Eötvös Loránd University

Faculty of Science

Investigation of importance weighting for representation learning in hierarchical Variational Autoencoders

Thesis

Bernadett Vas

Applied Mathematics MSc

Computer Science specialization

Supervisors:

Gergő Orbán

Ferenc Csikor

HUN-REN Wigner Research Centre for Physics

Department of Computational Sciences

Budapest, 2024

Contents

Acknowledgements

I would like to express my gratitude to my supervisors, Gergő Orbán and Ferenc Csikor for introducing me with the topic of this work, and for guiding me through in writing this thesis. Your inspiring explanations and constant feedbacks contributed much to complete this project and enabled me to dive into an amazing research topic which impressed me in several ways. Thank you for all your patience and help, without which this work could not have been completed.

I would like to thank András Lukács for all his time and energy he put in giving me valuable advises, both professionally and personally throughout my studies. I gained much experience during our collaborations, while you also gave me the chance to join interesting research activities, which enriched my university studies.

I am grateful for my family and friends, who are always there for me, and I can always count on their help. Thank you for the endless support I got from you not only during the writing of this thesis but also through all the years at the university.

Introduction

Representation learning is an intensely studied field within machine learning, which aims to extract useful representations from data. Learning valuable representations is a key task in machine learning since it is necessary for designing efficient algorithms. Moreover, they can be examined for interpretability purposes as well as it is assumed that they encode many explanatory factors [2].

Nowadays, deep learning is one of the most popular areas in machine learning due to its powerful models which have unprecedented performance across a wide range of applications and tasks. Hence it comes as no surprise that more attention is directed towards them in representation learning as well. These algorithms possess desirable properties such as successfully dealing with non-linear, complex relationships in the data, expressivity and scalability to large datasets. An important family of models in deep learning widely applied for learning valuable features are generative models, especially Variational Autoencoders (VAE). In this thesis the centre of attention is the Importance Weighted Variational Autoencoder (IWAE), which is examined in an extended, hierarchical form. Instead of a single stochastic latent layer, a hierarchy of two layers is learned, and a key point in this work is to investigate the representational properties of this extended model.

The starting point of this thesis are the works from Csikor et al. [9], [8], which have the purpose of building hierarchical VAEs that can appropriately model the early visual cortex of mammals. The main objective of their work was to utilize deep generative networks to create a model of the first and second primary visual cortices, V1 and V2, analyze the representations learnt by this model and compare them to the responses found in the visual cortices of mammals, particularly of monkeys. The TDVAE network is amongst the proposed models, which is a two-latent hierarchical VAE featuring a top-down recognition model, enhanced with components inspired by neuroscience.

The present work sets the goal to extend the TDVAE model by incorporating the importance weighted scheme introduced by the IWAE. Furthermore, to carry out an analysis of the emerged representation in both of the models and compare the results. Building on the appealing properties of IWAE, such as enriching the learnt representation and learning a more flexible and exotic variational posterior, our expectations are that the TDVAE model enhanced with importance weighting (TD-IWAE) will result in both more accurate and expressive features regarding the learnt posterior than the one learnt by TDVAE.

Although, the topic of the present work lies in the intersection of two extensive fields of deep generative learning and neuroscience, it will mainly focus on the background theory and construction of the presented models along with the analysis of the experiments performed on the learned representations.

The structure of this thesis is the following. In Chapter 1, a brief overview is presented about topics essential in the present work. An outline about VAE models, the role of generative modelling in neuroscience and importance sampling is discussed. Chapter 2 details the main topic of the thesis, the Importance Weighted Autoencoders. Chapter 3 presents an introduction to hierarchical VAE models, and then the TDVAE model is described. In Chapter 4, the TDVAE model extended with the IWAE scheme is proposed. From now on, we will refer to this model as TD-IWAE. The experimental results are presented in Chapter 5, concerning the training of the implemented models, and the analysis performed on the learned representations. The IWAE was implemented both for the standard single latent layer VAE, and for its extended, hierarchical version. The natural images trained non-hierarchical IWAE is point of interest on its own but I chose to focus on my main contribution, the extension of this model to the hierarchical version. Lastly, Chapter 6 contains a summary of the present work with possible future research directions.

Chapter 1

Background

1.1 Variational Autoencoder, a brief introduction

In this section we will review the topics fundamental in the present work, crucial to have an insight in order to understand the further chapters: the VAE and variational inference. The section is written based on the works [17], [18], [6], [3].

Probabilistic models have a central role in machine learning and they are used in numerous applications. Their purpose is to provide an unsupervised framework for learning about data. Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ be our dataset, containing iid data points, and for the sake of simplicity suppose that $\mathbf{x}$ denotes one single data point from this dataset. It is assumed that there is an unknown process $p^*(\mathbf{x})$ generating our data and we wish to learn the parameters $\boldsymbol{\theta}$ of a model $p_\theta(\mathbf{x})$ approximating this process.

A wide-spread and effective approach to learn the distribution of the data $p(\mathbf{x})$ is to involve unobservable variables, called latent variables into the learning process. These can be interpreted as factors corresponding to essential information inferred from the observable variables. The models utilizing latent variables are called Latent Variable Models and they seek to learn the joint distribution of the latent $\mathbf{z}$ and observed $\mathbf{x}$. Here we are interested in continuous latents. Since we are looking for $p_\theta(\mathbf{x})$, we have to marginalize out the latent factor: $p_\theta(\mathbf{x}) = \int_z p_\theta(\mathbf{x}, \mathbf{z})\, d\mathbf{z}$.

Maximum likelihood estimation is a common method to fit probabilistic models, where the goal is to find the optimal parameters θ which maximizes the log-likelihood $\log p_\theta(\mathbf{x})$. In other words, we are looking for a model that best explains our data. Since our dataset consists of iid data points, the marginal log-likelihood breaks down to a sum of single data points: $\log p_\theta(\mathbf{x}_1, ..., \mathbf{x}_N) = \sum_{i=1}^{N} \log p_\theta(\mathbf{x}_i)$. For clarity of the overview, from now on let's only consider the log-likelihood of one data point $\mathbf{x}$.

To complete the model, the formulation of the joint requires the specification of the likelihood function as well as the prior distribution. Here, we assume that the latent variables are generated according to the predetermined prior $p_\theta(\mathbf{z})$. Taking into consideration the prior of the latent as well, the joint can be decomposed as the product of the posterior and prior distribution. This factorization has the advantage that it relies on more easily computable distributions than the joint posterior. Hence, the marginal log-likelihood is in the form of

$$\log p_\theta(\mathbf{x}) = \log \int_z p_\theta(\mathbf{x}, \mathbf{z})\, d\mathbf{z} = \log \int_z p_\theta(\mathbf{x}|\mathbf{z}) p_\theta(\mathbf{z})\, d\mathbf{z}$$

This factorization assumes sampling the prior, and describes a generative process of $\mathbf{x}$ given the corresponding $\mathbf{z}$: As the first step, draw a sample from the prior $\mathbf{z} \sim p_\theta(\mathbf{z})$, and then use the generated $\mathbf{z}$ to obtain a sample $\mathbf{x} \sim p_\theta(\mathbf{x}|\mathbf{z})$. In addition, note that the factorization reflects the mathematical formulation of the associated directed graphical model.

A problem arises when the above integral $\int_z p_\theta(\mathbf{x}, \mathbf{z})\, d\mathbf{z}$ is intractable and we are not able to evaluate it analytically. This could happen owing to the high dimensionality of the latent space or the complex, non-linear relationship between the latent variables and the observed ones defined by the likelihood. As a consequence, the posterior $p_\theta(\mathbf{z}|\mathbf{x})$ is also intractable, due to its relationship with the marginal likelihood given by Bayes' theorem:

$$p_\theta(\mathbf{z}|\mathbf{x}) = \frac{p_\theta(\mathbf{x}|\mathbf{z}) p_\theta(\mathbf{z})}{p_\theta(\mathbf{x})} \tag{1.1}$$

The lack of feasible evaluation of the marginal likelihood leads us back to apply approximation schemes in such scenarios. The approximation method called variational inference is a state-of-the-art approach in the context of latent variable models, and it is gaining more popularity thanks to its application in deep learning. The basis of variational inference lies on Bayesian inference, and its aim is to search for a tractable distribution which can be used as an approximation for $p_\theta(\mathbf{z}|\mathbf{x})$. The best known deep latent variable model is the Variational Autoencoder (VAE) originally proposed by Kingma et al. [17]. Also, it was a parallel discovery by Rezende et al. [28]. The task, just as in the VAE model, is formulated as an optimization problem, where we wish to find the variational distribution $q_\phi(\mathbf{z}|\mathbf{x})$ which is the closest to the true posterior in terms of Kullback-Leibler divergence (KL-divergence). The VAE performs variational inference by calculating the approximate posterior $q(\mathbf{z}|\mathbf{x})$ with deep neural networks for the continuous latent $\mathbf{z}$.

Due to the intractability, in the VAE the optimization is not directly performed on the marginal log-likelihood, instead the Evidence Lower Bound is used. As the name suggests, it is a lower bound on $\log p_\theta(\mathbf{x})$, derived for one single data point as follows:

$$\begin{aligned}
\log p_\theta(\mathbf{x}) &= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x})\right] \\
&= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{p_\theta(\mathbf{z}|\mathbf{x})}\right] \\
&= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} \cdot \frac{q_\phi(\mathbf{z}|\mathbf{x})}{p_\theta(\mathbf{z}|\mathbf{x})}\right] \\
&= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{q_\phi(\mathbf{z}|\mathbf{x})}{p_\theta(\mathbf{z}|\mathbf{x})}\right] \\
&= \mathcal{L}_{VAE}(\mathbf{x}) + KL[q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x})]
\end{aligned} \tag{1.2}$$

What we get is the sum of the KL-divergence between the variational and the true posterior and another term which will be used as the optimization objective for the model. It would be straightforward to minimize the KL-divergence of the variational distribution however we are not able to evaluate $p_\theta(\mathbf{z}|\mathbf{x})$. Fortunately, the KL-divergence is always non-negative, therefore simply omitting it from the above equation we can arrive to a lower bound on $\log p_\theta(\mathbf{x})$ which is called the Evidence Lower Bound (ELBO). The great thing is that by maximizing the ELBO we can achieve two of our goals: first, maximizing $\log p_\theta(\mathbf{x})$ and second, improving the tightness of the bound, we can decrease the KL-divergence between the variational and true posterior.

The ELBO can also be derived by applying Jensen's inequality. This way is less illustrative but shows an important trick to arrive to the ELBO [6]:

$$\begin{aligned}
\log p_\theta(\mathbf{x}) &= \log \int_z p_\theta(\mathbf{x}|\mathbf{z}) p_\theta(\mathbf{z})\, d\mathbf{z} \\
&= \log \int_z p_\theta(\mathbf{x}|\mathbf{z}) p_\theta(\mathbf{z}) \frac{q_\phi(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})}\, d\mathbf{z} \\
&= \log \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\frac{p_\theta(\mathbf{x}|\mathbf{z}) p_\theta(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] \\
&\geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x}|\mathbf{z}) p_\theta(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] \\
&= \mathcal{L}_{VAE}(\mathbf{x})
\end{aligned} \tag{1.3}$$

In order to see from what components the ELBO is built, we have to further alter its form:

$$\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x}|\mathbf{z}) p_\theta(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] - KL[q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z})] \tag{1.4}$$

The first term is the reconstruction term, indicating how well $\mathbf{x}$ is decoded from its latent $\mathbf{z}$, and the second term is the KL-divergence between the variational posterior and the prior which can be seen as a regularizer for $\boldsymbol{\phi}$.

The Variational Autoencoder as mentioned before, is a latent variable model which performes variational inference with deep neural networks. It is comprised of a recognition and a generative network. The recognition model is a probabilistic encoder as it returns a probability distribution for a given $\mathbf{x}$, namely the approximate posterior $q_\phi(\mathbf{z}|\mathbf{x})$. Similarly the generative part is a stochastic decoder, and for a given $\mathbf{z}$ it outputs the distribution for $p_\theta(\mathbf{x}|\mathbf{z})$. An illustration of VAE can be observed in Figure 1.1.

An important property of VAE that it performs amortized inference, which means that the variational parameters $\boldsymbol{\phi}$ are shared between the observations. Instead of optimizing the model for each of the individual data points with a different group of parameters, it learns one collection of $\boldsymbol{\phi}$ and make use of them to output a posterior distribution for each $\mathbf{x}$.

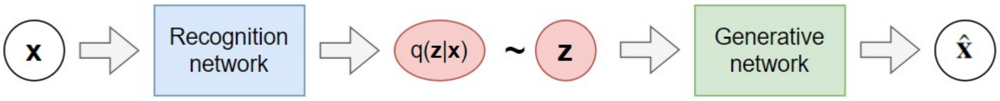


Figure 1.1: The architecture of VAE

The recognition and generative networks are jointly optimized with respect to $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$. Calculating the gradients of the ELBO for the optimization procedure is a bit tricky since it contains an expectation corresponded with the sampling from $q_\phi(\mathbf{z}|\mathbf{x})$. There is no problem of computing the gradients with respect to $\boldsymbol{\theta}$, as the differentiation and the expectation is commutative. The case of the variational parameters $\boldsymbol{\phi}$ is more complicated, therefore the reparametrization trick was introduced in Kingma et al. [17].

The strategy is to introduce a random noise variable $\boldsymbol{\epsilon}$ which is independent both of $\mathbf{x}$ and

$\boldsymbol{\phi}$, then express $\mathbf{z}$ as a deterministic, differentiable function of $\boldsymbol{\epsilon}$: $\mathbf{z} = g(\boldsymbol{\epsilon}, \boldsymbol{\phi}, \mathbf{x})$. It follows that from now on the expectation is performed over the sampling $\boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$ and the differentiation operation with respect to $\boldsymbol{\phi}$ can be moved inside the expectation. As the last step, in both cases the expectation is approximated with Monte Carlo estimation. The detailed derivation can be found in the original paper [17]. The optimization procedure can be performed with stochastic optimization algorithms as it is traditional in deep learning.

Conventionally in the VAE framework the distribution for $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ is assumed to be Normal or Bernoulli, the posterior $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$ is parameterized as a Normal with diagonal covariance matrix, and the prior is chosen to follow a standard Gauss distribution. In this scenario, the KL-divergence can be calculated with a simple analytical expression. The approximation for $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$ is performed by MLPs.

1.2 Neuroscience and generative modelling

To provide a an overview of generative modelling and its role in neuroscience, the sources [14], [18], [8] [9], [34] were used.

When considering biological vision, machine learning models of natural images can be used to investigate the properties of the visual processes. For this, machine learning offers two major classes of models, which are generative modelling and discriminative modelling.

Disciriminative models are trained in a supervised manner to solve a specific task, for instance to differentiate between data points, using the observations as input and their corresponding targets. In contrast, generative models aim to understand how the data is generated and to approximate the underlying distribution from where the data is coming from. It solves a more general task by estimating a joint probability distribution of the variables instead of learning a conditional probability of the targets given the observations. Note that generative models which learn $p(\mathbf{x}, \mathbf{z})$ can also be used to predict $p(\mathbf{z}|\mathbf{x})$ by applying Bayes-rule. Other appealing properties of generative models are that they can be used to improve prediction tasks by utilizing them as feature extractors. In addition, since they are trained unsupervised and can be used to generate data, one can make use of them to produce more labelled data or to augment the underlying training set with new data points.

