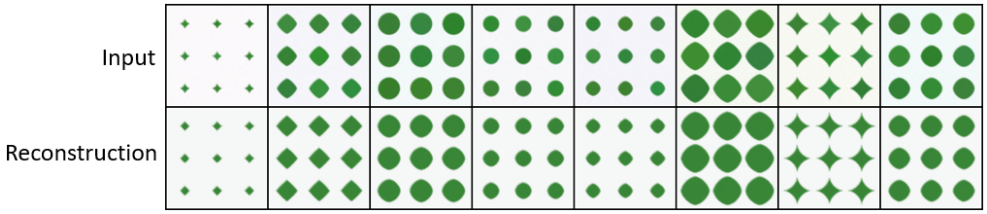

Figure 5.16: Topography image reconstruction (after the second phase of training).

Figure 5.17, on the other hand, demonstrates that the model learned the cell-topography artificial relationship, embedded in the dataset: a high-elongation input cell image results in high-radius topographies and vice versa. An evident problem, however, is that of poor diversity in the generated topography samples. The generated shapes possess the needed radius, however, they poorly resemble the topographies from the dataset with respect to roundness. To address this problem, it is proposed to augment the training process with a GAN objective, which is discussed below.

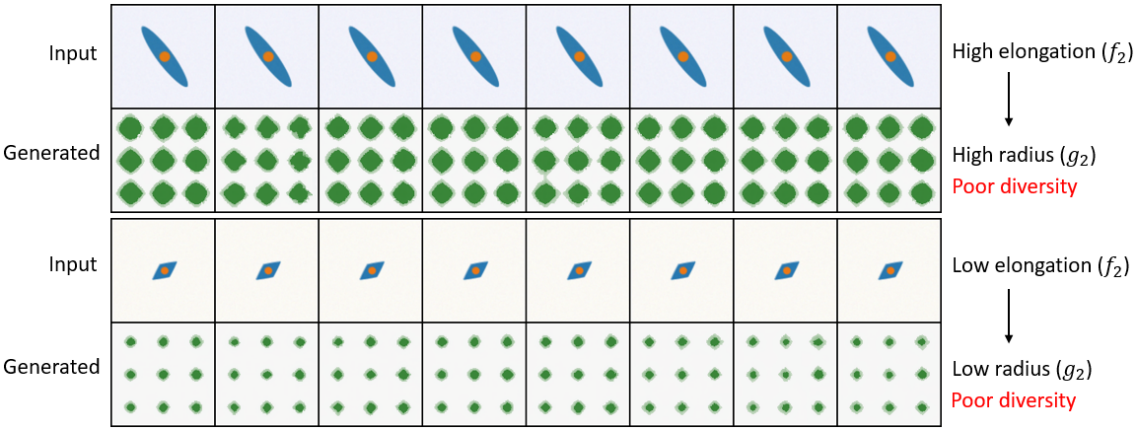

Figure 5.17: Cell-conditioned topography design (after the second phase of training).

Adding a GAN objective

The problem of poor diversity in generated cell-conditioned topography samples arises, arguably, from the fact that in the original last-step training objective (3.20) the second term aims to minimize the distance between a true topography from a (topography, cell) training pair and a topography generated as a result of sampling $z_f \sim q_{\phi_f^*}(z_f|x)$ from the posterior and $l_\varepsilon \sim p(l_\varepsilon)$ from the prior. This implies that, for a given cell image, the second term in the objective stimulates the model to generate a topography as close to a particular true topography as possible, but using only the information encoded in the z_f subspaces. In other words, given a high-elongation cell, the model is forced to generate a particular high-radius topography using no information about its roundness. Such setting encourages the model to generate topography images with a correct radius g_1, but averaged by the residual roundness g_2 property in order to achieve low reconstruction error on average. However, the original intention behind the objective (3.20) was in fact to stimulate the model to generate *any realistic* topography, i.e. with arbitrary but concrete roundness, and with a radius corresponding to the elongation of the input cell: a topography with a correct radius but with arbitrary roundness should be deemed equally good in the ideal case. Hence, the similarity metric used has failed to express this intention.

In [33] the authors discuss the limitations of pixel-wise similarity metrics for images and propose the VAE/GAN approach, where a VAE architecture is enriched with a discriminator model, which learns a measure of similarity between images instead of formally defining it. The proposed approach is exploited in the present work: an auxiliary discriminator network is introduced, which aims to discern between true and generated topographies, by maximizing the objective (5.2). The discriminator outputs $D(p)$, which is the probability that a given topography p originates from the dataset P, i.e. is not generated.

$$F_D(x,p) = \mathbb{E}_{p\sim P}\ log\, D(p)\ -\ \mathbb{E}_{p(l_\varepsilon)q_{\phi_f^*}(z_f|x)}\ log\left(1 - D(p_G)\right) \quad p_G \sim p_\vartheta(p|l_\varepsilon, z_f) \tag{5.2}$$

At the same time, the base model, with all components frozen except for the decoder $p_\vartheta(p|l_\varepsilon, z_f)$, is treated as a generator. Apart from the main objective, it aims to mislead the discriminator and to generate topography samples that resemble true topographies. Notably, only the topography samples generated in the task of cell-conditioned topography design are used in the auxiliary GAN-objective term, while for the task of topography reconstruction the original likelihood term is used. The maximization objective is formulated below (5.3).

$$\begin{aligned} F_{VAE/GAN}(x,p) = \mathbb{E}_{q_{\varphi_l^*}(l_\varepsilon|p)q_{\varphi_f^*}(z_f|p)}\ log\, p_\vartheta(p|l_\varepsilon, z_f)\ +\ \eta\ \mathbb{E}_{p(l_\varepsilon)q_{\phi_f^*}(z_f|x)}\ log\, p_\vartheta(p|l_\varepsilon, z_f) \\ +\ \xi\ \mathbb{E}_{p(l_\varepsilon)q_{\phi_f^*}(z_f|x)}\ log\, D(p_G) \quad p_G \sim p_\vartheta(p|l_\varepsilon, z_f) \end{aligned} \tag{5.3}$$

Joint optimization of both objectives was performed alternately, such that on a given batch the weights of the discriminator were first updated, and subsequently the weights of the generator (decoder $p_\vartheta(p|l_\varepsilon, z_f)$) were updated. Furthermore, it was found that the discriminator model was learning relatively faster, which hindered the training process of the main model. Therefore, the weights of the discriminator were updated every 10 batches, while the weights of the generator were updated on each batch. The models were trained for 26 epochs with the learning rate 0.001 and η increased to 2; ξ, which controls the importance of the auxiliary objective, was set to 10^6.

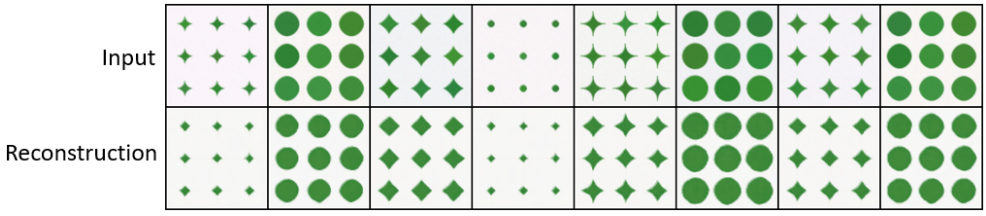

Figure 5.18: Topography image reconstruction (adding a GAN objective).