In neuroscience it is a common practice to utilize unsupervised generative modelling of natural images to understand visual perception [9]. This approach is motivated by multiple factors. First, the unsupervised training paradigm better for a biological learning scenario as training labels are unfeasible to assume. Second, the latent variable generative modelling framework permits a richer representation to be learned. Models that are trained to solve predetermined tasks learn features that are specific to the underlying task. In contrast, with utilizing generative models task general representations can be studied. Third, probabilistic representations are essential to face the natural challenges of humans and other animals, hence a computational framework that accommodates such probabilistic computations are biologically important to examine.

As deep learning tools have come into focus, considerable amount of work has been put on investigating deep models as well. Their ability to express complex, non-linear relationships make them a favorable tool in modelling. The VAE is popular tool used in deep unsupervised representation learning, therefore it is does not come as a surprise that they were utilized in modelling the visual system as well, like in [12]. In fact, several reasons stands for why these networks can be used to arrive to a genuine model of the neuronal processes in the visual system.

First of all, since generative models are trained to learn the underlying distribution from where the data is generated from, they are required to capture intricate information patterns. This property allows them to be effectively trained on natural images that are known to possess a complicated structure, and also to create useful representations of the observations. In contrast, models trained in a task-specific way may not flexible enough to capture the required complex patterns.

Secondly, VAEs can easily architectually adapt to model the computations and anatomy of the visual pathway in mammals, as it will be further detailed in later chapters. For instance, a key feature of the visual cortex is that it possess a hierarchical structure in terms of processing the input signal. The two main studies on which this present work is based [9], [8] exploited the properties of VAE models that they can be generalized to a hierarchical structure, providing a match with the biological visual pathway. From an interpretability perspective, parallels can be drawn between these models and the anatomy of the visual cortex.For instance, the stochastic layers from different hierarchy levels can be associated with the hierarchical layers of the visual cortex, or the activity of model neurons can be interpreted as the responses of cortical neurons to visual stimuli.

In relation with the previous argument, top-down connections, which is a crucial property of the visual hierarchy, can also be modelled by utilizing hierarchical VAEs. The cortical hierarchy, along with the feed-forward computations, also holds top-down interactions, meaning that information flows from higher cortical areas to lower layers as well, providing contextual knowledge to the first layers. One would ask why this could be useful? During perception inferences are made about features at different levels of the hierarchy, and this inference is distorted by sensory noise and occlusion. As it was stated in [34], performing basic visual tasks is hard, meaning that the interpretation of visual information at this level could be uncertain. However, high levels might help this inference by providing the lower layers with learned information. For instance the orientation of an edge that is represented in a lower level corrupted area might not be established with high certainty, but higher layers could transfer information about co-occurrences of edges, forming the representation of the edge by others in its surrounding visual fields. Based on this, one can think of the top-down path in the visual system that the high-level signals can assist the processing of lower layers in the hierarchy. Therefore it is essential to incorporate top-down connections in the model of the visual pathway, as it is in the case of TDVAE which is in the centre of this thesis.

1.3 Importance Sampling

This section briefly outlines the key concepts present in this thesis, such as Monte Carlo estimation and Importance Sampling, using the works of [24], [3], [30].

In many applications we can face with the problem of evaluating an expected value. Given the expression $\mu = \mathbb{E}_{p(\mathbf{x})}[f(\mathbf{x})] = \int p(\mathbf{x})f(\mathbf{x})d\mathbf{x}$ where f is an integrable function and the expectation is taken under the distribution p, its possible that the integral cannot be exactly calculated (for example the distribution is complex or the dimension of $\mathbf{x}$ is high). In these cases, one can apply Monte Carlo method to approximate the above integral. Drawing N idependent samples from $p(\mathbf{x})$, the Monte Carlo estimator is in the form:

$$\hat{\mu}_N = \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}_i) \qquad\qquad \mathbf{x}_i \sim p(\mathbf{x})$$

From the Law of Large Numbers, the estimator is unbiased, $\mathbb{E}_{p(\mathbf{x})}[\hat{\mu}_N] = \mu$, and its variance is $Var[\hat{\mu}_N] = \frac{\sigma^2}{N}$ where $\sigma^2 = Var_{p(\mathbf{x})}[f(\mathbf{x})]$.

Importance Sampling is a Monte Carlo method which serves as a technique for the above problem, namely to estimate the expected value under the distribution $p(\mathbf{x})$, in the case when $p(\mathbf{x})$ is too complicated and it is not straightforward to sample from it.

We assume that it is easy to evaluate the target distribution $p(\mathbf{x})$ in a given point $\mathbf{x}$. The trick consists of choosing a different distribution $q(\mathbf{x})$, from which we can easily sample from, obtain samples from $q(\mathbf{x})$ instead from $p(\mathbf{x})$ and then make corrections in the result as one would sampled from $p(\mathbf{x})$. The correction is required since for example the samples drawn from regions where $q(\mathbf{x})$ assigns greater probability than $p(\mathbf{x})$ would do, will be over-represented. The correction terms will be considered as importance weights. To put it precisely, our expression can be rewritten as:

$$\mathbb{E}_{p(\mathbf{x})}[f(\mathbf{x})] = \int f(\mathbf{x})p(\mathbf{x})d\mathbf{x} = \int f(\mathbf{x})\frac{p(\mathbf{x})}{q(\mathbf{x})}q(\mathbf{x})d\mathbf{x} = \mathbb{E}_{q(\mathbf{x})}\left[f(\mathbf{x})\frac{p(\mathbf{x})}{q(\mathbf{x})}\right]$$

Recalling that we can approximate the integral by drawing independent samples from $q(\mathbf{x})$ and taking the average:

$$\mathbb{E}_{q(\mathbf{x})}\left[f(\mathbf{x})\frac{p(\mathbf{x})}{q(\mathbf{x})}\right] \approx \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}_i)\frac{p(\mathbf{x}_i)}{q(\mathbf{x}_i)} \qquad\qquad \mathbf{x}_i \sim q(\mathbf{x})$$

The terms $\frac{p(\mathbf{x}_i)}{q(\mathbf{x}_i)}$ are the importance weights, the correction quantities counting for the fact that we did not sample from the target distribution. Note that since the expectation is taken with respect to $q(\mathbf{x})$ it is also an unbiased estimator of μ. As intuition suggests, the result of the estimator highly depends on how much $q(\mathbf{x})$ is similar to the target $p(\mathbf{x})$. A key prerequisite regarding the distribution $q(\mathbf{x})$ is that it shouldn't be zero for all locations where the target $p(\mathbf{x})$ is non-zero, meaning that we should able to draw samples for all $\mathbf{x}$ to which $p(\mathbf{x})$ assigns non-zero probability.

Importance Sampling is a technique to calculate an expected value under some distribution, but does not give a method to generate samples from the underlying distribution. Fortunately, the algorithm Sampling-Importance-Resampling (SIR) [30] provides us a way to obtain samples from the target distribution $p(\mathbf{x})$. As before, assume we have a proposal distribution $q(\mathbf{x})$. The procedure consists of two steps:

1. Generate an independent random sample $\mathbb{X} = \{X_1, ..., X_n\}$ from $q(\mathbf{x})$. Calculate the weights $w_i = \frac{p(\mathbf{x}_i)}{q(\mathbf{x}_i)}$ for every i, $i = 1, .., n$ and normalize them.
2. Resample from the obtained sample $\mathbb{X} = \{X_1, ..., X_n\}$ with replacement according to the normalized weights. The resulted sample $\hat{\mathbb{X}} = \{\hat{X}_1, ..., \hat{X}_m\}$ is usually smaller than $\mathbb{X}$.

As $n \to \infty$ the resulted sample $\hat{\mathbb{X}}$ approximately can be seen as we have sampled from the underlying distribution $p(\mathbf{x})$.

Chapter 2

Importance Weighted Autoencoders

2.1 Overview and detailed properties of IWAEs

The optimization objective of the standard VAE is the Evidence Lower Bound on the data log-likelihood. The authors of [4] introduced a different approach to train a VAE, and demonstrated that the proposal has several advantages compared to the baseline VAE. The new model is called the Importance Weighted Autoencoder (IWAE). As opposed to a single sample drawn from the variational posterior, the IWAE objective builds on possible arbitrary number of samples. The goal of this section is to summarize the proposed IWAE model based on its original paper [4].

The main incentive for using a different objective is that the baseline VAE ELBO restricts the expressivity of the network. This limitation is caused by the assumption that the variational posterior is characterized by a simplified parametric form, which presumption may be too strict regarding the shape of the posterior. In addition, typically one sample is drawn from the recognition model during training hence the objective strictly penalizes the posterior samples which cannot explain the observed data properly. This phenomenon can happen if the majority of the samples are drawn from low probability regions of the variational posterior. The objective severly penalizes the samples which are not drawn from high probability locations, nevertheless its possible that the recognition network can capture these high probability regions and perform proper inference.

The authors of [4] proposed to overcome these drawbacks by using multiple, independently drawn samples from the distribution $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$. Utilizing more samples during traning leads to a more efficient use of samples. The IWAE objective is a variant of the vanilla ELBO, and defined as follows:

$$\mathcal{L}_{IWAE}^{k}(\mathbf{x}) = \mathbb{E}_{\mathbf{z}_1,\mathbf{z}_2,\ldots,\mathbf{z}_k \sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log \frac{1}{k}\sum_{i=1}^{k}\frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z}_i)}{q_{\boldsymbol{\phi}}(\mathbf{z}_i|\mathbf{x})}\right] \tag{2.1}$$

The IWAE model shares the same architecture with the traditional VAE, but uses $\mathcal{L}_{IWAE}^{k}$ instead of the standard baseline ELBO $\mathcal{L}_{VAE}$. In the above expression the term $\frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z}_i)}{q_{\boldsymbol{\phi}}(\mathbf{z}_i|\mathbf{x})} = w_i$ is called the unnormalized importance weight of the sample $\mathbf{z}_i$.

In the work of Burda et al.[4] several properties of the IWAE ELBO are proven. First, one can observe that the modified ELBO is also a lower bound on $\log p_{\boldsymbol{\theta}}(\mathbf{x})$. This follows from Jensen's inequality and the fact that the unnormalized importance weights are an unbiased estimator of $p_\theta(\mathbf{x})$:

Theorem 2.1.1. $\log p_\theta(\boldsymbol{x}) \geq \mathcal{L}^k_{IWAE}(\boldsymbol{x})$

Proof. Observe that $\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[w] = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] = \int_{\mathbf{z}_i} q_\phi(\mathbf{z}_i|\mathbf{x})\frac{p_\theta(\mathbf{x},\mathbf{z}_i)}{q_\phi(\mathbf{z}_i|\mathbf{x})}\, d\mathbf{z}_i = p_\theta(\mathbf{x})$

Then, by applying Jensen's inequality:

$$\mathcal{L}^k_{IWAE}(\mathbf{x}) = \mathbb{E}_{\mathbf{z}_1,\ldots,\mathbf{z}_k\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\log\left(\frac{1}{k}\sum_{i=1}^{k} w_i\right)\right] \leq \log\left(\mathbb{E}_{\mathbf{z}_1,\ldots,\mathbf{z}_k\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\frac{1}{k}\sum_{i=1}^{k} w_i\right]\right) =$$

$$= \log\left(\frac{1}{k}\sum_{i=1}^{k}\mathbb{E}_{\mathbf{z}_1,\ldots,\mathbf{z}_k\sim q_\phi(\mathbf{z}|\mathbf{x})}[w_i]\right) = \log p_\theta(\mathbf{x})$$

□

The main advantage of $\mathcal{L}^k_{IWAE}$ that it is a strictly tighter lower bound on the log-likelihood than the conventional ELBO objective. Moreover, the more samples are drawn from the approximate posterior, the tighter the bound is:

$$\log p_\theta(\mathbf{x}) \geq \mathcal{L}^{k+1}_{IWAE}(\mathbf{x}) \geq \mathcal{L}^k_{IWAE}(\mathbf{x})$$

Theorem 2.1.2.

1. $\mathcal{L}^k_{IWAE} \geq \mathcal{L}^m_{IWAE}$ *for* $k \geq m$
2. *If* $\frac{p_\theta(\boldsymbol{x},z_i)}{q_\phi(z_i|\boldsymbol{x})}$ *is bounded, then* $\lim_{k\to\infty}\mathcal{L}^k_{IWAE}(\boldsymbol{x}) = \log p_\theta(\boldsymbol{x})$

Proof.

1. The proof builds on Jensen's inequality and on the observation that given a sequence of k numbers $a_1,\ldots,a_k$, choosing uniformly at random $m \leq k$ distinct elements $a_{i_1},\ldots,a_{i_m}$ from them, then the following holds: $\mathbb{E}_{a_{i_1},\ldots,a_{i_m}}\left[\frac{\sum_{j=1}^m a_{i_j}}{m}\right] = \frac{\sum_{i=1}^k a_i}{k}$. The equality is true due to the linearity of expectation and the fact that each elements are chosen with equal probability, which implies that $\mathbb{E}[a_{i_j}] = \sum_{i=1}^k a_i \cdot \frac{1}{k}$. Applying the above we can derive that

$$\begin{aligned}
\mathcal{L}^k_{IWAE}(\mathbf{x}) &= \mathbb{E}_{\mathbf{z}_1,\ldots,\mathbf{z}_k\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\log\frac{1}{k}\sum_{i=1}^{k}\frac{p_\theta(\mathbf{x},\mathbf{z}_i)}{q_\phi(\mathbf{z}_i|\mathbf{x})}\right] \\
&= \mathbb{E}_{\mathbf{z}_1,\ldots,\mathbf{z}_k\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\log\mathbb{E}_{\mathbf{z}_{i_1},\ldots,\mathbf{z}_{i_m}\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\frac{1}{m}\sum_{j=1}^{m}\frac{p_\theta(\mathbf{x},\mathbf{z}_{i_j})}{q_\phi(\mathbf{z}_{i_j}|\mathbf{x})}\right]\right] \\
&\geq \mathbb{E}_{\mathbf{z}_1,\ldots,\mathbf{z}_k\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\mathbb{E}_{\mathbf{z}_{i_1},\ldots,\mathbf{z}_{i_m}\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\log\frac{1}{m}\sum_{j=1}^{m}\frac{p_\theta(\mathbf{x},\mathbf{z}_{i_j})}{q_\phi(\mathbf{z}_{i_j}|\mathbf{x})}\right]\right] \qquad (2.2)\\
&= \mathbb{E}_{\mathbf{z}_1,\ldots,\mathbf{z}_m\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\log\frac{1}{m}\sum_{i=1}^{m}\frac{p_\theta(\mathbf{x},\mathbf{z}_i)}{q_\phi(\mathbf{z}_i|\mathbf{x})}\right] \\
&= \mathcal{L}^m_{IWAE}(\mathbf{x}).
\end{aligned}$$

2. Consider the iid samples $\mathbf{z}_1, ..., \mathbf{z}_k \sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ and the random variables $Y_i = \dfrac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}_i)}{q_{\boldsymbol{\phi}}(\mathbf{z}_i|\mathbf{x})}$. These variables are also iid and if Y_i is bounded for all i, then its expectation is finite and in our case $\mathbb{E}_{\mathbf{z}_1,...,\mathbf{z}_k \sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z}_i)}{q_{\boldsymbol{\phi}}(\mathbf{z}_i|\mathbf{x})}\right] = p_{\boldsymbol{\theta}}(\mathbf{x})$. Consider the random variable $M_k = \frac{1}{k}\sum_{i=1}^{k} \frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z}_i)}{q_{\boldsymbol{\phi}}(\mathbf{z}_i|\mathbf{x})}$. It follows from the Strong Law of Large Numbers that the average converges to the expected value with probability one: $P(M_k \rightarrow p_{\boldsymbol{\theta}}(\mathbf{x})) = 1$. Since the logarithm is a continuous function, $\log M_k \rightarrow \log p_{\boldsymbol{\theta}}(\mathbf{x})$ almost surely. This implies convergence in distribution, hence $\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log M_k\right] \rightarrow \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x})\right] = \log p_{\boldsymbol{\theta}}(\mathbf{x})$.