As a result, the model learned to generate diverse topography images in the task of cell-conditioned topography design with the radius feature g_2 corresponding to elongation of the respective cell, as shown in Figure 5.19. The quality of topography reconstruction decreased further, which was expected due to introduction of an auxiliary objective; the results are shown in Figure 5.18.

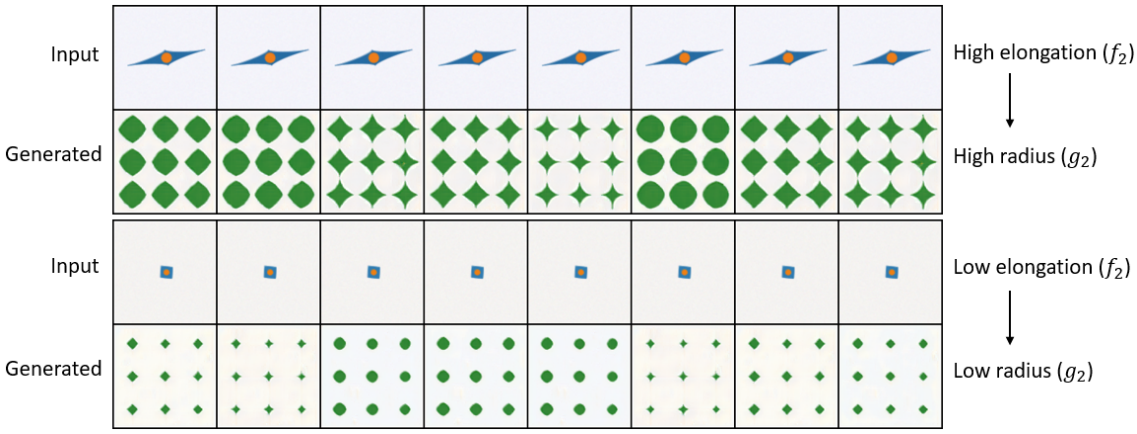

Figure 5.19: Cell-conditioned topography design (adding a GAN objective).

Finally, the model is tested in the third application scenario: *in silico* topography design based on a cell feature value. Figure 5.20 shows different generated topographies for a series of increasing values of cell elongation f_2, taken as input to the model. It can be seen that the radius g_2 of the generated topographies increases, while the residual feature, roundness g_1, varies.

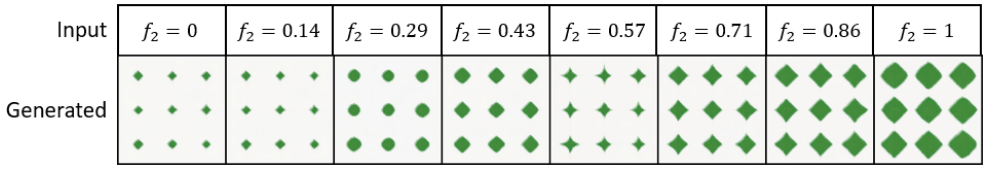

Figure 5.20: Feature value-conditioned topography design (based on cell elongation f_2 value).

5.2 Experiments on the ALP screening dataset

In the first step, the goal is to derive a disentangled latent representation of cell images, where only cell area f is taken as the cell feature of interest. Therefore, a single z_f subspace should correspond to cell area, while z_ε should encode all the residual variation in cell images and, ideally, should be invariant to cell area. In the second step, a single topography-model encoder $q_{\varphi_f}(z_f|p)$ is trained with the goal to identify the influence of topographies on cells, and in the last step, the inverse cell-topography mapping is learned to allow for cell-conditioned topography design.

5.2.1 Disentangled latent representation of real cell images

The dimensionality of the latent subspace z_f is selected as 2, as in the synthetic case. Regarding the residual space z_ε, since the number of true generative factors of real cell images is unknown, the dimensionality $dim(z_\varepsilon)$ is chosen experimentally. Notably, images of real cells seem to have arbitrary shapes and occupy any parts of a square (Figure 4.10, 4.11) while being centered initially. Therefore, it is hypothesized that the dimensionality of a latent manifold representing cell images

should be of the same order as the number of pixels: $64 * 64 = 4096$. The following values were tested: $dim(z_\varepsilon) \in \{1024, 4096\}$. To chose the dimensionality of z_ε, plain VAE models were trained for approximately 500 epochs with the learning rate decreasing from 0.01 to 0.0001; β_ε was set to 0.0001. The achieved quality of cell image reconstruction is illustrated in Figure 5.21, which shows that more dimensions allow for a more detailed reconstruction.

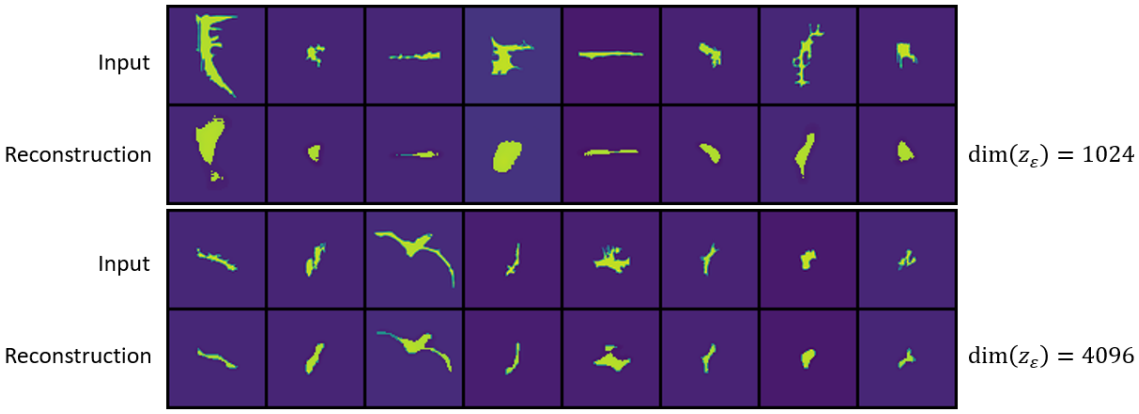

Figure 5.21: VAE: cell image reconstruction, $dim(z_\varepsilon) \in \{1024, 4096\}$.

The hyperparameters in the first step include $\beta_\varepsilon, \beta_f, \beta_{pr}, \alpha_f$, among which the last three pertain to the latent variable z_f. Their values were taken as in the synthetic case: $\beta_f = 10$, $\beta_{pr} = 1$, $\alpha_f = 100000$. As opposed to the synthetic case, however, all the components of the cell model were trained in a single objective (3.17), as described in 3.3.1. To fine-tune β_ε, the following strategy was used. At first β_ε is set to a high value ($\beta_\varepsilon = 10$), which severely limits the capacity of the residual latent subspace and stimulates the model to use the latent variable z_f to encode cell area-related information. Gradually, β_ε is reduced to relax pressure on z_ε, thereby allowing the model to capture more factors of variation in the data. However, if β_ε is too low, the capacity of residual latent space is large enough for the model to ignore z_f and use only z_ε. It was found that for $\beta_\varepsilon < 0.2$ the model starts to ignore z_f, and visual quality of disentanglement decreases. Therefore, $\beta_\varepsilon = 0.2$ is taken in the final model. To facilitate training, the decoder of the model (except for the first fully connected layer) was initialized from the respective VAE with $dim(z_\varepsilon) = 4096$. The model was trained for 322 epochs with the learning rate decreasing from 0.001 to 0.0001. The results on the validation set are provided in Table 5.5.