□

It directly follows that using $k > 1$ posterior samples gives us a tighter lower bound since the IWAE formulation contains the standard ELBO as a special case. Drawing $k = 1$ sample simplifies back to the VAE ELBO:

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) \geq \mathcal{L}_{IWAE}^{k+1}(\mathbf{x}) \geq ... \geq \mathcal{L}_{IWAE}^{1}(\mathbf{x}) = \mathcal{L}_{VAE}(\mathbf{x})$$

It was also observed in the original work of Burda et al. that the strictly tighter lower bound results in improved test likelihood estimation and consequently a generative performance of higher quality. In addition, drawing multiple samples provides more flexibility to the network hence allows it to capture the posterior more accurately. The fact that the model can explore more complex distributions to approximate the true posterior implies that the factorial assumption is reduced and more dependencies in the latent space can be captured.

Examining Equation 2.1, it is noticeable that in contrast to the standard ELBO, in the IWAE ELBO we can not formulate a KL-divergence term due to the fact that the averaging is performed inside the logarithm. This would imply updates with higher variances, but according to the authors in their experiments there was no significant difference between the standard VAE updates and the IWAE updates with $k = 1$.

The difference between the standard VAE and IWAE methods can be understood more deeply if one takes a closer look at the gradients of the networks. The differentiation is performed for the generative parameters $\boldsymbol{\theta}$ and variational parameters $\boldsymbol{\phi}$, in the latter case the reparametrization trick can be applied, similar to the standard VAE gradient derivation.

The gradients of IWAE ELBO with respect to $\boldsymbol{\theta}$ can be obtained in a straightforward manner by employing the chain rule and the linearity of the operator $\nabla_{\boldsymbol{\theta}}$:

$$
\begin{aligned}
\nabla_{\boldsymbol{\theta}}\mathcal{L}^{k}_{IWAE}(\mathbf{x}) &= \nabla_{\boldsymbol{\theta}}\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log\frac{1}{k}\sum_{i=1}^{k} w_i(\mathbf{x},\mathbf{z}_i,\boldsymbol{\theta},\boldsymbol{\phi})\right] \\
&= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\nabla_{\boldsymbol{\theta}}\log\frac{1}{k}\sum_{i=1}^{k} w_i(\mathbf{x},\mathbf{z}_i,\boldsymbol{\theta},\boldsymbol{\phi})\right] \\
&= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\nabla_{\boldsymbol{\theta}}\log\frac{1}{k}+\nabla_{\boldsymbol{\theta}}\log\sum_{i=1}^{k} w_i(\mathbf{x},\mathbf{z}_i,\boldsymbol{\theta},\boldsymbol{\phi})\right] \\
&= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\frac{1}{\sum_{i=1}^{k} w_i(\mathbf{x},\mathbf{z}_i,\boldsymbol{\theta},\boldsymbol{\phi})}\cdot\sum_{i=1}^{k}\nabla_{\boldsymbol{\theta}}w_i(\mathbf{x},\mathbf{z}_i,\boldsymbol{\theta},\boldsymbol{\phi})\right] \\
&= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\sum_{i=1}^{k}\frac{1}{\sum_{i=1}^{k} w_i(\mathbf{x},\mathbf{z}_i,\boldsymbol{\theta},\boldsymbol{\phi})}\nabla_{\boldsymbol{\theta}}w_i(\mathbf{x},\mathbf{z}_i,\boldsymbol{\theta},\boldsymbol{\phi})\right] \\
&= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\sum_{i=1}^{k}\frac{w_i}{\sum_{i=1}^{k} w_i(\mathbf{x},\mathbf{z}_i,\boldsymbol{\theta},\boldsymbol{\phi})}\nabla_{\boldsymbol{\theta}}\log w_i(\mathbf{x},\mathbf{z}_i,\boldsymbol{\theta},\boldsymbol{\phi})\right] \\
&= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\sum_{i=1}^{k}\tilde{w}_i\nabla_{\boldsymbol{\theta}}\log w_i(\mathbf{x},\mathbf{z}_i,\boldsymbol{\theta},\boldsymbol{\phi})\right]
\end{aligned}
\tag{2.3}
$$

where in order to arrive to the form similar to the gradient of the VAE ELBO, we substituted $\nabla_{\boldsymbol{\theta}}w_i(\mathbf{x},\mathbf{z}_i,\boldsymbol{\theta},\boldsymbol{\phi})$ with the expression $w_i\cdot\nabla_{\boldsymbol{\theta}}\log w_i(\mathbf{x},\mathbf{z}_i,\boldsymbol{\theta},\boldsymbol{\phi})$. This equality comes from the rearrangement of $\nabla_{\boldsymbol{\theta}}\log w_i(\mathbf{x},\mathbf{z}_i,\boldsymbol{\theta},\boldsymbol{\phi})=\frac{1}{w_i}\cdot\nabla_{\boldsymbol{\theta}}w_i$. The terms $\tilde{w}_i=\frac{w_i}{\sum_{i=1}^{k}w_i}$ are the normalized importance weights.

The gradients with respect to $\boldsymbol{\phi}$ are calculated by using the reparametrization trick. The steps of the above derivations are similar, after we introduced the random variable $\boldsymbol{\epsilon}\sim p(\boldsymbol{\epsilon})$ and expressed $\mathbf{z}_i$ as a deterministic function of $\boldsymbol{\epsilon}_i$. Usually, $\boldsymbol{\epsilon}_1,...,\boldsymbol{\epsilon}_k$ are sampled from a standard Gaussian distribution.

$$
\begin{aligned}
\nabla_{\boldsymbol{\phi}}\mathcal{L}^{k}_{IWAE}(\mathbf{x}) &= \nabla_{\boldsymbol{\phi}}\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log\frac{1}{k}\sum_{i=1}^{k} w_i(\mathbf{x},\mathbf{z}_i,\boldsymbol{\theta},\boldsymbol{\phi})\right] \\
&= \nabla_{\boldsymbol{\phi}}\mathbb{E}_{p(\boldsymbol{\epsilon})}\left[\log\frac{1}{k}\sum_{i=1}^{k} w_i(\mathbf{x},g(\boldsymbol{\epsilon},\boldsymbol{\phi},\mathbf{x}),\boldsymbol{\theta})\right] \\
&= \mathbb{E}_{p(\boldsymbol{\epsilon})}\left[\nabla_{\boldsymbol{\phi}}\log\frac{1}{k}\sum_{i=1}^{k} w_i(\mathbf{x},g(\boldsymbol{\epsilon},\boldsymbol{\phi},\mathbf{x}),\boldsymbol{\theta})\right] \\
&= \mathbb{E}_{p(\boldsymbol{\epsilon})}\left[\sum_{i=1}^{k}\tilde{w}_i\nabla_{\boldsymbol{\phi}}\log w_i(\mathbf{x},g(\boldsymbol{\epsilon},\boldsymbol{\phi},\mathbf{x}),\boldsymbol{\theta})\right]
\end{aligned}
\tag{2.4}
$$

Analogously to the VAE gradient calculation, the expectation is approximated with Monte Carlo estimation:

$$
\nabla_{\theta,\phi}\mathcal{L}^{k}_{IWAE}(\mathbf{x}) = \sum_{i=1}^{k}\tilde{w}_i\nabla_{\theta,\phi}\log w_i(\mathbf{x},g(\boldsymbol{\epsilon},\boldsymbol{\phi},\mathbf{x}),\boldsymbol{\theta})
$$

For comparison, recall that the Monte Carlo estimate of the gradient of the standard VAE objective is in the form:

$$\nabla_{\theta,\phi}\mathcal{L}_{VAE}(\mathbf{x}) = \mathbb{E}_{p(\boldsymbol{\epsilon})}\left[\nabla_{\theta,\phi}\log w_i(\mathbf{x}, g(\boldsymbol{\epsilon}, \boldsymbol{\phi}, \mathbf{x}), \boldsymbol{\theta})\right] = \sum_{i=1}^{k}\frac{1}{k}\nabla_{\theta,\phi}\log w_i(\mathbf{x}, g(\boldsymbol{\epsilon}, \boldsymbol{\phi}, \mathbf{x}), \boldsymbol{\theta})$$

One can see that while in the VAE model the samples are equally weighted, the IWAE utilizes weights proportional to the importance weights.

2.2 Reinterpretation of IWAE

A related work from Cremer et al. [7] provides another essential perspective of the IWAE model which is crucial for understanding what is really happening in the IWAE framework. It states that despite the general interpretation of IWAE ELBO such that it comes up with a tighter lower bound, one should look at it as the standard VAE ELBO which utilizes a more complex variational distribution. The discussion in this part is based on the works of Cremer [7], [6].

The IWAE ELBO incorporates importance weighting, giving each sample a relative importance weight during the loss calculation. This method alters the sampling procedure and modifies the distribution we are actually sampling from given the importance weights. Cremer et al. [7] formulates the implicit distribution which the IWAE model uses to approximate the true posterior. The unnormalized implicit variational posterior q_{IW} is defined as

$$q_{IW}(\mathbf{z}|\mathbf{x}, \mathbf{z}_2, .., \mathbf{z}_k) = \frac{\frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}}{\frac{1}{k}\sum_{j=1}^{k}\frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}_j)}{q_{\boldsymbol{\phi}}(\mathbf{z}_j|\mathbf{x})}} q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) = \frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})}{\frac{1}{k}\left(\frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})} + \sum_{j=2}^{k}\frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}_j)}{q_{\boldsymbol{\phi}}(\mathbf{z}_j|\mathbf{x})}\right)} \tag{2.5}$$

where $\mathbf{z}, \mathbf{z}_2, ..., \mathbf{z}_k \sim q(\mathbf{z}|\mathbf{x})$ and $\mathbf{z}$ could be denoted as $\mathbf{z}_1$ as well but for notational purposes it was left as $\mathbf{z}$ since it is the latent point where the distribution is evaluated in. The distribution depends on the drawn samples, as it is indicated in the conditioning.

One can easily see that if $k = 1$ then $q_{IW}(\mathbf{z}|\mathbf{x}, \mathbf{z}_2, .., \mathbf{z}_k)$ is equivalent with $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$. A major observation regarding the proposed distribution is that plugging it in the VAE ELBO instead of $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$, in expectation it is equivalent with the IWAE ELBO taking its samples from $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$. We will not delve into the details of the proof, which can be found in the original paper, but note that it serves as a base for the proof in a later chapter for the model TD-IWAE.

Theorem 2.2.1. $\mathbb{E}_{z_2,...,z_k\sim q_{\phi}(z|x)}\left[\mathcal{L}_{VAE}[q_{IW}]\right] = \mathcal{L}_{IWAE}[q_{\boldsymbol{\phi}}]$

Furthermore, another important property of q_{IW} is that it converges to the true posterior $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$ as $k \to \infty$.

Theorem 2.2.2. *$q_{IW}(z|\boldsymbol{x}, z_2, .., z_k) \to p_{\boldsymbol{\theta}}(z|\boldsymbol{x})$ as $k \to \infty$.*

Proof. To see this, lets rewrite q_{IW} in order to reveal the true posterior in the expression:

$$q_{IW}(\mathbf{z}|\mathbf{x}, \mathbf{z}_2, .., \mathbf{z}_k) = \frac{p_\theta(\mathbf{x}, \mathbf{z})}{\frac{1}{k}\left(\sum_{j=1}^{k} \frac{p_\theta(\mathbf{x}, \mathbf{z}_j)}{q_\phi(\mathbf{z}_j|\mathbf{x})}\right)} = \frac{p_\theta(\mathbf{x})}{\frac{1}{k}\left(\sum_{j=1}^{k} \frac{p_\theta(\mathbf{x}, \mathbf{z}_j)}{q_\phi(\mathbf{z}_j|\mathbf{x})}\right)} p_\theta(\mathbf{z}|\mathbf{x})$$

As it was already mentioned, the average of the importance weights are an unbiased estimator of $p_\theta(\mathbf{x})$ and their expected value can be approximated such as:

$$p_\theta(\mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] \approx \frac{1}{k}\sum_{j=1}^{k} \frac{p_\theta(\mathbf{x}, \mathbf{z}_j)}{q_\phi(\mathbf{z}_j|\mathbf{x})}$$

If $q_\phi(\mathbf{z}|\mathbf{x})$ is non-zero for all $\mathbf{x}$ where $p_\theta(\mathbf{z}|\mathbf{x})$ is non-zero, the Strong Law of Large Numbers implies that the average approaches $p_\theta(\mathbf{x})$ as $k \to \infty$ with probability 1. Since it is the denominator in the previous expression, then $q_{IW}(\mathbf{z}|\mathbf{x}, \mathbf{z}_2, .., \mathbf{z}_k)$ converges to $p_\theta(\mathbf{z}|\mathbf{x})$ with probability 1. □

The proposed q_{IW} distribution is unnormalized, but Cremer also introduces a normalized distribution, q_{EW}, which is defined as the expectation of q_{IW} over the samples $\mathbf{z}_2, ..., \mathbf{z}_k$. A proof for showing that it is normalized is given in the works of Cremer et al. [7], [6].

$$q_{EW}(\mathbf{z}|\mathbf{x}) = \mathbb{E}_{\mathbf{z}_2,...,\mathbf{z}_k}\left[q_{IW}(\mathbf{z}|\mathbf{x}, \mathbf{z}_2, .., \mathbf{z}_k)\right] \tag{2.6}$$

The new distribution q_{EW}, in contrast to q_{IW}, is not dependent on the samples $\mathbf{z}_2, ..., \mathbf{z}_k$ since the expectation can be seen as a marginalization over these variables. A question intuitively arises regarding q_{EW}: If the VAE ELBO with q_{IW} equals to the IWAE ELBO in expectation, then what happens if q_{EW} is used in the vanilla ELBO? Calculating q_{EW} is intractable [6], but according to the authors, it would yield an upper bound for the IWAE ELBO: $\mathcal{L}_{VAE}[q_{EW}] \geq \mathcal{L}_{IWAE}[q]$. For proof, refer to [7].

As the IWAE is trained in a way that the objective weights the samples drawn from the variational posterior, one have to take care of performing the sampling procedure during model evaluation. Resampling is required during prediction in order to correctly sample from the learnt implicit distribution. The appropriate algorithm doing this is equivalent with the Sampling Importance Resampling method already discussed in this work. The algorithm is as follows (taken from Cremer et al. [7]):

2.3 IWAE ELBO and Importance Sampling

The connection between the IWAE and the importance sampling Monte Carlo method can be acknowledged by the work from Bachman and Precup [1]. It states that the IWAE ELBO is equivalent to variational inference where the variational posterior is adjusted towards the true posterior using normalized importance sampling.

The scenario in context of the VAE is that it would be desirable to draw samples from the true posterior $p_\theta(\mathbf{z}|\mathbf{x})$, but due to its intractability, it is not applicable. Recall that in Sampling-Importance-Resampling a proposal distribution $q_\phi(\mathbf{z}|\mathbf{x})$ is used as the sampling distribution

Algorithm 1 Sampling $q_{EW}(\mathbf{z}|\mathbf{x})$

```
Input: $k$: number of samples drawn
for $i = 1, ..., k$ do
    Sample $\mathbf{z}_i \sim q(\mathbf{z}|\mathbf{x})$
    Calculate $w_i = \frac{p(\mathbf{x},\mathbf{z}_i)}{q(\mathbf{z}_i|\mathbf{x})}$
end for
Normalization: $\tilde{w}_i = \frac{w_i}{\sum_{i=1}^{k} w_i}$ for $\forall i$
Resample according to the obtained normalized weights $\tilde{\mathbf{w}}_i$: $j \sim Categorical(\tilde{\mathbf{w}}_i)$
Return: $\mathbf{z}_j$
```

instead of the underlying complex distribution, then the normalized importance weights are used to perform a second sampling. In the IWAE ELBO we would like to apply exactly this technique. In this case the importance weights are $w_i = \dfrac{p_\theta(\mathbf{z}_i|\mathbf{x})}{q_\phi(\mathbf{z}_i|\mathbf{x})}$. Their normalized form is

$$\hat{w}_i = \frac{\frac{p_\theta(\mathbf{z}_i|\mathbf{x})}{q_\phi(\mathbf{z}_i|\mathbf{x})}}{\sum_{j=1}^{k} \frac{p_\theta(\mathbf{z}_j|\mathbf{x})}{q_\phi(\mathbf{z}_j|\mathbf{x})}} = \frac{\frac{p_\theta(\mathbf{x},\mathbf{z}_i)}{q_\phi(\mathbf{z}_i|\mathbf{x})}}{\sum_{j=1}^{k} \frac{p_\theta(\mathbf{x},\mathbf{z}_j)}{q_\phi(\mathbf{z}_j|\mathbf{x})}} = \frac{\frac{p_\theta(\mathbf{x}|\mathbf{z}_i)p_\theta(\mathbf{z}_i)}{q_\phi(\mathbf{z}_i|\mathbf{x})}}{\sum_{j=1}^{k} \frac{p_\theta(\mathbf{x}|\mathbf{z}_j)p_\theta(\mathbf{z}_j)}{q_\phi(\mathbf{z}_j|\mathbf{x})}}$$

The samples drawn according to the above weights can be considered as that they were approximately drawn from $p_\theta(\mathbf{z}|\mathbf{x})$ as $k \longrightarrow \infty$. Since during training the number of samples can be drawn is restricted by the available computational resources, therefore it is not really common to draw that many samples. Consequently, the distribution from which the samples are coming after the resampling step is still considered as an approximation of the true posterior. Let this approximate distribution be $q_k(\mathbf{z}|\mathbf{x})$. A sample obtained from $q_k(\mathbf{z}|\mathbf{x})$ is highly dependent on its associated importance weight as well. According to Bachman and Precup, the original ELBO utilizing $q_k(\mathbf{z}|\mathbf{x})$ can be expressed as

$$\log p_\theta(\mathbf{x}) \geq \mathbb{E}_{(\mathbf{z}_i,\hat{w}_i)\sim q_k(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x}|\mathbf{z}_i)p_\theta(\mathbf{z}_i)}{\hat{w}_i q_\phi(\mathbf{z}_i|\mathbf{x})}\right] \tag{2.7}$$

The importance weights $\hat{w}_i$ depend on the other $\mathbf{z}_j, j \neq i$ samples, but we can marginalize over the resampling step. This transforms the expectation to be only based on the firstly drawn k samples and to not depend on the weights anymore. The summation over all possible resampling outcome ensures that the result reflects the behavior of the resampling. The above expression then is altered as

$$
\begin{aligned}
\log p_{\boldsymbol{\theta}}(\mathbf{x}) \geq\ & \mathbb{E}_{\mathbf{z}_1,\mathbf{z}_2,..,\mathbf{z}_k \sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\sum_{i=1}^{k} \hat{w}_i \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}_i)p_{\boldsymbol{\theta}}(\mathbf{z}_i)}{\hat{w}_i q_{\boldsymbol{\phi}}(\mathbf{z}_i|\mathbf{x})}\right] \\
=\ & \mathbb{E}_{\mathbf{z}_1,\mathbf{z}_2,..,\mathbf{z}_k \sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\sum_{i=1}^{k} \hat{w}_i \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}_i)p_{\boldsymbol{\theta}}(\mathbf{z}_i)}{\left(\frac{\frac{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}_i)p_{\boldsymbol{\theta}}(\mathbf{z}_i)}{q_{\boldsymbol{\phi}}(\mathbf{z}_i|\mathbf{x})}}{\sum_{j=1}^{k}\frac{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}_j)p_{\boldsymbol{\theta}}(\mathbf{z}_j)}{q_{\boldsymbol{\phi}}(\mathbf{z}_j|\mathbf{x})}}\right) q_{\boldsymbol{\phi}}(\mathbf{z}_i|\mathbf{x})}\right] \\
=\ & \mathbb{E}_{\mathbf{z}_1,\mathbf{z}_2,..,\mathbf{z}_k \sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\sum_{i=1}^{k} \hat{w}_i \log \sum_{j=1}^{k} \frac{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}_j)p_{\boldsymbol{\theta}}(\mathbf{z}_j)}{q_{\boldsymbol{\phi}}(\mathbf{z}_j|\mathbf{x})}\right] \\
=\ & \mathbb{E}_{\mathbf{z}_1,\mathbf{z}_2,..,\mathbf{z}_k \sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log \sum_{j=1}^{k} \frac{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}_j)p_{\boldsymbol{\theta}}(\mathbf{z}_j)}{q_{\boldsymbol{\phi}}(\mathbf{z}_j|\mathbf{x})}\right] \\
=\ & \mathcal{L}_{IWAE}^{k}(\mathbf{x})
\end{aligned}
$$

using that $\sum_{i=1}^{k} \hat{w}_i = 1$. The last term, as indicated, is exactly the ELBO used in the IWAE model.

One can observe that in the second line the expression of the denominator equals to the q_{IW} distribution proposed by Cremer et al., except for the multiplier k:

$$
\left(\frac{\frac{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})p_{\boldsymbol{\theta}}(\mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}}{\sum_{j=1}^{k}\frac{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}_j)p_{\boldsymbol{\theta}}(\mathbf{z}_j)}{q_{\boldsymbol{\phi}}(\mathbf{z}_j|\mathbf{x})}}\right) \cdot q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) = \frac{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})p_{\boldsymbol{\theta}}(\mathbf{z})}{\sum_{j=1}^{k}\frac{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}_j)p_{\boldsymbol{\theta}}(\mathbf{z}_j)}{q_{\boldsymbol{\phi}}(\mathbf{z}_j|\mathbf{x})}} = \frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{\frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})} + \sum_{j=2}^{k}\frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z}_j)}{q_{\boldsymbol{\phi}}(\mathbf{z}_j|\mathbf{x})}}
$$

Chapter 3

TDVAE

3.1 Insights into hierarchical models

Hierarchical VAE models, a generalized form of vanilla VAEs, have become the focus of many research studies in the last few years. Their popularity can be explained firstly by the fact that they can be used to model hierarchical dependecies or concepts, secondly that they can provide increased expressiveness of the variational posterior and prior distributions. In a traditional VAE architecture one stochastic layer is used which in itself is not always able to learn complex data distributions as owning a simple form of the posterior distribution. Increased capacity is achieved by utilizing several stochastic layers in the model. Stacking multiple latent variables can relax the limitations of learning more complex posterior representations, and enables the network to capture a richer posterior distribution [5], [31], [9]. Several outstanding hierarchical generative models were published in recent years, such as NVAE [32], VDVAE [5], Ladder VAE [31], which was further improved with the model BIVA [23]. A brief outline of hierarchical VAEs is introduced here, mostly based on [9].

This work will focus on hierarchical VAE models with two stochastic layers hence their theoretical background will be introduced in this fashion as well. In addition, the models considered here possess a Markovian structure, meaning that the generative process forms a Markov chain. In this sense the latent variable $\mathbf{z}_i$ in the hierarchy depends only on the previous $\mathbf{z}_{i+1}$, and is independent of all the previous latents [21].

Considering the two latent variables $\mathbf{z}_1$, and $\mathbf{z}_2$, the joint distribution that the generative model learns is $p_\theta(\mathbf{x}, \mathbf{z}_1, \mathbf{z}_2)$. This can be factorized as:

$$p_\theta(\mathbf{x}, \mathbf{z}_1, \mathbf{z}_2) = p_\theta(\mathbf{x}|\mathbf{z}_1, \mathbf{z}_2) \cdot p_\theta(\mathbf{z}_1|\mathbf{z}_2) \cdot p_\theta(\mathbf{z}_2) = p_\theta(\mathbf{x}|\mathbf{z}_1) \cdot p_\theta(\mathbf{z}_1|\mathbf{z}_2) \cdot p_\theta(\mathbf{z}_2)$$

Since the true posterior $p_\theta(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x})$ is intractable, the hierarchical model utilizes the variational posterior, which is in the form $q_\phi(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x})$.

The factorization of this posterior defines how the recognition model is structured and there are more options for this decomposition. The first approach is referred to as the bottom-up (or chain) manner where the latents are in the opposite arrangement compared to their order in the generative model:

$$q_\phi(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x}) = q_\phi(\mathbf{z}_2|\mathbf{x}, \mathbf{z}_1) \cdot q_\phi(\mathbf{z}_1|\mathbf{x}) = q_\phi(\mathbf{z}_2|\mathbf{z}_1) \cdot q_\phi(\mathbf{z}_1|\mathbf{x})$$

While the second approach is called the top-down factorization, firstly introduced by [31]:

$$q_{\boldsymbol{\phi}}(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x}) = q_{\boldsymbol{\phi}}(\mathbf{z}_1|\mathbf{x}, \mathbf{z}_2) \cdot q_{\boldsymbol{\phi}}(\mathbf{z}_2|\mathbf{x})$$

Sampling from the joint variational posterior when having a hierarchy of latent variables can also be viewed as performing ancestral sampling. In order to sample from $q_{\boldsymbol{\phi}}(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x})$ one needs to get a sample obtained from $q_{\boldsymbol{\phi}}(\mathbf{z}_2|\mathbf{x})$ and then use this $\mathbf{z}_2$ from drawing a sample from $q_{\boldsymbol{\phi}}(\mathbf{z}_1|\mathbf{x}, \mathbf{z}_2)$. In other words, for inference we need to sample from the distribution conditioned on the variable's parent variables [18].

In this thesis we will concentrate on hierarchical VAE with the top-down recognition model with the latent variables $\mathbf{z}_1, \mathbf{z}_2$. The ELBO for the two-level top-down VAE can be derived similarly to the one-layer case. Firstly, we have to rewrite the expression for $\log p_{\boldsymbol{\theta}}(\mathbf{x})$:

$$\begin{aligned}
\log p_{\boldsymbol{\theta}}(\mathbf{x}) &= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x})\right] \\
&= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})}\left[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}_1, \mathbf{z}_2)}{p_{\boldsymbol{\theta}}(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x})}\right] \\
&= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})}\left[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}_1, \mathbf{z}_2)}{p_{\boldsymbol{\theta}}(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x})} \cdot \frac{q_{\boldsymbol{\phi}}(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x})}{q_{\boldsymbol{\phi}}(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x})}\right] \\
&= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})}\left[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}_1, \mathbf{z}_2)}{q_{\boldsymbol{\phi}}(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x})} \cdot \frac{q_{\boldsymbol{\phi}}(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x})}{p_{\boldsymbol{\theta}}(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x})}\right] \\
&= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})}\left[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}_1) \cdot p_{\boldsymbol{\theta}}(\mathbf{z}_1|\mathbf{z}_2) \cdot p_{\boldsymbol{\theta}}(\mathbf{z}_2)}{q_{\boldsymbol{\phi}}(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x})} \cdot \frac{q_{\boldsymbol{\phi}}(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x})}{p_{\boldsymbol{\theta}}(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x})}\right] \\
&= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})}\left[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}_1) \cdot p_{\boldsymbol{\theta}}(\mathbf{z}_1|\mathbf{z}_2) \cdot p_{\boldsymbol{\theta}}(\mathbf{z}_2)}{q_{\boldsymbol{\phi}}(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x})}\right] + \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})}\left[\log \frac{q_{\boldsymbol{\phi}}(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x})}{p_{\boldsymbol{\theta}}(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x})}\right] \\
&= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})}\left[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}_1) \cdot p_{\boldsymbol{\theta}}(\mathbf{z}_1|\mathbf{z}_2) \cdot p_{\boldsymbol{\theta}}(\mathbf{z}_2)}{q_{\boldsymbol{\phi}}(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x})}\right] + KL[q_{\boldsymbol{\phi}}(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x})||p_{\boldsymbol{\theta}}(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x})]
\end{aligned}$$

where the second term is the Kullback-Leibler divergence between the variational and true posterior distributions. Since this latter term is non-negative, we can obtain a lower bound on $\log p_{\boldsymbol{\theta}}(\mathbf{x})$:

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) \geq \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})}\left[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}_1) \cdot p_{\boldsymbol{\theta}}(\mathbf{z}_1|\mathbf{z}_2) \cdot p_{\boldsymbol{\theta}}(\mathbf{z}_2)}{q_{\boldsymbol{\phi}}(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x})}\right]$$

Generally, the approximate distributions and priors follow a Gaussian distribution with diagonal covariance. In contrast, in the present work we chose a Gaussian for parameterizing the latent variable $\mathbf{z}_2$ and Laplace distribution for the latent $\mathbf{z}_1$. Note that the parameters of the conditional distributions of latents are determined by neural networks, while the unconditional $p_{\boldsymbol{\theta}}(\mathbf{z}_2)$ is set to be a factorized standard Gaussian. For $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}_1)$ we stuck to use a Gaussian as it is usual in the literature.

To offer a comprehensive overview of hierarchical models, we should also mention the drawbacks of these networks. Training with many layers of stochastic variables could be quite challenging due to their increased complexity which requires larger memory and computational power, longer training times and it also means higher sensitivity caused by the propagation in deeper networks. Furthermore, posterior collapse also poses difficulties for the training methodology.

3.1.1 Posterior collapse

The phenomenon of posterior collapse was recognized by several earlier studies and it is a well-known problem of the optimization procedure of training VAEs [31], [10]. Posterior collapse refers to the event when the variational posterior is getting closer to the prior, in terms of their KL-divergence. To put it another way, some dimensions of the learnt posterior could become inactive. Consequently, these dimensions of the latent space does not contain much meaningful information about x. In hierarchical models, it predominantly affects the upper stochastic layers.

The collapse is mainly caused by the KL-term in the ELBO [31], since this regularization term could be too strict to let the posterior to be learnt during training. Hence, the collapsed units can also be identified by the relatively small KL-divergence value corresponding the given unit. The generative performance of the model can also assist this phenomenon, since a network that cannot capture the key features of the data, is also not able to produce good quality reconstructed images.

A well-proven solution for preventing the collapse of the latent space is utilizing β-annealing [31]. Essentially, this trick modifies the importance of the different terms in the ELBO during training by introducing a weighting factor β to the KL-divergence term. At the beginning, β is set to a relatively small value, for instance to 0.1 and then it is progressively increased over some epochs until reaches 1, which case is identical to the original ELBO. Smaller β over the epochs early in training means less weight to the regularization, allowing the model to learn and incorporate some useful information into the latent space. Then, it is less likely to vanish in the later stages of the training when β is increased and when the KL-term is in full capacity with $\beta = 1$ [31].