Epoch	$\boldsymbol{\beta_\varepsilon}$	KL_ε	Reconstruction error
11	10	94.7	7826.4
19	5	105.0	6650.2
41	2	95.6	5498.2
160	1	96.9	4745.6
184	0.5	107.1	4715.6
249	0.5	120.3	4679.5
262	0.2	168.0	4657.9
322	0.2	178.7	4635.0

Table 5.5: Fine-tuning β_ε.

The resulting z_f space is visualized in Figure 5.22. It can be seen that the learned full prior excludes a region of the aggregated posterior corresponding to high cell area. That is explained by the fact that the distribution of the cell area feature f is highly skewed: Figure 5.23. The residual latent subspace z_ε is visualized in Figure 5.24 using a 2D Principal component analysis (PCA) projection; bold dots represent centers of the posterior distributions for a set of cell images.

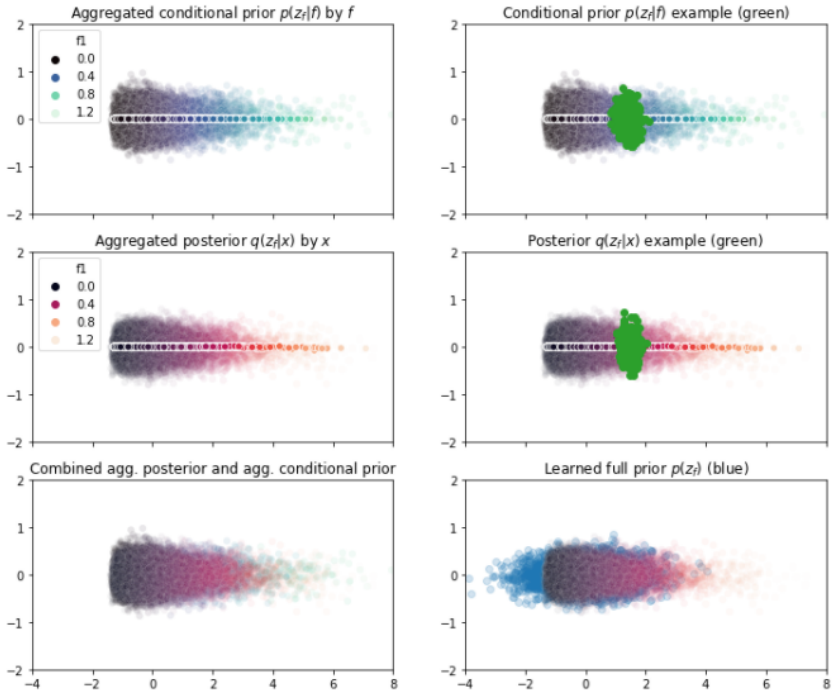

Figure 5.22: Visualization of the z_f latent subspace ($\beta_f = 1$, $\beta_{pr} = 1$, $\alpha_f = 100000$).

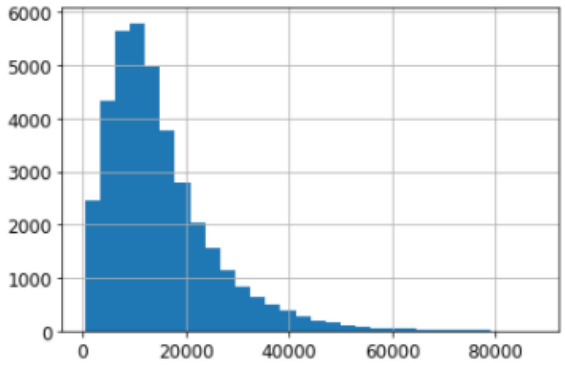

Figure 5.23: Histogram of the cell area distribution. Values $>$ 88000 (99.9% quantile) are excluded.

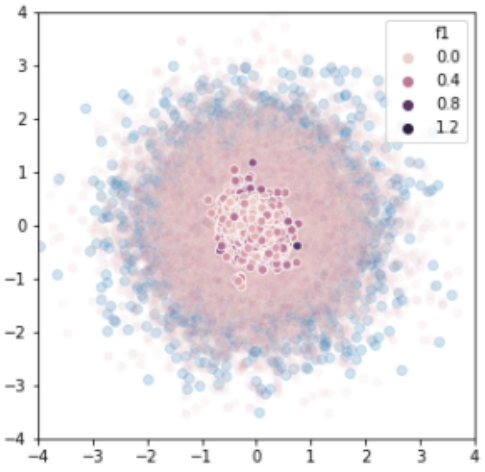

Figure 5.24: Visualization of the z_ε space: aggregated posterior colored by cell area value; prior $p(z_\varepsilon)$ in blue.

To evaluate disentanglement in the latent space, a qualitative latent space traversal approach is used. The (scaled) feature f corresponding to cell area is changed from 0 to 1 and is passed to the conditional prior $p_{\theta_f}(z_f|f)$, while z_ε is kept fixed at the mean of the posterior distribution $q_{\phi_f}(z_f|x)$ for a given cell image. Figure 5.25 shows that traversing the z_f space yields a change in the area of generated cells, while their shape is mostly preserved. To evaluate whether z_ε captures the residual variation in cell images except for area and whether it is invariant to cell area, different samples from the prior distribution $p(z_\varepsilon)$ are taken, while z_f is kept fixed at the mean of the posterior distribution for a given cell image. Figure 5.26 shows that as a result of this procedure, the model generates a variety of cell images with area similar to the input image.

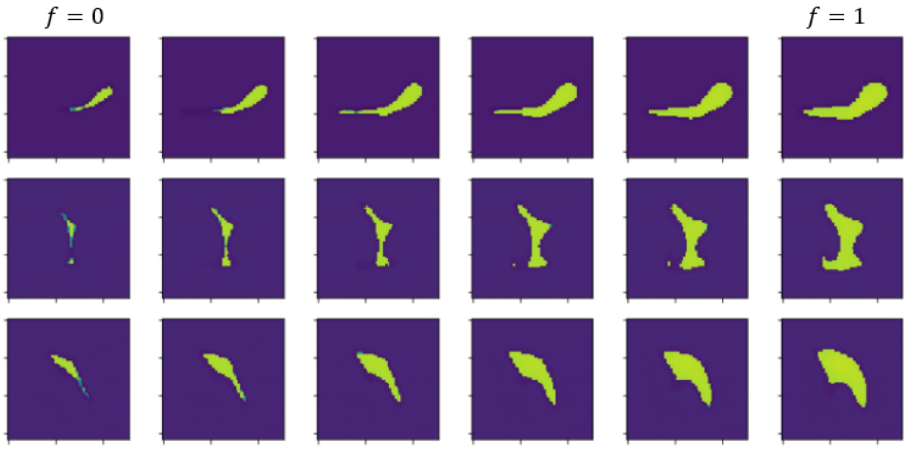

Figure 5.25: Latent space traversal: z_f (cell area).

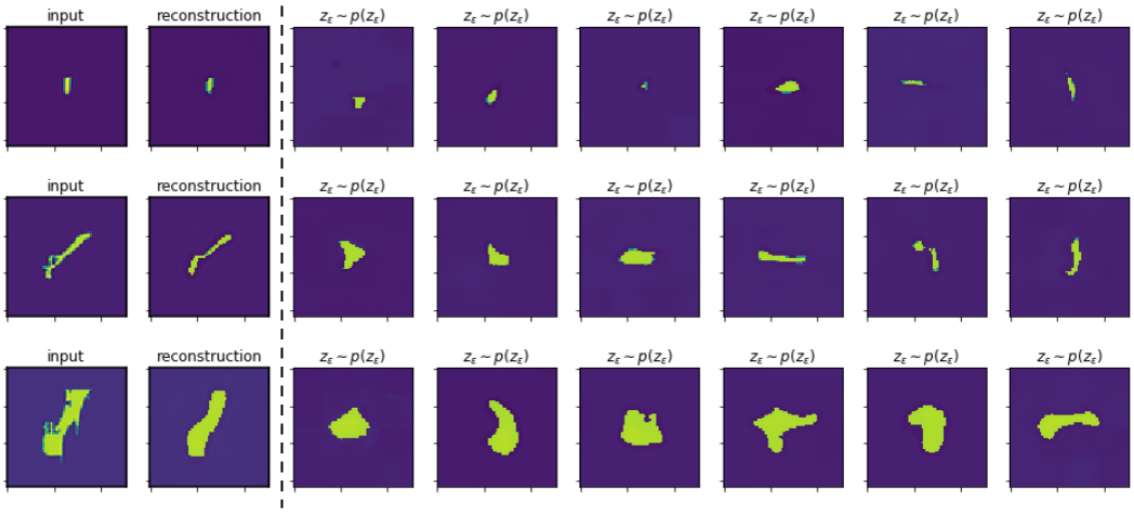

Figure 5.26: Sampling from $p(z_\varepsilon)$.

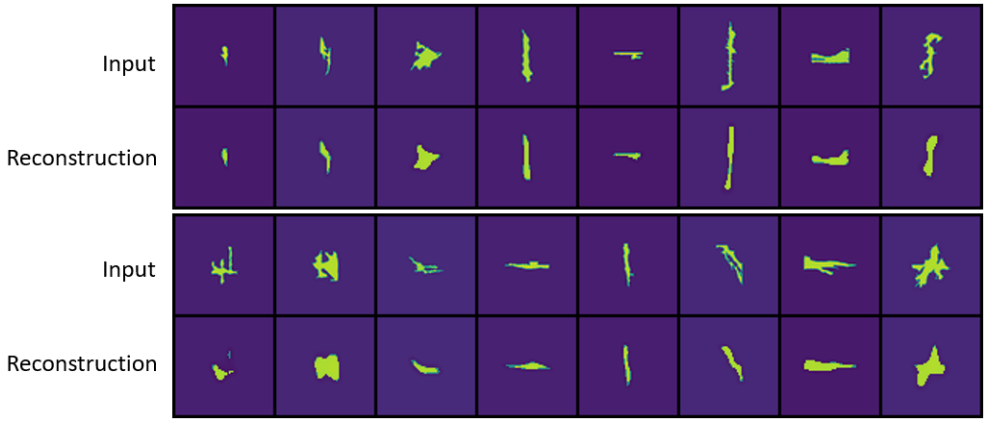

Figure 5.27: Cell image reconstruction.

The quality of cell image reconstruction is notably worse with a disentangled latent space in comparison with a VAE, as shown in Figure 5.27. However, the model is still able to capture main visual properties and shape.

5.2.2 Modeling the influence of topographies on cells for experiment simulation

In the second step, the encoder $q_{\varphi_f}(z_f|p)$ should learn the possible influence of topographies on cells, and specifically on cell area, by maximizing the objective (3.19). The only hyperparameter to fine-tune is β_{pf}. A control-case dataset was created by pairing topographies and cells randomly. Several models with different β_{pf} values were trained for 10-12 epochs with the learning rate 0.001. The validation results are provided in Table 5.6, where KL stands for $KL\left(q_{\varphi_f}(z_f|p)\,||\,p_{\theta^*_{pr}}(z_f)\right)$.

$\boldsymbol{\beta_{pf}}$	Relationship	Epoch	KL	Reconstruction error
10	main case	10	1.207	4728.3
10	control case	10	1.213	4736.7
50	main case	10	0.465	4746.1
50	control case	12	0.472	4749.7
100	main case	11	0.284	4759.8
100	control case	12	0.238	4764.9
200	main case	10	0.158	4780.4
200	control case	10	0.147	4782.7
400	main case	12	0.056	4799.6
400	control case	12	0.072	4799.9

Table 5.6: Selection of the hyperparameter β_{pf}.

It can be seen that increasing β_{pf} leads to lower KL and higher reconstruction error. However, no significant difference in reconstruction error between the main and control cases is observed at any level of β_{pf}. That implies the model has not identified any influence of topographies on cell area. Consequently, according to the model, any cell image is possible on any given topography.

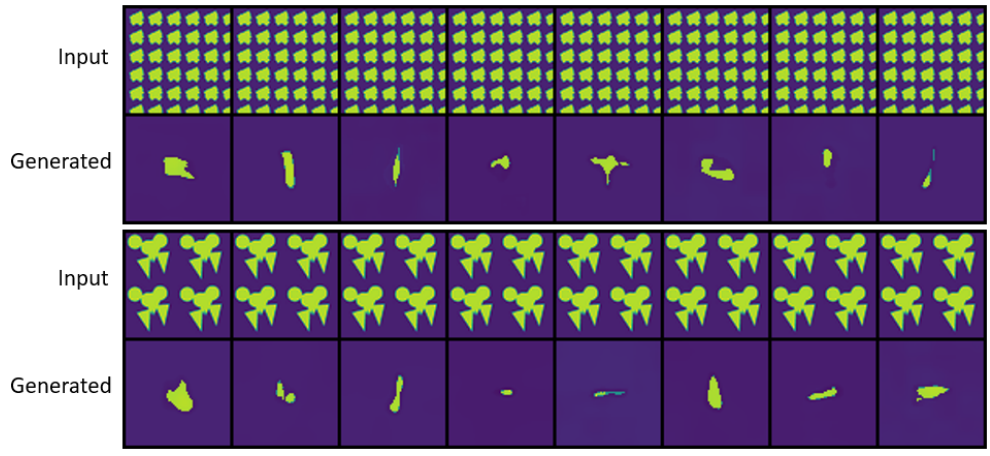

Figure 5.28: Simulation of the experiment: generating cell images conditioned on a topography image. No influence on cell area.

Since no influence of topographies on cell area has been identified, it is expected that in topography-conditioned cell image generation any single topography would result in a variety of cell images with arbitrary area. This is indeed observed, as demonstrated in Figure 5.28. Notably,

however, that mostly small- or middle-sized cells are generated, which is due to the fact that the full prior $p_{\theta_{pr}}(z_f)$ (Figure 5.22) does not cover the high-area region of the aggregated posterior distribution, as discussed above.