3.2 TDVAE

The present thesis is based on the TDVAE model taken from Csikor et al. [9], [8], which is a hierarchical VAE featuring two stochastic latent layers and a top-down recognition network. In this section we will review this network, detail its structure, specific architectural choices and outline the top-down, hierarchical ELBO used for its training procedure.

The works from Csikor et al. focuses on inspecting a model of the early visual cortex with two-latent layer hierarchical VAEs. The centre of attention of the examinations are the latent representations learnt by these models on natural images. Among the investigated architectures, the TDVAE and its learnt latent variables was discussed in more details in the paper [8]. The results were compared to the cortical responses found in macaques as well. The structure of the model is inspired from neuroscience to match the anatomy and conditions of the mammalian vision.

3.2.1 The connection between the visual cortex and TDVAE

The task of the visual system is to process and extract useful information from the incoming visual signals. The first step in the perception of such visual information happens in the visual cortex, where the neuronal computations are organized into a hierarchy [16].

The first layer of this cortical hierarchy is the primary visual cortex also known as V1, which accounts for capturing local orientation of edges and lines, and spatial scale. [11], [16]. It is also the most studied area of the visual pathway [16].

The second visual area V2 obtains information from V1, and extracts more complex features from the visual image by incorporating the characteristics provided by V1. This area is sensitive to changes in color, spatial frequency, and patterns [16], and it was also shown that it is highly selective to natural image texture statistics [35], [11]. The stronger responses for texture stimuli were investigated in macaques, and was found in V2 but not in V1. V2 also has connection with the higher parts of the hierarchy which consists of the areas V3, V4, and V5.

In neuroscience, a neuron's receptive field refers to the set of neurons from which it receives its input [20]. In other words, it is a part of the stimulus which effects the activity of a neuron. In the context of neural networks, in the field of image processing, the term can also be used for the region in the input image which determines the output of a single neuron in the first layer [22]. In deeper layers, it is referred to the set of neurons to which it is connected to in the previous layer. A neuron's projective field denotes the collection of neurons to which it projects its output [19].

The processes in the primary visual cortex could be modelled by a set of Gabor-filters. The Gabor-filter is a linear filter which is a result of modulating a sinusoidal function with a Gaussian function [15]. The filters are widely used in image processing, since they could be used effectively for capturing the frequencies and orientations in the image. The response of a cell to an image is given by a convolution operation of the Gabor-filter with the image [27]. A filter will give high response to some particular patterns similar to neurons in the primary visual cortex which are selective to certain orientations and spatial frequencies.

It has been observed that there are receptive fields in the mammalian primary visual cortex which are spatially localized, oriented and bandpass [25]. The term bandpass stands for filters which possess a specific frequency range, and only spatial frequencies falling into this range are allowed to pass through. For modelling the primary visual cortex it is desired to obtain a learning algorithm trained on natural images with comparable response properties. Olshausen and Field [25] designed a learning algorithm which produces Gabor-like filters with the above properties resembling to the receptive fields found in mammalians primary visual cortex. They argue that the developed properties of the model's receptive fields emerged from optimizing a cost function which accounts not only for reconstruction quality but also for sparseness at the same time.

Processes in the primary visual cortex are working in a sparse way, meaning that only a relatively small set of neurons are firing to specific stimuli. The biological vision utilized this type of encoding because natural images also possess a sparse structure, meaning that compared to their size, they do not contain much meaningful information. Mostly there are repeated patterns with low variability, introducing redundancy [26]. The sparse encoding of the natural scenes enables an efficient representation for the following layers of the hierarchy [25].

Intuitively follows that when modelling the visual cortex with a learning algorithm, we are looking for a sparse coding of the natural images shown to the algorithm, where the image is composed only of a small set of basis functions. As images can be described as a linear combination of basis functions, it means we prefer that only few coefficients are not zero in the combination. Olshausen and Field [25] utilized this property in their work, and concluded that the algorithm which optimized for generating sparse codes for the natural stimuli, indeed produces localized, bandpass and receptive fields. The Gabor-filters developed with these three properties correspond to the basis functions from which the images are constructed as a linear combination. The coefficients of the linear combination form the image code. It is argued that the optimization objective used in [25] allows not only for sparsity, but also for overcompleteness for the coefficients in this linear combination. Utilizing more basis functions than the image dimension introduce higher flexibility and prevention from information loss in the representation as there is no restriction for the number of basis functions [29].

The model developed by Olshausen et al. is a linear generative model, and served as an inspiration for other researchers as well. In the work from Geadah et al. [12] a one latent-layer variational autoencoder, the SVAE was introduced, possessing architectural elements inspired from Olshausen et al. The model integrated the key properties of the sparse coding model in the VAE framework, which consisted of 3 architectural changes: applying a single linear layer as the generative model instead of a complex network, using overcomplete latent representation meaning that there are more latent dimensions than image pixels, and employing a sparse distribution as the prior for the latent variables instead of a Gaussian. The latter grants that the neurons of latent vectors have a sparse activity distribution. The authors showed that the model could represent the early visual cortical response properties of mammals.

The model TDVAE is a Variational Autoencoder with a hierarchical structure having two stochastic layers. Its design are summarized in the papers from Csikor et al. [9], [8], and here we will also overview the architectural specialties based on these works.

As intuition dictates, these first and second stochastic layers forming the latent space for $\mathbf{z}_1$ and $\mathbf{z}_2$ can be corresponded with the primary and secondary visual cortices, V1 and V2, respectively. The recognition network can be seen as the model of visual cortical processes, hence its neuron's activity as cortical neuronal activities [8].

The architecture based on the general form of VAEs, but several choices motivated by neuroscience were made, in order to bring it closer to the composition of the early visual pathway.

First of all, building on previous work in using generative methods to model the visual cortices of mammalians, the prior distribution for the latent $\mathbf{z}_1$ was chosen to be Laplacian, encouraging sparsity in the $\mathbf{z}_1$ representation. In addition, in the generative model the relationship of $\mathbf{z}_1$ and the observations is constrained to be linear, hence the MLP which projects $\mathbf{z}_1$ to $\hat{\mathbf{x}}$ is a simple linear layer: $\mathbf{W}\mathbf{z}_1 + \mathbf{b} = \hat{\mathbf{x}}$. Also, the latent dimension for $\mathbf{z}_1$ was chosen to be overcomplete. As the input images have size 20×20, meaning 400 input dimensions, the latent vector $\mathbf{z}_1$ possesses 450 dimensions.

Another key characteristic of the visual pathway is that along with the feedforward processes performed upwardly in the hierarchy that act for more and more complex representations, top-down connections can also be found [13]. These are feedback paths which transfer higher-order information to lower layers in every stage of the hierarchy. In other words, it provides contextual information for the first layers from upper stages, shaping the representation learnt there. The structure of the cortical hierarchy gives motivation to formulate the recognition network using the top-down manner, and consequently applying the top-down type factorization of the variational posterior. The architecture of the recognition network firstly processes the input stimulus $\mathbf{x}$, creates a representation for it, and then this representation is used for not only to calculate the parameters of the variational posterior for $\mathbf{z}_2$, but also for computing the parameters of the posterior of $\mathbf{z}_1$. The shared layer ensures that the signal goes through the layer V1 before entering V2. The connections coming from V2 forms a contextual prior for the level V1, which we can express in the context of generative modelling as a conditional distribution of the latent $\mathbf{z}_1$ given $\mathbf{z}_2$. Another MLP in the recognition network is constructed in order to merge the signals coming from the stimulus and from V2 for the purpose of implementing the feedback pathway.

Lastly, in order to mimic the environment to which the visual system accomodated itself to, and to let the representations be developed by characteristics of natural stimuli, the dataset used for training was composed of patches of natural images. For performing inference and investigating the learnt properties, especially the texture selectivity, a dataset consisting of patches extracted from texture images was used.

All things considered, the specific assumptions of the architecture are essential inductive biases which enables the model to learn representations reproducing the principal properties of the first layers of the cortical areas.

3.2.2 The architecture of TDVAE

Similar to other VAE models, TDVAE is composed of a recognition and a generative network which own a hierarchical structure, more specifically it utilizes two stochastic latent layers, $\mathbf{z}_1$ and $\mathbf{z}_2$. As it was already discussed in the previous chapters, in this case the we seek to learn the joint distribution $p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}_1, \mathbf{z}_2)$ and owing to the fact that TDVAE features a top-down recognition network, the joint variational posterior is factorized as

$$q_\phi(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x}) = q_\phi(\mathbf{z}_1|\mathbf{x}, \mathbf{z}_2) \cdot q_\phi(\mathbf{z}_2|\mathbf{x}) \tag{3.1}$$

The major question naturally emerges: how does the ELBO look like in this scenario? Recall that for a two-latent layer hierarchical VAE the ELBO is derived as a lower bound on the data log-likelihood. This hierarchical ELBO can be further reshaped using the fact that the variational posterior has a top-down composition:

$$\begin{aligned} \log p_\theta(\mathbf{x}) &\geq \mathbb{E}_{q_\phi(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x}|\mathbf{z}_1) \cdot p_\theta(\mathbf{z}_1|\mathbf{z}_2) \cdot p_\theta(\mathbf{z}_2)}{q_\phi(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x})}\right] \\ &= \mathbb{E}_{q_\phi(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x}|\mathbf{z}_1) \cdot p_\theta(\mathbf{z}_1|\mathbf{z}_2) \cdot p_\theta(\mathbf{z}_2)}{q_\phi(\mathbf{z}_1|\mathbf{x}, \mathbf{z}_2) \cdot q_\phi(\mathbf{z}_2|\mathbf{x})}\right] \\ &= \mathcal{L}_{TD}(\mathbf{x}) \end{aligned} \tag{3.2}$$

The notation $\mathcal{L}_{TD}(\mathbf{x})$ stands for the ELBO for the two-level top-down hierarchical TDVAE. What's more, we can further transform it into an objective with a similar form as it was in the case of the one-layer VAE ELBO, by separating out the reconstruction term and KL-divergence term.

$$\mathcal{L}_{TD}(\mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x}|\mathbf{z}_1) \cdot p_\theta(\mathbf{z}_1|\mathbf{z}_2) \cdot p_\theta(\mathbf{z}_2)}{q_\phi(\mathbf{z}_1|\mathbf{x}, \mathbf{z}_2) \cdot q_\phi(\mathbf{z}_2|\mathbf{x})}\right]$$

$$= \mathbb{E}_{q_\phi(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z}_1)\right] + \mathbb{E}_{q_\phi(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{z}_1|\mathbf{z}_2)}{q_\phi(\mathbf{z}_1|\mathbf{x},\mathbf{z}_2)}\right] + \mathbb{E}_{q_\phi(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{z}_2)}{q_\phi(\mathbf{z}_2|\mathbf{x})}\right] \tag{3.3}$$

where each term can be written as:

$$\mathbb{E}_{q_\phi(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z}_1)\right] = \mathbb{E}_{q_\phi(\mathbf{z}_1|\mathbf{x},\mathbf{z}_2)\cdot q_\phi(\mathbf{z}_2|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z}_1)\right] \tag{3.4}$$

$$\mathbb{E}_{q_\phi(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{z}_1|\mathbf{z}_2)}{q_\phi(\mathbf{z}_1|\mathbf{x},\mathbf{z}_2)}\right] = \mathbb{E}_{q_\phi(\mathbf{z}_2|\mathbf{x})}\left[-KL[q_\phi(\mathbf{z}_1|\mathbf{x},\mathbf{z}_2)||p_\theta(\mathbf{z}_1|\mathbf{z}_2)]\right] \tag{3.5}$$

$$\mathbb{E}_{q_\phi(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{z}_2)}{q_\phi(\mathbf{z}_2|\mathbf{x})}\right] = -KL[q_\phi(\mathbf{z}_2|\mathbf{x})||p_\theta(\mathbf{z}_2)] \tag{3.6}$$

It is easy to observe from this form that the KL-divergence is applied layerwise, between the approximate posterior and prior of each latent. Hence the first KL term in the expression is related to $\mathbf{z}_1$ and the second one is to $\mathbf{z}_2$. In practice, these KL-terms could be too restrictive for learning the posteriors, especially in the case of the higher latent layer. In order to leave room for this purpose, one can apply β-annealing during training for both of the KL-terms.

It is essential to summarize what distributions are used to parameterize the latent variables. As it was already mentioned, this first stochastic layer, $\mathbf{z}_1$ was chosen to follow a Laplace distribution, enabling sparsity. The second layer is left to have a Gaussian distribution. Since the second stochastic layer has an influence on shaping the prior for $\mathbf{z}_1$, it is also referred as the contextual prior for $\mathbf{z}_1$.

All the conditional distributions for $\mathbf{z}_1$ and $\mathbf{z}_2$ are parameterized by individual MLPs, more precisely one network is corresponded to produce $\boldsymbol{\mu}$, and another one is to calculate $\boldsymbol{\sigma}^2$. An outline of the specific distributions used in this framework is as follows:

- $p(\mathbf{z}_2) = \mathcal{N}(\mathbf{z}_2; \mathbf{0}, \mathbf{I})$ - The prior of $\mathbf{z}_2$ is a standard Normal;
- $p(\mathbf{z}_1|\mathbf{z}_2) = \mathcal{L}(\mathbf{z}_1; \boldsymbol{\mu}(\mathbf{z}_2), \boldsymbol{b}\mathbf{I}(\mathbf{z}_2))$ - The contextual prior for $\mathbf{z}_1$ is a Laplacian conditioned on $\mathbf{z}_2$;
- $q(\mathbf{z}_2|\mathbf{x}) = \mathcal{N}(\mathbf{z}_2; \boldsymbol{\mu}(\mathbf{x}), \boldsymbol{\sigma}^2\mathbf{I}(\mathbf{x}))$ - The variational posterior for $\mathbf{z}_2$ is a Normal distribution which only depends on the observation;
- $q(\mathbf{z}_1|\mathbf{x},\mathbf{z}_2) = \mathcal{L}(\mathbf{z}_1; \boldsymbol{\mu}(\mathbf{x},\mathbf{z}_2), \boldsymbol{b}\mathbf{I}(\mathbf{x},\mathbf{z}_2))$ - The variational posterior for $\mathbf{z}_1$ is a Laplace distribution influenced by the observations as well as by the latent $\mathbf{z}_2$;
- $p(\mathbf{x}|\mathbf{z}_1) = \mathcal{N}(\hat{\mathbf{x}}; \boldsymbol{\sigma}^2\mathbf{I})$ - The likelihood is a Gaussian where the output of the generative network is interpreted as the mean of the distribution and $\boldsymbol{\sigma}$ is the standard deviation of the independent Gaussian observation noise, fixed to be 0.4.

The TDVAE architecture is composed of dense layers, and the nonlinearity used is the softplus activation function. There are no residual connections in either of the components to preserve that the dependencies are only present between consecuting variables.

The architecture of the TDVAE model is depicted in Figure 3.1 and Figure 3.2. The reconstruction network is composed of four MLP blocks and the generative part contains one MLP and one linear layer for producing the output. None of the MLP blocks are shared between the recognition and generative network. The input image $\mathbf{x}$ is fed to the first block, to MLP.a, which produces the encoding $\mathbf{l}_x$. Then, $\mathbf{l}_x$ serves as input for MLP.b which outputs the mean and standard deviation of the distribution $q_\phi(\mathbf{z}_2|\mathbf{x})$. As the parameters of the posterior is provided, a sampling is performed to obtain $\mathbf{z}_2$ which is then passed to the next block, to MLP.c. The next intermediate representation, $\mathbf{l}_z$ is produced. For merging the two signals $\mathbf{l}_x$ and $\mathbf{l}_z$ a concatenation is implemented, than the result is fed to MLP.d, which provides the parameters location and scale for the distribution $q_\phi(\mathbf{z}_1|\mathbf{x},\mathbf{z}_2)$. We need to sample from this posterior in order to get $\mathbf{z}_1$.

The generative part receives the latent $\mathbf{z}_2$ and it is directly fed to an MLP, that generates the parameters $\mu(\mathbf{z}_2)$ and $b(\mathbf{z}_2)$ of the distribution $p_\theta(\mathbf{z}_1|\mathbf{z}_2)$. Then to the sample $\mathbf{z}_1 \sim p_\theta(\mathbf{z}_1|\mathbf{z}_2)$ a dense layer is applied that generates the output $\hat{\mathbf{x}}$.

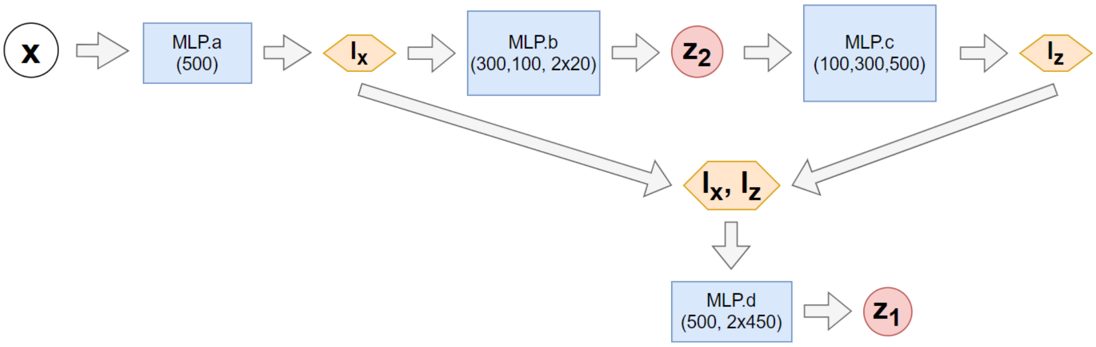


Figure 3.1: The recognition model of TDVAE

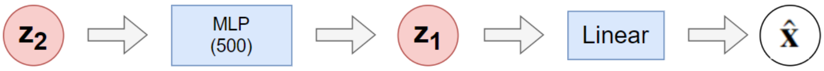


Figure 3.2: The generative model of TDVAE

Chapter 4

TD-IWAE

The goal of this thesis is to extend the hierarchical generative framework in the direction of Importance Weighted Variational Autoencoder. My goal is to derive a novel form of ELBO, implement it in the Pytorch environment and test it through experiments by training on natural images. The standard TDVAE will be used as a benchmark for performance and the evaluation of TD-IWAE will be performed in light of this model. The hope is that this new scheme will enrich the learned representations in an interpretable way. In this chapter firstly the importance weighted TD-IWAE will be introduced, then the details of how the properties of IWAE can be generalized to the top-down hierarchical TDVAE will be described along with the differences in the training methodology.

The TD-IWAE share the same architecture with the TDVAE, the main difference between them lies in the optimization objective function and in the sampling procedure. The IWAE ELBO averages the k unnormalized importance weights belonging to the corresponding samples drawn from the variational posterior. In the two-layer case we draw k samples from the distribution $q_\phi(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x})$ which is performed through ancestral sampling, meaning that for the ith sample firstly obtain $\mathbf{z}_2^i \sim q_\phi(\mathbf{z}_2|\mathbf{x})$ and then using this sample perform $\mathbf{z}_1^i \sim q_\phi(\mathbf{z}_1|\mathbf{z}_2^i, \mathbf{x})$. For the ith sample $(\mathbf{z}_1^i, \mathbf{z}_2^i)$ the associated unnormalized importance weight can be formulated in a fashion similar to the single-layer case:

$$w_i = \frac{p_\theta(\mathbf{x}, \mathbf{z}_1^i, \mathbf{z}_2^i)}{q_\phi(\mathbf{z}_1^i, \mathbf{z}_2^i|\mathbf{x})} \tag{4.1}$$

Building on the formulation of the IWAE ELBO, we can define the ELBO for the importance weighted TDVAE as

$$\begin{aligned} \mathcal{L}_{TD-IWAE}^k(\mathbf{x}) &= \mathbb{E}_{q_\phi(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})}\left[\log \frac{1}{k}\sum_{i=1}^{k} \frac{p_\theta(\mathbf{x}, \mathbf{z}_1^i, \mathbf{z}_2^i)}{q_\phi(\mathbf{z}_1^i, \mathbf{z}_2^i|\mathbf{x})}\right] \\ &= \mathbb{E}_{q_\phi(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})}\left[\log \frac{1}{k}\sum_{i=1}^{k} \frac{p_\theta(\mathbf{x}|\mathbf{z}_1^i) \cdot p_\theta(\mathbf{z}_1^i|\mathbf{z}_2^i) \cdot p_\theta(\mathbf{z}_2^i)}{q_\phi(\mathbf{z}_1^i|\mathbf{x}, \mathbf{z}_2^i) \cdot q_\phi(\mathbf{z}_2^i|\mathbf{x})}\right] \end{aligned} \tag{4.2}$$

where the samples $(\mathbf{z}_1^i, \mathbf{z}_2^i)$ are drawn from $q_\phi(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x})$. In comparison with the TDVAE ELBO, this expression cannot be transformed further to reveal KL-divergences since we cannot exchange the summation and the logarithm. In practice it has a crucial impact on the implementation and on the training instability owing to the fact that the expectation has to be approximated with Monte-Carlo sampling instead of calculating the exact KL-terms analytically.

Following the principles of VAEs, the $\mathcal{L}^k_{TD-IWAE}$ derived above can be used as an optimization objective for training the TD-IWAE as it is also a lower-bound on the data log-likelihood. The model can be trained with stochastic optimization methods such as SGD, Adam, using the reparametrization trick.

Theorem 4.0.1. $\log p_\theta(\boldsymbol{x}) \geq \mathcal{L}^k_{TD-IWAE}(\boldsymbol{x})$

Proof. We can argue that the weights w_i are an unbiased estimate of $p_\theta(\mathbf{x})$:

$$\begin{aligned}\mathbb{E}_{q_\phi(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})}\left[\frac{p_\theta(\mathbf{x},\mathbf{z}_1,\mathbf{z}_2)}{q_\phi(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})}\right] &= \iint_{\mathbf{z}_1,\mathbf{z}_2} q_\phi(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x}) \cdot \frac{p_\theta(\mathbf{x},\mathbf{z}_1,\mathbf{z}_2)}{q_\phi(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})}\, d\mathbf{z}_2\, d\mathbf{z}_1 \\ &= \iint_{\mathbf{z}_1,\mathbf{z}_2} p_\theta(\mathbf{x},\mathbf{z}_1,\mathbf{z}_2)\, d\mathbf{z}_2\, d\mathbf{z}_1 = p_\theta(\mathbf{x})\end{aligned} \tag{4.3}$$

The arguments in support of the statement are the application of Jensen-inequality and the Equation 4.3.

$$\begin{aligned}\mathcal{L}^k_{TD-IWAE}(\mathbf{x}) &= \mathbb{E}_{q_\phi(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})}\left[\log\left(\frac{1}{k}\sum_{i=1}^{k}\frac{p_\theta(\mathbf{x},\mathbf{z}_1^i,\mathbf{z}_2^i)}{q_\phi(\mathbf{z}_1^i,\mathbf{z}_2^i|\mathbf{x})}\right)\right] \leq \log\left(\mathbb{E}_{q_\phi(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})}\left[\frac{1}{k}\sum_{i=1}^{k}\frac{p_\theta(\mathbf{x},\mathbf{z}_1^i,\mathbf{z}_2^i)}{q_\phi(\mathbf{z}_1^i,\mathbf{z}_2^i|\mathbf{x})}\right]\right) \\ &= \log\left(\mathbb{E}_{q_\phi(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})}\left[\frac{1}{k}\sum_{i=1}^{k} w_i\right]\right) = \log\left(\frac{1}{k}\sum_{i=1}^{k}\mathbb{E}_{q_\phi(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})}\left[w_i\right]\right) \\ &= \log\left(\frac{1}{k}\sum_{i=1}^{k} p_\theta(\mathbf{x})\right) = \log p_\theta(\mathbf{x})\end{aligned}$$

□

The proofs of the properties of $\mathcal{L}^k_{IWAE}$ are also applicable in the two-latent case, consequently, similar theorems can be stated about $\mathcal{L}^k_{TD-IWAE}$. The special case of $k = 1$ returns the original TDVAE formulation which, along with Theorem 4.0.2 gives the corollary that the TD-IWAE ELBO provides a lower bound than the TDVAE ELBO.

Theorem 4.0.2.

1. $\mathcal{L}^k_{TD-IWAE} \geq \mathcal{L}^m_{TD-IWAE}$ *for* $k \geq m$
2. *If* $\dfrac{p_\theta(\boldsymbol{x}, z_1^i, z_2^i)}{q_\phi(z_1^i, z_2^i|\boldsymbol{x})}$ *is bounded, then* $\lim_{k\to\infty} \mathcal{L}^k_{TD-IWAE}(\boldsymbol{x}) = \log p_\theta(\boldsymbol{x})$.

In order to understand what variational posterior is formulated by the importance sampling calculation done in the ELBO during training, we can follow the reasoning proposed in the reinterpretation of IWAE from Cremer et al. [7]. The multisample procedure is performed on both latent layers, and the ELBO loss implicitly weights the obtained samples, given a weight w_i corresponded to the sample $(\mathbf{z}_1^i, \mathbf{z}_2^i)$. Then, the implicit distribution $q_{IW}(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x}, (\mathbf{z}_1^2, \mathbf{z}_2^2), ..., (\mathbf{z}_1^k, \mathbf{z}_2^k))$ and the expected importance weighted distribution $q_{EW}(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x})$ can be formulated in a similar way as it was with defining $q_{IW}(\mathbf{z}|\mathbf{x})$. The case of $k = 1$ falls back to the original approximate posterior $q_{IW}(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x}) = q_{\boldsymbol{\phi}}(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x})$.

$$q_{IW}(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x}, (\mathbf{z}_1^2, \mathbf{z}_2^2), ..., (\mathbf{z}_1^k, \mathbf{z}_2^k)) = \frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}_1, \mathbf{z}_2)}{\frac{1}{k}\left(\frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z}_1,\mathbf{z}_2)}{q_{\boldsymbol{\phi}}(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})} + \sum_{i=2}^{k} \frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z}_1^i,\mathbf{z}_2^i)}{q_{\boldsymbol{\phi}}(\mathbf{z}_1^i,\mathbf{z}_2^i|\mathbf{x})}\right)} \tag{4.4}$$

$$\begin{aligned} q_{EW}(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x}) &= \mathbb{E}_{(\mathbf{z}_1^2,\mathbf{z}_2^2),...,(\mathbf{z}_1^k,\mathbf{z}_2^k)}\left[q_{IW}(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x}, (\mathbf{z}_1^2, \mathbf{z}_2^2), ..., (\mathbf{z}_1^k, \mathbf{z}_2^k))\right] \\ &= \mathbb{E}_{(\mathbf{z}_1^2,\mathbf{z}_2^2),...,(\mathbf{z}_1^k,\mathbf{z}_2^k)}\left[\frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}_1, \mathbf{z}_2)}{\frac{1}{k}\left(\frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z}_1,\mathbf{z}_2)}{q_{\boldsymbol{\phi}}(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})} + \sum_{i=2}^{k} \frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z}_1^i,\mathbf{z}_2^i)}{q_{\boldsymbol{\phi}}(\mathbf{z}_1^i,\mathbf{z}_2^i|\mathbf{x})}\right)}\right] \end{aligned} \tag{4.5}$$

Following the lines of the proof that the IWAE ELBO is equivalent with the VAE ELBO with q_{IW} in expectation, we provide the derivation of that the TD-IWAE lower bound is the same as the TDVAE bound with the above defined $q_{IW}(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x}, (\mathbf{z}_1^2, \mathbf{z}_2^2), ..., (\mathbf{z}_1^k, \mathbf{z}_2^k))$. Moreover, we show that $q_{EW}(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x})$ is a normalized distribution, using the proof from [7] as a basis.

Theorem 4.0.3. $\mathbb{E}_{(z_1,z_2)^{2,...,k}}\left[\mathcal{L}_{TD}[q_{IW}(z_1, z_2|\boldsymbol{x}, .)]\right] = \mathcal{L}_{TD-IWAE}[q_{\boldsymbol{\phi}}(z_1, z_2|\boldsymbol{x})]$

Proof. For clearer readability, the following notational simplifications are introduced:

- The samples $(\mathbf{z}_1^i, \mathbf{z}_2^i), (\mathbf{z}_1^{i+1}, \mathbf{z}_2^{i+1}), ..., (\mathbf{z}_1^j, \mathbf{z}_2^j)$ drawn from $q_{\boldsymbol{\phi}}(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x})$ will be denoted as $(\mathbf{z}_1, \mathbf{z}_2)^{i,i+1,..,j}$
- The conditional dependence of q_{IW} on the samples will be denoted as:
  $q_{IW}(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x}, (\mathbf{z}_1^2, \mathbf{z}_2^2), ..., (\mathbf{z}_1^k, \mathbf{z}_2^k)) = q_{IW}(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x}, .)$

The expectation in the ELBO $\mathcal{L}_{TDVAE}[q_{IW}]$ will be instantly written in an integral form instead of taking an expectation with an unnormalized distribution. As before, we will also use the notation that $\mathbf{z} = \mathbf{z}_1$.