5.2.3 Modeling the inverse mapping for cell-conditioned topography design

In the last step of training (3.20) the goal is to model the inverse cell-topography relationship, which is achieved by forcing the model to use information captured by z_f, currently representing cell area, for topography reconstruction. However, since cell area was found to be unaffected by topographies, the latent variable z_f would not contribute in cell-conditioned topography design. Therefore, as compared with the synthetic-data case, only the first phase of training is followed, where $\eta = 0$ and only the reconstruction quality is pursued. Since $\eta = 0$, the model constitutes a plain VAE with a single hyperparameter β_l that controls the capacity of the latent space l_ε. The model was trained for approximately 500 epochs with the learning rate 0.001 and with β_l decreasing from 10 to 0.1. In Figure 5.29 the performance of the model in the task of topography image reconstruction is presented. It can be seen that the model is able to decently capture the details of topographies in all three size categories.

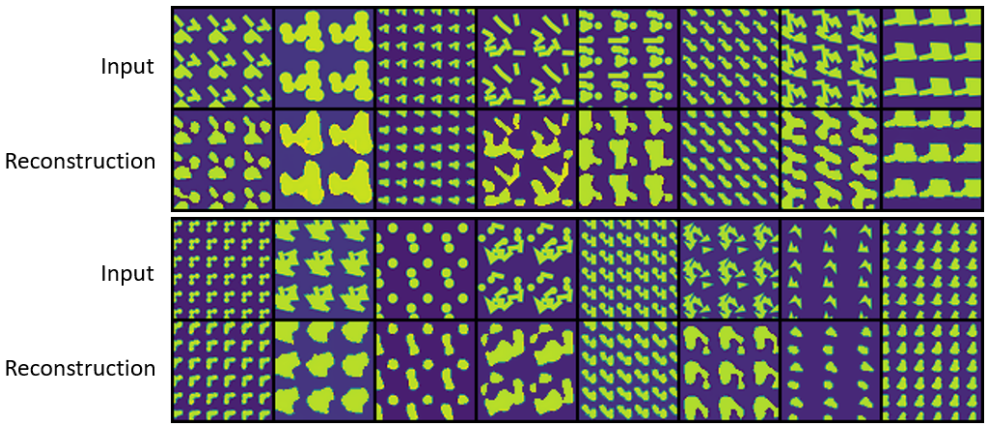

Figure 5.29: Topography image reconstruction.

Experimental results show that, indeed, setting $\eta > 0$ in the second phase does not improve the likelihood of the generated topographies conditioned on cells. However, even though the topography model is a VAE, and any cell area is possible on any given topography according to the model, it can still be used for unconditional topography design. Figure 5.30 shows a number of generated topographies as a result of sampling from the prior $p(l_\varepsilon)$. It can be seen that the model is able to generate topographies from specific size categories of the original data (top row). These are probably close to concrete topographies from the dataset. However, the model is also capable of generating out-of-domain topography samples with unseen patterns (bottow row).

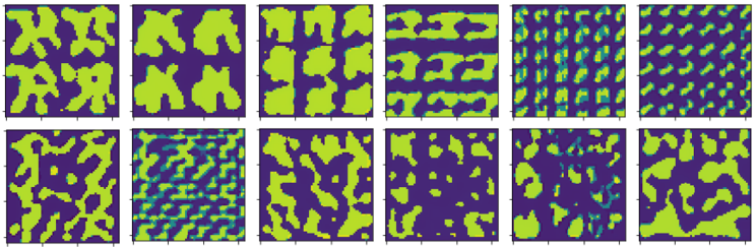

Figure 5.30: Unconditional topography generation: sampling from $p(l_\varepsilon)$.

Chapter 6

Conclusions

In this thesis we investigated whether the generative approach in machine learning can be used to simulate biomaterials research experiments in a data-driven fashion, in particular considering the experiments on cell-surface topography interaction. With the generative approach we aimed to address the key challenges inherent to such experiments: 1. the high dimensionality of both the input (topography design) space and the output (cell response) space, and 2. a high level of uncertainty and potential complexity of the relationship between these spaces.

We formulated a generative modeling perspective on the cell-surface topography experiment, according to which the observed cell response is regarded as an outcome of a generative process, subject to the experiment input conditions, represented by the surface topography. Specifically, the cell response is assumed to originate from a set of latent variables linked to particular measurable cell properties, and a residual noise latent variable. In turn, the surface topography is considered a factor affecting the probability distributions of the cell features-linked latent variables. We further proposed a deep generative simulation model that fits into the outlined perspective and derived an implementation for a special case, where both the cell response and the surface topography have an image representation. The proposed model acts as a simulation model of the physical experiment by generating cell images conditioned on a given surface topography. Furthermore, it acts as a tool for topography design, which can generate topographies associated with a given cell image based on data.

The proposed model was tested on two datasets, including a synthetic and a real-world dataset. In both cases a disentangled latent representation of cell images was derived, in which particular latent variables captured the data variation pertaining to particular cell features, while being disentangled from the residual variation in the data. The experimental results on the synthetic dataset showed that the model is capable to learn the embedded relationship between topographies and cells based on the provided image training data, which allowed to simulate cell images with accurately expressed topography-dependent characteristics in response to a given topography. Furthermore, the synthetic-case model accurately synthesized topography images that are likely to result in a given cell image according to the training data.

6.1 Limitations and future work

The proposed framework has a number of limitations. Firstly, in its current form, the model is limited to visual attributes of the cell response. Whereas other measures of the cell response, such as gene expression profiles or biomarker expression information may be of value for a researcher in certain applications. Hence, the challenge for future work is to find a way to learn latent representations of a combined-modality cell response, which could include both image and numerical data. Furthermore, numerical parameterizations of the topography design space could be more useful in topography production, since a generated topography image could be hard to translate to a production-compatible form.

Secondly, the model relies on the assumption that a cell response, represented by an image, is composed of a set of independent cell features, while the actual cell features, as evidenced by the ALP screening dataset, are to varying degrees correlated. Accordingly, learning a disentangled latent representation with independent latent variables linked to specific cell features may succeed in a limited number of cases, where independent cell features of interest are either carefully selected or artificially created. Hence, such an approach is neither salable, nor useful to automate screening data analysis and topography design in the general case. This observation suggests a future research direction aimed to drop the independence assumption and to develop latent representations

of the cell response that allow for correlated cell features, yet preserve interpretability at the same time. For instance, the model should desirably allow for cell feature value-conditioned generation of cell and topography images, which is not possible in a plain VAE model. One way to address this problem could be to consider a single latent variable z, which is, however, weakly stimulated to disentangle cell features-related factors of variation to the extent possible by encoding them in different latent dimensions. Hypothetically, this can be achieved by using a (multi-)conditional prior distribution $p_{\theta_f}(z|f_1,..,f_n)$, where a combination of all cell features serves as the condition. Additionally, a number of auxiliary regressors or classifiers $q_{\omega_{f_i}}(f_i|z)$ for all cell features of interest f_i can be used to further encourage disentanglement without directly splitting the latent space into independent subspaces. An example of a training objective for this perspective is provided below in (6.1).

$$\mathbb{E}_{q_\phi(z|x)}\ log\, p_\theta(x|z)\ -\ \beta\, KL\left(q_{\phi_f}(z|x)\,||\,p_{\theta_f}(z|f_1,..,f_n)\right)\ +\ \sum_i \alpha_{f_i}\ \mathbb{E}_{q_\phi(z|x)}\ log\, q_{\omega_{f_i}}(f_i|z) \tag{6.1}$$

Thirdly, the model proposed in this thesis uses a particularly strong an generally unjustified assumption that those factors of variation in topographies that influence certain cell features are 1. mutually independent and 2. are independent from the residual variation in topographies. Clearly, however, a single topography feature can influence multiple cell features, and a combination of topography features can influence a single or multiple cell features. Moreover, topography features can influence cell features only to a small extent. Hence, by forcing the residual latent variable l_ε of the topography model to ignore such topography features would result in the loss of information encoded by the topography-model latent space. Furthermore, topographies may in principle influence the residual variation in cell images, which is not currently of interest for the user, but the present model does not allow for such relationship, since the encoder $q(z_\varepsilon|p)$ is not included in the architecture.