$$
\begin{aligned}
\mathbb{E}_{(\mathbf{z}_1,\mathbf{z}_2)^{2,\dots,k}}\left[\mathcal{L}_{TDVAE}[q_{IW}]\right] &= \mathbb{E}_{(\mathbf{z}_1,\mathbf{z}_2)^{2,\dots,k}}\left[\iint_{\mathbf{z}_1,\mathbf{z}_2} q_{IW}(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x},.)\log\frac{p_\theta(\mathbf{x},\mathbf{z}_1,\mathbf{z}_2)}{q_{IW}(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x},.)}\,d\mathbf{z}_2\,d\mathbf{z}_1\right] \\
&= \mathbb{E}_{(\mathbf{z}_1,\mathbf{z}_2)^{2,\dots,k}}\left[\iint_{\mathbf{z}_1,\mathbf{z}_2} q_{IW}(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x},.)\log\frac{p_\theta(\mathbf{x},\mathbf{z}_1,\mathbf{z}_2)}{\frac{p_\theta(\mathbf{x},\mathbf{z}_1,\mathbf{z}_2)}{\frac{1}{k}\left(\sum_{i=1}^{k}\frac{p_\theta(\mathbf{x},\mathbf{z}_1^i,\mathbf{z}_2^i)}{q_\phi(\mathbf{z}_1^i,\mathbf{z}_2^i|\mathbf{x})}\right)}}\,d\mathbf{z}_2\,d\mathbf{z}_1\right] \\
&= \mathbb{E}_{(\mathbf{z}_1,\mathbf{z}_2)^{2,\dots,k}}\left[\iint_{\mathbf{z}_1,\mathbf{z}_2} q_{IW}(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x},.)\log\frac{1}{k}\sum_{i=1}^{k}\frac{p_\theta(\mathbf{x},\mathbf{z}_1^i,\mathbf{z}_2^i)}{q_\phi(\mathbf{z}_1^i,\mathbf{z}_2^i|\mathbf{x})}\,d\mathbf{z}_2\,d\mathbf{z}_1\right] \\
&= \mathbb{E}_{(\mathbf{z}_1,\mathbf{z}_2)^{2,\dots,k}}\left[\iint_{\mathbf{z}_1,\mathbf{z}_2} k\frac{\frac{p_\theta(\mathbf{x},\mathbf{z}_1,\mathbf{z}_2)}{q_\phi(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})}}{\sum_{i=1}^{k}\frac{p_\theta(\mathbf{x},\mathbf{z}_1^i,\mathbf{z}_2^i)}{q_\phi(\mathbf{z}_1^i,\mathbf{z}_2^i|\mathbf{x})}}q_\phi(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})\log\frac{1}{k}\sum_{i=1}^{k}\frac{p_\theta(\mathbf{x},\mathbf{z}_1^i,\mathbf{z}_2^i)}{q_\phi(\mathbf{z}_1^i,\mathbf{z}_2^i|\mathbf{x})}\,d\mathbf{z}_2\,d\mathbf{z}_1\right] \\
&= \mathbb{E}_{(\mathbf{z}_1,\mathbf{z}_2)^{2,\dots,k}}\left[\iint_{\mathbf{z}_1^1,\mathbf{z}_2^1} k\frac{\frac{p_\theta(\mathbf{x},\mathbf{z}_1^1,\mathbf{z}_2^1)}{q_\phi(\mathbf{z}_1^1,\mathbf{z}_2^1|\mathbf{x})}}{\sum_{i=1}^{k}\frac{p_\theta(\mathbf{x},\mathbf{z}_1^i,\mathbf{z}_2^i)}{q_\phi(\mathbf{z}_1^i,\mathbf{z}_2^i|\mathbf{x})}}q_\phi(\mathbf{z}_1^1,\mathbf{z}_2^1|\mathbf{x})\log\frac{1}{k}\sum_{i=1}^{k}\frac{p_\theta(\mathbf{x},\mathbf{z}_1^i,\mathbf{z}_2^i)}{q_\phi(\mathbf{z}_1^i,\mathbf{z}_2^i|\mathbf{x})}\,d\mathbf{z}_2^1\,d\mathbf{z}_1^1\right] \\
&= \mathbb{E}_{(\mathbf{z}_1,\mathbf{z}_2)^{1,2,\dots,k}}\left[k\frac{\frac{p_\theta(\mathbf{x},\mathbf{z}_1^1,\mathbf{z}_2^1)}{q_\phi(\mathbf{z}_1^1,\mathbf{z}_2^1|\mathbf{x})}}{\sum_{i=1}^{k}\frac{p_\theta(\mathbf{x},\mathbf{z}_1^i,\mathbf{z}_2^i)}{q_\phi(\mathbf{z}_1^i,\mathbf{z}_2^i|\mathbf{x})}}\cdot\log\frac{1}{k}\sum_{i=1}^{k}\frac{p_\theta(\mathbf{x},\mathbf{z}_1^i,\mathbf{z}_2^i)}{q_\phi(\mathbf{z}_1^i,\mathbf{z}_2^i|\mathbf{x})}\right] \\
&= \mathbb{E}_{(\mathbf{z}_1,\mathbf{z}_2)^{1,2,\dots,k}}\left[\frac{\sum_{i=1}^{k}\frac{p_\theta(\mathbf{x},\mathbf{z}_1^1,\mathbf{z}_2^1)}{q_\phi(\mathbf{z}_1^1,\mathbf{z}_2^1|\mathbf{x})}}{\sum_{i=1}^{k}\frac{p_\theta(\mathbf{x},\mathbf{z}_1^i,\mathbf{z}_2^i)}{q_\phi(\mathbf{z}_1^i,\mathbf{z}_2^i|\mathbf{x})}}\cdot\log\frac{1}{k}\sum_{i=1}^{k}\frac{p_\theta(\mathbf{x},\mathbf{z}_1^i,\mathbf{z}_2^i)}{q_\phi(\mathbf{z}_1^i,\mathbf{z}_2^i|\mathbf{x})}\right] \\
&= \mathbb{E}_{(\mathbf{z}_1,\mathbf{z}_2)^{1,2,\dots,k}}\left[\frac{\sum_{i=1}^{k}\frac{p_\theta(\mathbf{x},\mathbf{z}_1^i,\mathbf{z}_2^i)}{q_\phi(\mathbf{z}_1^i,\mathbf{z}_2^i|\mathbf{x})}}{\sum_{i=1}^{k}\frac{p_\theta(\mathbf{x},\mathbf{z}_1^i,\mathbf{z}_2^i)}{q_\phi(\mathbf{z}_1^i,\mathbf{z}_2^i|\mathbf{x})}}\cdot\log\frac{1}{k}\sum_{i=1}^{k}\frac{p_\theta(\mathbf{x},\mathbf{z}_1^i,\mathbf{z}_2^i)}{q_\phi(\mathbf{z}_1^i,\mathbf{z}_2^i|\mathbf{x})}\right] \\
&= \mathbb{E}_{(\mathbf{z}_1,\mathbf{z}_2)^{1,2,\dots,k}}\left[\log\frac{1}{k}\sum_{i=1}^{k}\frac{p_\theta(\mathbf{x},\mathbf{z}_1^i,\mathbf{z}_2^i)}{q_\phi(\mathbf{z}_1^i,\mathbf{z}_2^i|\mathbf{x})}\right] \\
&= \mathcal{L}_{TD-IWAE}[q_\phi]
\end{aligned}
$$

where after taking the expectation over all the sampled $\mathbf{z}$s, the multiplier k can be written as a sum of k items, since $\mathbf{z}_i$ shares the same expectation with $\mathbf{z}_1$.

□

Theorem 4.0.4. *$q_{EW}(z_1, z_2|\boldsymbol{x})$ is a normalized distribution.*

Proof. For clearer readability, the samples $(\mathbf{z}_1^i, \mathbf{z}_2^i)$, $(\mathbf{z}_1^{i+1}, \mathbf{z}_2^{i+1})$, ..., $(\mathbf{z}_1^j, \mathbf{z}_2^j)$ drawn from $q(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x})$ will be denoted as $(\mathbf{z}_1, \mathbf{z}_2)^{i,i+1,\dots,j}$.

$$
\begin{aligned}
\iint_{\mathbf{z}_1,\mathbf{z}_2} q_{EW}(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})\,d\mathbf{z}_2\,d\mathbf{z}_1 &= \iint_{\mathbf{z}_1,\mathbf{z}_2} \mathbb{E}_{(\mathbf{z}_1,\mathbf{z}_2)^{2,\dots,k}}\left[\frac{p_\theta(\mathbf{x},\mathbf{z}_1,\mathbf{z}_2)}{\frac{1}{k}\left(\frac{p_\theta(\mathbf{x},\mathbf{z}_1,\mathbf{z}_2)}{q_\phi(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})}+\sum_{i=2}^{k}\frac{p_\theta(\mathbf{x},\mathbf{z}_1^i,\mathbf{z}_2^i)}{q_\phi(\mathbf{z}_1^i,\mathbf{z}_2^i|\mathbf{x})}\right)}\right]d\mathbf{z}_2\,d\mathbf{z}_1\\
&= \iint_{\mathbf{z}_1,\mathbf{z}_2} \frac{q_\phi(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})}{q_\phi(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})}\cdot\mathbb{E}_{(\mathbf{z}_1,\mathbf{z}_2)^{2,\dots,k}}\left[\frac{p_\theta(\mathbf{x},\mathbf{z}_1,\mathbf{z}_2)}{\frac{1}{k}\left(\frac{p_\theta(\mathbf{x},\mathbf{z}_1,\mathbf{z}_2)}{q_\phi(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})}+\sum_{i=2}^{k}\frac{p_\theta(\mathbf{x},\mathbf{z}_1^i,\mathbf{z}_2^i)}{q_\phi(\mathbf{z}_1^i,\mathbf{z}_2^i|\mathbf{x})}\right)}\right]d\mathbf{z}_2\,d\mathbf{z}_1\\
&= \mathbb{E}_{(\mathbf{z}_1,\mathbf{z}_2)}\mathbb{E}_{(\mathbf{z}_1,\mathbf{z}_2)^{2,\dots,k}}\left[\frac{\frac{p_\theta(\mathbf{x},\mathbf{z}_1,\mathbf{z}_2)}{q_\phi(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})}}{\frac{1}{k}\left(\frac{p_\theta(\mathbf{x},\mathbf{z}_1,\mathbf{z}_2)}{q_\phi(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})}+\sum_{i=2}^{k}\frac{p_\theta(\mathbf{x},\mathbf{z}_1^i,\mathbf{z}_2^i)}{q_\phi(\mathbf{z}_1^i,\mathbf{z}_2^i|\mathbf{x})}\right)}\right]\\
&= k\cdot\mathbb{E}_{(\mathbf{z}_1,\mathbf{z}_2)^{1,\dots,k}}\left[\frac{\frac{p_\theta(\mathbf{x},\mathbf{z}_1^1,\mathbf{z}_2^1)}{q_\phi(\mathbf{z}_1^1,\mathbf{z}_2^1|\mathbf{x})}}{\sum_{i=1}^{k}\frac{p_\theta(\mathbf{x},\mathbf{z}_1^i,\mathbf{z}_2^i)}{q_\phi(\mathbf{z}_1^i,\mathbf{z}_2^i|\mathbf{x})}}\right]\\
&= \sum_{j=1}^{k}\mathbb{E}_{(\mathbf{z}_1,\mathbf{z}_2)^{1,\dots,k}}\left[\frac{\frac{p_\theta(\mathbf{x},\mathbf{z}_1^j,\mathbf{z}_2^j)}{q_\phi(\mathbf{z}_1^j,\mathbf{z}_2^j|\mathbf{x})}}{\sum_{i=1}^{k}\frac{p_\theta(\mathbf{x},\mathbf{z}_1^i,\mathbf{z}_2^i)}{q_\phi(\mathbf{z}_1^i,\mathbf{z}_2^i|\mathbf{x})}}\right]\\
&= \mathbb{E}_{(\mathbf{z}_1,\mathbf{z}_2)^{1,\dots,k}}\left[\frac{\sum_{j=1}^{k}\frac{p_\theta(\mathbf{x},\mathbf{z}_1^j,\mathbf{z}_2^j)}{q_\phi(\mathbf{z}_1^j,\mathbf{z}_2^j|\mathbf{x})}}{\sum_{i=1}^{k}\frac{p_\theta(\mathbf{x},\mathbf{z}_1^i,\mathbf{z}_2^i)}{q_\phi(\mathbf{z}_1^i,\mathbf{z}_2^i|\mathbf{x})}}\right]\\
&= 1
\end{aligned}
$$

where we applied the linearity of expectation and the fact that $\mathbf{z}_i$ shares the same expectation with $\mathbf{z}_1$.

□

It is worth noting that by reformulating the expression of $q_{IW}(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})$, one can see that it depends on the true posterior:

$$
q_{IW}(\mathbf{z}_1^1,\mathbf{z}_2^1|\mathbf{x},(\mathbf{z}_1^2,\mathbf{z}_2^2),\dots,(\mathbf{z}_1^k,\mathbf{z}_2^k)) = \frac{p_\theta(\mathbf{x},\mathbf{z}_1^1,\mathbf{z}_2^1)}{\frac{1}{k}\left(\sum_{i=1}^{k}\frac{p_\theta(\mathbf{x},\mathbf{z}_1^i,\mathbf{z}_2^i)}{q_\phi(\mathbf{z}_1^i,\mathbf{z}_2^i|\mathbf{x})}\right)} = \frac{p_\theta(\mathbf{x})}{\frac{1}{k}\left(\sum_{i=1}^{k}\frac{p_\theta(\mathbf{x},\mathbf{z}_1^i,\mathbf{z}_2^i)}{q_\phi(\mathbf{z}_1^i,\mathbf{z}_2^i|\mathbf{x})}\right)}p_\theta(\mathbf{z}_1^1,\mathbf{z}_2^1|\mathbf{x})
$$

This also implies, as the denominator converges to $p_\theta(\mathbf{x})$ if $k\to\infty$, that $q_{IW}(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})\to p_\theta(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})$, analogously to the single layer case.

While performing prediction with the trained model, it is essential to take care of the resampling for the purpose of properly drawing samples from the implicit distribution. The method is shown in Algorithm 2, formulated based on the routine by Cremer et al. for IWAE. Owing to the fact that now we are sampling from a joint posterior, the method returns a pair of vectors $(\mathbf{z}_1^j,\mathbf{z}_2^j)$. To marginalize the sample, one can just simply ignore the other vector from the returned pair. With this procedure we can also consider a sample from $q_\phi(\mathbf{z}_1|\mathbf{x})$.

Algorithm 2 Sampling $q_{EW}(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x})$

```
Input: $k$: number of samples drawn
for $i = 1, ..., k$ do
    Sample $(\mathbf{z}_1^i, \mathbf{z}_2^i) \sim q_{\boldsymbol{\phi}}(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x})$ where $q_{\boldsymbol{\phi}}(\mathbf{z}_1, \mathbf{z}_2|x) = q_{\boldsymbol{\phi}}(\mathbf{z}_1|\mathbf{x}, \mathbf{z}_2) \cdot q_{\boldsymbol{\phi}}(\mathbf{z}_2|\mathbf{x})$
    Calculate $w_i = \frac{p_\theta(\mathbf{x}, \mathbf{z}_1^i, \mathbf{z}_2^i)}{q_{\boldsymbol{\phi}}(\mathbf{z}_1^i, \mathbf{z}_2^i|\mathbf{x})}$
end for
Normalization: $\tilde{w}_i = \frac{w_i}{\sum_{i=1}^{k} w_i}$ for $\forall i$
Resample according to the obtained normalized weights $\tilde{\mathbf{w}}_i$: $j \sim Categorical(\tilde{\mathbf{w}}_i)$
Return: $(\mathbf{z}_1^j, \mathbf{z}_2^j)$
```

Chapter 5

Experiments

This chapter presents the analysis carried out on the learnt latent spaces in TDVAE and TD-IWAE. Our first objective is to have an insight into the learnt representations, and our second goal is to compare those inferred by TDVAE and TD-IWAE. Note that when we are drawing samples from the variational posteriors, in TD-IWAE we perform it by using the discussed Sampling-Importance-Resampling procedure.