Overall, it can be concluded that despite the idea to learn the relationship between two datasets in latent spaces is indeed warranted for cases with high dimensionality of the data and high uncertainty of the relationship, specifically the concept of a shared latent space as a means to achieve that has a limited applicability and can be useful only in cases with naturally factorized data. In future work, alternative ways to learn the relationship between the latent spaces of two distinct datasets could be considered. For instance, the relationship can be learned using auxiliary components $q(z_{cell}|z_{topography})$, $q(z_{topography}|z_{cell})$, representing the (probabilistic) mappings between the latent spaces of both datasets.

Finally, the current perspective on the relationship between datasets modeled in the latent space is limited in the type of relationship that can be learned. The current framework implies that a topography maps to a connected region in the z_f subspace. Whereas the situation when a topography could result in either low or high (but not intermediate) value of a cell feature f is not possible in the current model. To circumvent this problem, kernel-based methods could be embedded in the training process to allow for disjoint posterior distributions in the affected space.

Bibliography

[1] Andrea Asperti and Matteo Trentin. Balancing reconstruction error and kullback-leibler divergence in variational autoencoders. *IEEE Access*, 8:199440–199448, 2020. 10

[2] Piotr Baniukiewicz, E Josiah Lutton, Sharon Collier, and Till Bretschneider. Generative adversarial networks for augmenting training data of microscopic cell images. *Frontiers in Computer Science*, 1:10, 2019. 14

[3] Nick Beijer, Aliaksei Vasilevich, Bayram Pilavci, Roman Truckenmüller, Yiping Zhao, Shantanu Singh, Bernke Papenburg, and Jan Boer. Topowellplate: A well-plate-based screening platform to study cell–surface topography interactions. *Advanced Biosystems*, 1, 03 2017. 6, 7

[4] Nick RM Beijer, Zarina M Nauryzgaliyeva, Estela M Arteaga, Laurent Pieuchot, Karine Anselme, Jeroen van de Peppel, Aliaksei S Vasilevich, Nathalie Groen, Nadia Roumans, Dennie GAJ Hebels, et al. Dynamic adaptation of mesenchymal stem cell physiology upon exposure to surface micropatterns. *Scientific reports*, 9(1):1–14, 2019. 6

[5] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013. 11

[6] Tristan Bepler, Ellen D Zhong, Kotaro Kelley, Edward Brignole, and Bonnie Berger. Explicitly disentangling image content from translation and rotation with spatial-vae. *arXiv preprint arXiv:1909.11663*, 2019. 12

[7] Diane Bouchacourt, Ryota Tomioka, and Sebastian Nowozin. Multi-level variational autoencoder: Learning disentangled representations from grouped observations. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 12

[8] Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-vae. *arXiv preprint arXiv:1804.03599*, 2018. 12, 18

[9] Ricky TQ Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in variational autoencoders. *arXiv preprint arXiv:1802.04942*, 2018. 11, 13

[10] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 2180–2188, 2016. 11

[11] Zheng Ding, Yifan Xu, Weijian Xu, Gaurav Parmar, Yang Yang, Max Welling, and Zhuowen Tu. Guided variational autoencoder for disentanglement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7920–7929, 2020. 12

[12] Carl Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016. 8, 9

[13] Emilien Dupont. Learning disentangled joint continuous and discrete representations. *arXiv preprint arXiv:1804.00104*, 2018. 12

[14] Cian Eastwood and Christopher KI Williams. A framework for the quantitative evaluation of disentangled representations. In *International Conference on Learning Representations*, 2018. 13

[15] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010. 62

[16] Peter Goldsborough, Nick Pawlowski, Juan C Caicedo, Shantanu Singh, and Anne E Carpenter. Cytogan: generative modeling of cell images. *BioRxiv*, page 227645, 2017. 14

[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 8

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 62

[19] I. Higgins, L. Matthey, A. Pal, Christopher P. Burgess, Xavier Glorot, M. Botvinick, S. Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017. 11, 13, 18

[20] Frits FB Hulshof, Bernke Papenburg, Aliaksei Vasilevich, Marc Hulsman, Yiping Zhao, Marloes Levers, Natalie Fekete, Meint de Boer, Huipin Yuan, Shantanu Singh, et al. Mining for osteogenic surface topographies: In silico design to in vivo osseo-integration. *Biomaterials*, 137:49–60, 2017. 6, 32, 35

[21] Marc Hulsman, Frits Hulshof, Hemant Unadkat, Bernke J Papenburg, Dimitrios F Stamatialis, Roman Truckenmüller, Clemens Van Blitterswijk, Jan De Boer, and Marcel JT Reinders. Analysis of high-throughput screening reveals the effect of surface topographies on cellular morphology. *Acta biomaterialia*, 15:29–38, 2015. 2, 6, 7, 16, 36, 37

[22] Maximilian Ilse, Jakub M Tomczak, Christos Louizos, and Max Welling. Diva: Domain invariant variational autoencoders. In *Medical Imaging with Deep Learning*, pages 322–348. PMLR, 2020. 4, 12, 13, 17, 18, 62

[23] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015. 62

[24] Gregory R Johnson, Rory M Donovan-Maiye, and Mary M Maleckar. Generative modeling with conditional autoencoders: Building an integrated cell. *arXiv preprint arXiv:1705.00092*, 2017. 14

[25] Theofanis Karaletsos, Serge Belongie, and Gunnar Rätsch. Bayesian representation learning with oracle constraints. *arXiv preprint arXiv:1506.05011*, 2015. 12

[26] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pages 2649–2658. PMLR, 2018. 11, 13

[27] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 62

[28] Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in neural information processing systems*, pages 3581–3589, 2014. 12, 13

[29] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 4, 8, 9, 17

[30] Michal Kozubek. When deep learning meets cell image synthesis. *Cytometry Part A*, 97(3):222–225, 2019. 14

[31] Tejas D Kulkarni, Will Whitney, Pushmeet Kohli, and Joshua B Tenenbaum. Deep convolutional inverse graphics network. *arXiv preprint arXiv:1503.03167*, 2015. 12

[32] Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. *arXiv preprint arXiv:1711.00848*, 2017. 10, 11, 13

[33] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In *International conference on machine learning*, pages 1558–1566. PMLR, 2016. 49

[34] Bach Q Le, Aliaksei Vasilevich, Steven Vermeulen, Frits Hulshof, Dimitrios F Stamatialis, Clemens A van Blitterswijk, and Jan de Boer. Micro-topographies promote late chondrogenic differentiation markers in the atdc5 cell line. *Tissue Engineering Part A*, 23(9-10):458–469, 2017. 6, 7

[35] Daniëlle G Leuning, Nick RM Beijer, Nadia A Du Fossé, Steven Vermeulen, Ellen Lievers, Cees Van Kooten, Ton J Rabelink, and Jan De Boer. The cytokine secretion profile of mesenchymal stromal cells is determined by surface structure of the microenvironment. *Scientific reports*, 8(1):1–9, 2018. 6

[36] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pages 4114–4124. PMLR, 2019. 12, 13

[37] Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3. Citeseer, 2013. 62

[38] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015. 10

[39] Vlado Menkovski and Simon Koop. TU/e Deep Learning (2IMM10) course. Lecture notes: Generative models., 2020. 8, 9, 10

[40] Anton Osokin, Anatole Chessel, Rafael E Carazo Salas, and Federico Vaggi. Gans for biological image synthesis. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2233–2242, 2017. 14

[41] Andreas Reimer, Aliaksei Vasilevich, Frits Hulshof, Priya Viswanathan, Clemens Blitterswijk, Jan Boer, and Fiona Watt. Scalable topographies to support proliferation and oct4 expression by human induced pluripotent stem cells. *Scientific Reports*, 6:18948, 01 2016. 6, 7

[42] Luis A Pérez Rey, Vlado Menkovski, and Jacobus W Portegies. Diffusion variational autoencoders. *arXiv preprint arXiv:1901.08991*, 2019. 38

[43] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR, 2014. 8, 9

[44] Karl Ridgeway. A survey of inductive biases for factorial representation-learning. *arXiv preprint arXiv:1612.05299*, 2016. 11

[45] Karl Ridgeway and Michael C Mozer. Learning deep disentangled embeddings with the f-statistic loss. *arXiv preprint arXiv:1802.05312*, 2018. 13

[46] Lars Ruthotto and Eldad Haber. An introduction to deep generative modeling. *GAMM-Mitteilungen*, page e202100008, 2021. 8, 9, 10

[47] Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*, 2017. 62

[48] Geoffrey F Schau, Guillaume Thibault, Mark A Dane, Joe W Gray, Laura M Heiser, and Young Hwan Chang. Variational autoencoding tissue response to microenvironment perturbation. In *Medical Imaging 2019: Image Processing*, volume 10949, pages 407 – 415. International Society for Optics and Photonics, SPIE, 2019. 14

[49] Narayanaswamy Siddharth, Brooks Paige, Jan-Willem Van de Meent, Alban Desmaison, Noah D Goodman, Pushmeet Kohli, Frank Wood, and Philip HS Torr. Learning disentangled representations with semi-supervised deep generative models. *arXiv preprint arXiv:1706.00400*, 2017. 12, 13

[50] Hemant V Unadkat, Marc Hulsman, Kamiel Cornelissen, Bernke J Papenburg, Roman K Truckenmüller, Anne E Carpenter, Matthias Wessling, Gerhard F Post, Marc Uetz, Marcel JT Reinders, et al. An algorithm-based topographical biomaterials library to instruct cell fate. *Proceedings of the National Academy of Sciences*, 108(40):16565–16570, 2011. 1, 6, 7, 32, 35, 36

[51] Aliaksei Vasilevich, Aurélie Carlier, David A Winkler, Shantanu Singh, and Jan de Boer. Evolutionary design of optimal surface topographies for biomaterials. *Scientific Reports*, 10(1):1–10, 2020. 7

[52] Aliaksei Vasilevich and Jan de Boer. Robot-scientists will lead tomorrow's biomaterials discovery. *Current Opinion in Biomedical Engineering*, 6:74–80, 2018. 1

[53] Aliaksei S Vasilevich, Aurélie Carlier, Jan de Boer, and Shantanu Singh. How not to drown in data: a guide for biomaterial engineers. *Trends in biotechnology*, 35(8):743–755, 2017. 2, 4, 14

[54] Aliaksei S Vasilevich, Frédéric Mourcin, Anouk Mentink, Frits Hulshof, Nick Beijer, Yiping Zhao, Marloes Levers, Bernke Papenburg, Shantanu Singh, Anne E Carpenter, et al. Designed surface topographies control icam-1 expression in tonsil-derived human stromal cells. *Frontiers in bioengineering and biotechnology*, 6:87, 2018. 6, 7

[55] Aliaksei S Vasilevich, Steven Vermeulen, Marloes Kamphuis, Nadia Roumans, Said Eroumé, Dennie GAJ Hebels, Jeroen van de Peppel, Rika Reihs, Nick RM Beijer, Aurélie Carlier, et al. On the correlation between material-induced cell shape and phenotypical response of human mesenchymal stem cells. *Scientific reports*, 10(1):1–15, 2020. 32, 35

[56] William F Whitney, Michael Chang, Tejas Kulkarni, and Joshua B Tenenbaum. Understanding visual concepts with continuation learning. *arXiv preprint arXiv:1602.06822*, 2016. 12

[57] Christopher Yau and Kieran Campbell. Bayesian statistical learning for big data biology. *Biophysical reviews*, 11(1):95–102, 2019. 14

[58] Lee Zamparo and Zhaolei Zhang. Deep autoencoders for dimensionality reduction of high-content screening data. *arXiv preprint arXiv:1501.01348*, 2015. 14

[59] Qingyu Zhao, Ehsan Adeli, Nicolas Honnorat, Tuo Leng, and Kilian M Pohl. Variational autoencoder for regression: Application to brain aging analysis. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 823–831. Springer, 2019. 13

Appendix A

Implementation details

A.1 Training procedure details

The selection of hyperparameters for each of the two datasets, discussed in Chapter 5, is done using a hold-out validation dataset consisting of approximately 20% of the available data. For both datasets and in each step of the training procedure (3.3.1), the batch size is set to 50, and Adam optimizer [27] is used. The (starting) learning rate used by the optimizer is gradually reduced from 0.01 to 0.0001 based on the observed quantitative performance of the model on the validation set, and based on the qualitative visual performance of the model with respect to image reconstruction, latent space disentanglement and conditional image generation. The number of training epochs is adjusted accordingly.

Specifically for the case of real data (ALP screening dataset), a data augmentation technique proposed in [22] was used during training of the variational autoencoders for cell and topography images, as well as during training of the disentangled cell model. An input image is randomly flipped, horizontally and vertically, and is randomly rotated (by a multiple of 90 degrees) before being passed to the model. Notably, this technique is not applied during the second step of the training procedure (3.3.1) and during the third step (when $\eta > 0$) in order to preserve the actual relationship between topography and cell images. Furthermore, this technique is not used for the synthetic dataset, since rotation angle is assumed to be a cell feature of interest, and thus should not be randomly changed.

A.2 Model architecture

The architecture of the proposed model is conceptually the same for both considered datasets, as shown below, and is based on the architecture of Domain-Invariant Variational Autoencoder [22]. Furthermore, the authors' implementation [22] of ResNet [18] convolutional and transpose-convolutional blocks is used, referred to as 'ResidualConv2d' and 'ResidualConvTranspose2d', respectively. The output of both decoders $p_\theta(x|z_\varepsilon, z_f)$, $p_\vartheta(p|l_\varepsilon, z_f)$ constitutes a probability distribution over pixel values encoded in 100-dimensional vectors. The distribution is modeled using the logistic mixture technique, proposed in PixelCNN++ [47] and also used in DIVA [22]. Accordingly, the decoders produce tensors having shapes (batch size, 100, 128, 128) or (batch size, 100, 64, 64) for the synthetic-data and real-data cases respectively, which are directly used in the loss function, and are also used to sample individual 3-channel images. All fully connected and convolutional layers, except for the output layers, are followed by a batch normalization layer [23]. Leaky ReLU [37] is used as the activation function, and Xavier uniform initialization [15] is used for all weights of the model.

The tables A.1, A.2, A.3, A.4 describe the architectures of all components of the model for the synthetic (ToyCell) dataset, and the tables A.5, A.6, A.7, A.8 provide the same information for the real (ALP screening) dataset. In the tables, *bs* stands for batch size; the unnamed parameters in brackets show: the number of output features for linear layers, the number of output channels and kernel size for convolutional layers. The architectures of the decoders and encoders of the trained variational autoencoders, mentioned in Chapter 5, match the provided decoder/encoder architectures of the full model.

	Details	**Output shape**
1	Linear(1024), BatchNorm1d, LeakyReLU, Reshape(64, 4, 4)	(bs, 64, 4, 4)
2	ResidualConvTranspose2d(64, 3), LeakyReLU	(bs, 64, 4, 4)
3	Upsample(8)	(bs, 64, 8, 8)
4	ResidualConvTranspose2d(64, 3), LeakyReLU	(bs, 64, 8, 8)
5	Upsample(16)	(bs, 64, 16, 16)
6	ResidualConvTranspose2d(64, 3), LeakyReLU	(bs, 64, 16, 16)
7	Upsample(32)	(bs, 64, 32, 32)
8	ResidualConvTranspose2d(64, 3), LeakyReLU	(bs, 64, 32, 32)
9	Upsample(64)	(bs, 64, 64, 64)
10	ResidualConvTranspose2d(64, 3), LeakyReLU	(bs, 64, 64, 64)
11	Upsample(128)	(bs, 64, 128, 128)
12	Conv2d(100, 3, stride=1, padding=1)	(bs, 100, 128, 128)
13	Conv2d(100, 1, stride=1, padding=0)	(bs, 100, 128, 128)

Table A.1: ToyCell dataset: Architecture of the decoders $p_\theta(x|z_\varepsilon, z_f)$, $p_\vartheta(p|l_\varepsilon, z_f)$.

	Details	**Output shape**
1	Conv2d(32, 3, stride=1, padding=1), BatchNorm2d, LeakyReLU	(bs, 32, 128, 128)
2	ResidualConv2d(32, 3) (resize), LeakyReLU	(bs, 32, 64, 64)
3	ResidualConv2d(32, 3) (identity), LeakyReLU	(bs, 32, 64, 64)
4	ResidualConv2d(64, 3) (resize), LeakyReLU	(bs, 64, 32, 32)
5	ResidualConv2d(64, 3) (identity), LeakyReLU	(bs, 64, 32, 32)
6	ResidualConv2d(64, 3) (resize), LeakyReLU	(bs, 64, 16, 16)
7	ResidualConv2d(64, 3) (identity), LeakyReLU	(bs, 64, 16, 16)
8	ResidualConv2d(64, 3) (resize), LeakyReLU	(bs, 64, 8, 8)
9	ResidualConv2d(64, 3) (identity), LeakyReLU	(bs, 64, 8, 8)
10	ResidualConv2d(64, 3) (resize), LeakyReLU, Reshape(1024)	(bs, 1024)
11.1 (μ)	Linear(dim(z))	(bs, dim(z))
11.2 (σ)	Linear(dim(z)), Softplus	(bs, dim(z))

Table A.2: ToyCell dataset: Architecture of the encoders $q_{\phi_\varepsilon}(z_\varepsilon|x)$, $q_{\phi_f}(z_f|x)$, $q_{\varphi_l}(l_\varepsilon|p)$, $q_{\varphi_f}(z_f|p)$.

	Details	**Output shape**
1	Linear(dim(z_f)), BatchNorm1d, LeakyReLU	(bs, dim(z_f))
2.1 (μ)	Linear(dim(z_f))	(bs, dim(z_f))
2.2 (σ)	Linear(dim(z_f)), Softplus	(bs, dim(z_f))

Table A.3: ToyCell dataset: Architecture of the conditional prior $p_{\theta_f}(z_f|f)$.

	Details	**Output shape**
1	LeakyReLU, Linear(1)	(bs, 1)

Table A.4: ToyCell dataset: Architecture of the auxiliary regressor $q_{\omega_f}(f|z_f)$.

	Details	**Output shape**
1	Linear(4096), BatchNorm1d, LeakyReLU, Reshape(256, 4, 4)	(bs, 256, 4, 4)
2	ResidualConvTranspose2d(256, 3), LeakyReLU	(bs, 256, 4, 4)
3	Upsample(8)	(bs, 256, 8, 8)
4	ResidualConvTranspose2d(256, 3), LeakyReLU	(bs, 256, 8, 8)
5	Upsample(16)	(bs, 256, 16, 16)
6	ResidualConvTranspose2d(256, 3), LeakyReLU	(bs, 256, 16, 16)
7	Upsample(32)	(bs, 256, 32, 32)
8	ResidualConvTranspose2d(256, 3), LeakyReLU	(bs, 256, 32, 32)
9	Upsample(64)	(bs, 256, 64, 64)
10	Conv2d(100, 3, stride=1, padding=1)	(bs, 100, 64, 64)
11	Conv2d(100, 1, stride=1, padding=0)	(bs, 100, 64, 64)

Table A.5: ALP dataset: Architecture of the decoders $p_\theta(x|z_\varepsilon, z_f)$, $p_\vartheta(p|l_\varepsilon, z_f)$.

	Details	**Output shape**
1	Conv2d(32, 3, stride=1, padding=1), BatchNorm2d, LeakyReLU	(bs, 32, 64, 64)
2	ResidualConv2d(32, 3) (resize), LeakyReLU	(bs, 32, 32, 32)
3	ResidualConv2d(32, 3) (identity), LeakyReLU	(bs, 32, 32, 32)
4	ResidualConv2d(64, 3) (resize), LeakyReLU	(bs, 64, 16, 16)
5	ResidualConv2d(64, 3) (identity), LeakyReLU	(bs, 64, 16, 16)
6	ResidualConv2d(128, 3) (resize), LeakyReLU	(bs, 128, 8, 8)
7	ResidualConv2d(128, 3) (identity), LeakyReLU	(bs, 128, 8, 8)
8	ResidualConv2d(256, 3) (resize), LeakyReLU, Reshape(4096)	(bs, 256, 4, 4)
11.1 (μ)	Linear(dim(z))	(bs, dim(z))
11.2 (σ)	Linear(dim(z)), Softplus	(bs, dim(z))

Table A.6: ALP dataset: Architecture of the encoders $q_{\phi_\varepsilon}(z_\varepsilon|x)$, $q_{\phi_f}(z_f|x)$, $q_{\varphi_l}(l_\varepsilon|p)$, $q_{\varphi_f}(z_f|p)$.

	Details	**Output shape**
1	Linear(dim(z_f)), BatchNorm1d, LeakyReLU	(bs, dim(z_f))
2.1 (μ)	Linear(dim(z_f))	(bs, dim(z_f))
2.2 (σ)	Linear(dim(z_f)), Softplus	(bs, dim(z_f))

Table A.7: ALP dataset: Architecture of the conditional prior $p_{\theta_f}(z_f|f)$.

	Details	**Output shape**
1	LeakyReLU, Linear(1)	(bs, 1)

Table A.8: ALP dataset: Architecture of the auxiliary regressor $q_{\omega_f}(f|z_f)$.