The models trained and examined are the following:

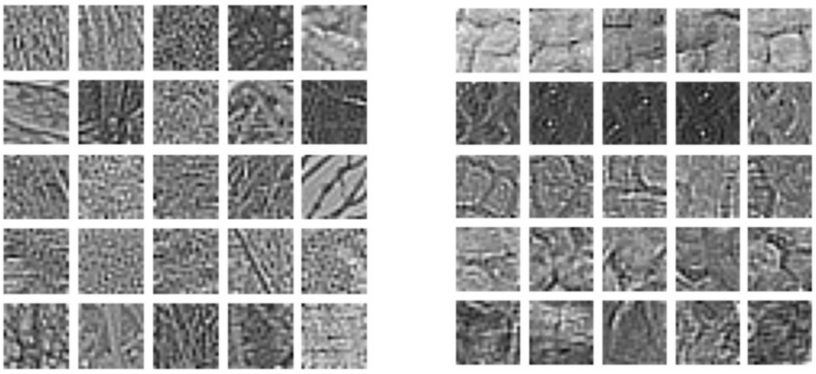
- TDVAE
- TD-IWAE trained with $k = 1$ samples
- TD-IWAE trained with $k = 4$ samples
- TD-IWAE trained with $k = 10$ samples
- TD-IWAE trained with $k = 50$ samples

All of the experiments are performed for each model, and this summary aims to outline the results for all of them. In some cases, in order to compare the baseline and importance weighted networks, the visualizations are depicted only for the vanilla TDVAE and the more interesting TD-IWAE models, and not for all of the networks. The exeriments are implemented in Python, using packages fundamental in machine learning and data science, such as Matplotlib, Seaborn, Sklearn, Pandas and Numpy. The model construction and training were performed in PyTorch.

5.1 Datasets and setup

In this thesis two datasets were used. The first one is a collection of 20×20 patches from natural images, used for training the models. The dataset originates from the van Hateren natural image database [33], but the dataset used in this thesis with the transformed and cut patches is taken from [9], [8]. All the patches are whitened and their intensity distribution was normalized to follow a standard normal distribution.

The second dataset is a set of 20×20 texture image patches, representing five texture classes (oat, leather, soil, carpet and bubbles). The dataset, transformed in the same way as the natural image dataset, is taken from the work of [9], [8]. Example patches from both the natural and texture test data are shown in Figure 5.1. Both of the datasets are separated into a training and test set, containing 640.000 and 64.000 patches, respectively. The models were trained on the training set of natural images, and then for qualitative and quantitative evaluation both test sets were used.



(a) Example natural patches from the test set

(b) Example texture patches from the test set. Each row contains patches belonging to the same family

Figure 5.1: Patches from the datasets

The models were trained from scratch with the default weight initialization, for 5000 epochs using the Adam optimizer with a learning rate of 0.0001 and batch size 400. For stabilizing the training procedure, gradient clipping was also utilized. In order to facilitate learning in the higher latent space, β-annealing is applied for the KL-term of $\mathbf{z}_2$. The lower latent layer did not require any annealing procedure. The optimization criterion is conventionally should be minimized, hence the negative of the ELBO was used for the objective function. Due to the fact that we have images with real values, $p_{\theta}(\mathbf{x}|\mathbf{z})$ is assumed to follow a multivariate Normal distribution, and then the reconstruction term $\log p_{\theta}(\mathbf{x}|\mathbf{z})$ was calculated as the mean-squared error between the original $\mathbf{x}$ and the reconstructed $\hat{\mathbf{x}}$, for each component, since the log-likelihood for a univariate Gaussian with mean μ and variance σ^2 is $-\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n-\mu)^2+c$. The latent $\mathbf{z}_1$ was chosen to have 450 dimensions in order to act for an overcomplete representation, and the latent $\mathbf{z}_2$ possesses 20 dimensions.

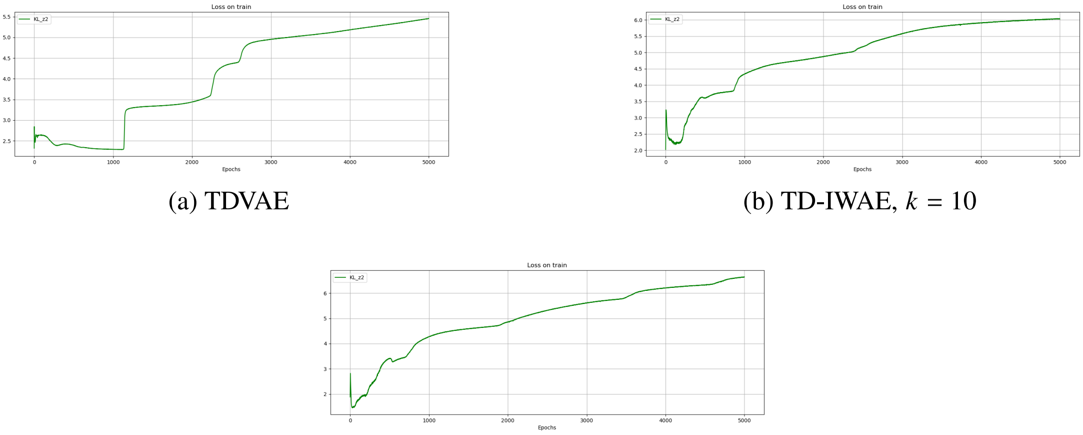
5.2 Training and evaluation

Initially, the models were trained without β-annealing, and as it could be seen in Figure 5.2, training with multiple samples resulted in a more smooth improvement of the KL-divergence term for $\mathbf{z}_2$. However, it seemed that the variational posterior for $\mathbf{z}_2$ is developing quite slowly. There was no serious posterior collapse as the KL-value for $\mathbf{z}_2$ started increasing, but in order to learn a richer posterior in a reasonable time, we decided to apply the β-annealing method to $\mathbf{z}_2$.

In the TDVAE model it was straightforward to introduce a multiplier in front of the divergence

$KL[q_\phi(\mathbf{z}_2|\mathbf{x})||p_\theta(\mathbf{z}_2)]$, but in the TD-IWAE model there are no separate KL-divergences. Hence as a solution we plugged in the β into the equation of its ELBO as the multiplier of $\big(\log p_\theta(\mathbf{z}_2^i) - \log q_\phi(\mathbf{z}_2^i|\mathbf{x})\big)$ for the reason that these terms plays the main part in the calculations in the KL-divergence between the variational posterior and prior of $\mathbf{z}_2$, even though this expression does not count as a full KL-divergence.

After that, the β-annealing was incorporated in all of the models, and the training method was performed with increasing β linearly from the value 0.1 until the value 1, throughout 2500 epochs. All of the models exhibited similar behaviour in terms of the KL-divergence for $\mathbf{z}_2$, shooting high at the beginning of the training and then slowly falling back to a reasonable value as example training curves show this in Figure 5.3. All the networks achieved a much higher value for KL-divergence corresponded with $\mathbf{z}_2$ than the ones trained without annealing. Thus, for evaluating the performances on the test datasets, the models which were trained with annealing were used for evaluation.

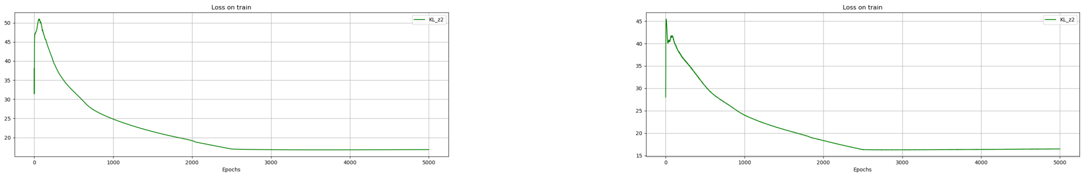


(a) TDVAE

(b) TD-IWAE, $k = 10$

(c) TD-IWAE, $k = 50$

Figure 5.2: The improvement of Kullback-Leibler divergence between $q_\phi(\mathbf{z}_2|\mathbf{x})$ and $p_\theta(\mathbf{z}_2)$ in models trained without β-annealing

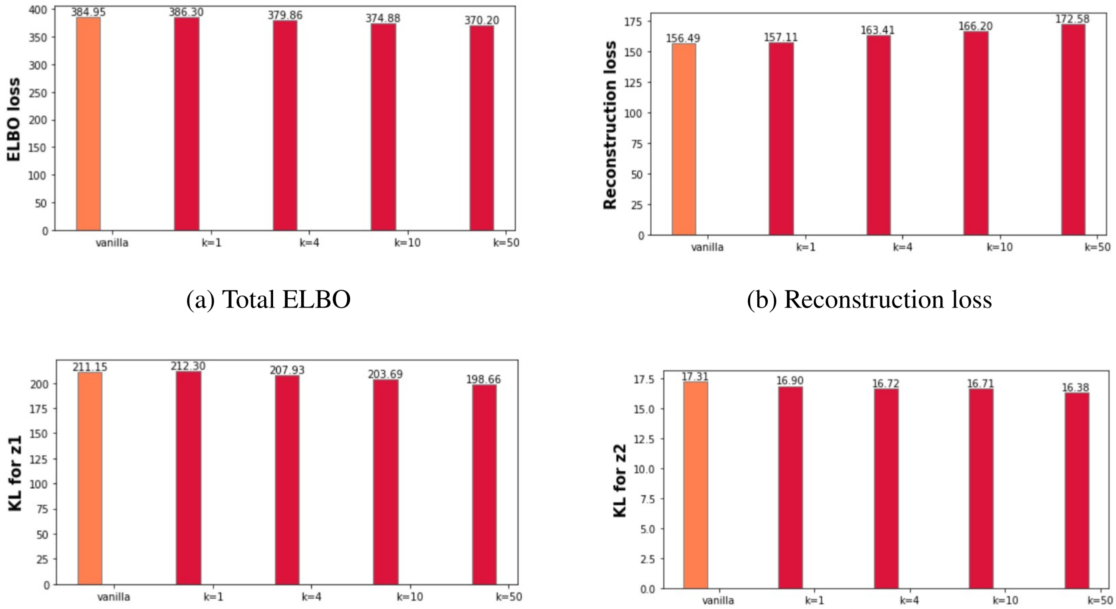


(a) TDVAE

(b) TD-IWAE, $k = 50$

Figure 5.3: The evolution of Kullback-Leibler divergence between $q_\phi(\mathbf{z}_2|\mathbf{x})$ and $p_\theta(\mathbf{z}_2)$ in models trained with linear β-annealing

The models trained on the natural images were evaluated on both the natural test set and texture test data considering the ELBO averaged over the test data points. The results are aggregated in the Figures 5.4, and 5.5. The ELBO, still viewed as a loss function, is lower on both datasets when $k > 1$ samples are used in TD-IWAE, and as the number of samples increased, the lower the ELBO becomes. This phenomenon is even more present in the results performed on the texture data. The reconstruction loss increased while the KL-divergence for $\mathbf{z}_1$ declined when more samples are drawn, and the KL-divergence for $\mathbf{z}_2$ stays approximately the same as the baseline results even with large k, on both datasets. In the case of $k = 1$ the TD-IWAE ELBO is roughly the same as the vanilla TDVAE ELBO, which is as expected. Altough the individual terms in average does not reflect if an increasing k had a positive impact on the performance, the overall ELBO is definitely lower while k grows larger, matching the experimental observations in the literature.



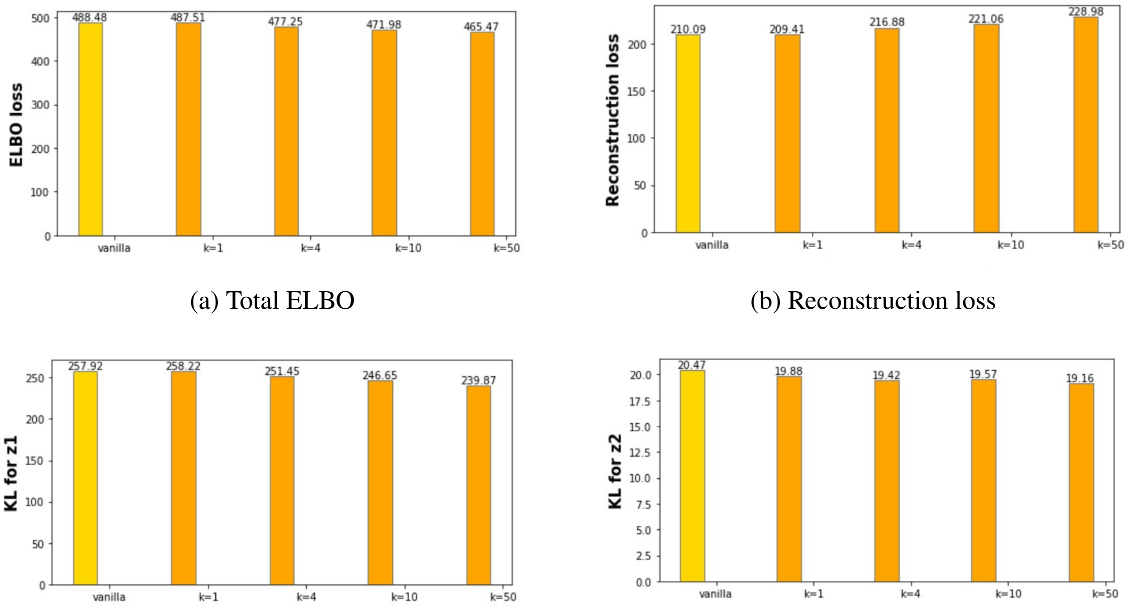
(a) Total ELBO

(b) Reconstruction loss

(c) KL-divergence for $\mathbf{z}_1$

(d) KL-divergence for $\mathbf{z}_2$

Figure 5.4: The average test ELBO terms given by the models on the natural test dataset. The label "vanilla" indicates the TDVAE model, while the labels for different k values denote the TD-IWAE models trained with the specified k



(a) Total ELBO

(b) Reconstruction loss

(c) KL-divergence for $\mathbf{z}_1$

(d) KL-divergence for $\mathbf{z}_2$

Figure 5.5: The average test ELBO terms given by the models on the texture test dataset. The label "vanilla" indicates the TDVAE model, while the labels for different k values denote the TD-IWAE models trained with the specified k

5.3 Examination of the latent space of $\mathbf{z}_2$

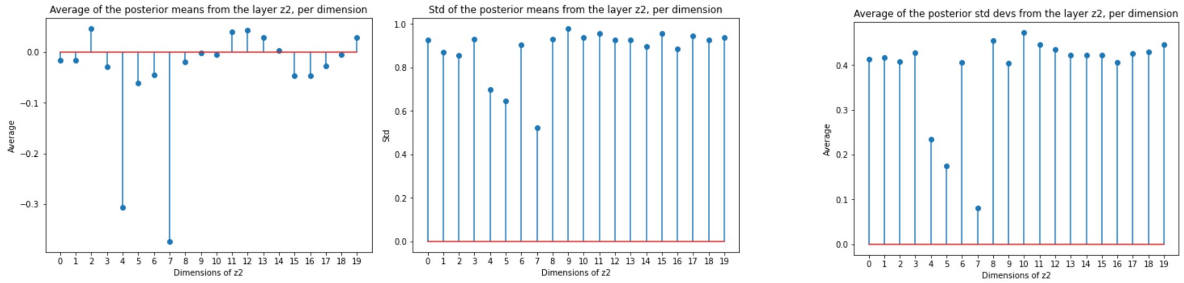
Along with the general examination of the latent space, the study of the higher latent layer is mainly focused on its sensitivity to texture characteristics. The motivation is coming from neuroscience as the V2 area in the cortical hierarchy is known to to be accounted for producing texture representations [8]. To examine the texture-selectivity of the $\mathbf{z}_2$ representations, the evaluation is performed on the texture test data. In addition, examining textures is valuable also from a computer vision perspective, since they are key characteristics of images [8].

Firstly, the variational posterior $q_\phi(\mathbf{z}_2|\mathbf{x})$ was studied through generating the posterior mean and std for the test images, and then aggregating their values per dimension by taking their average and standard deviation. The results are depicted in Figures 5.6 - 5.7 - 5.8 - 5.9 - 5.10.

It is observable in all models that while the average of posterior means close to zero in many dimensions, the standard deviation of these means are relatively high, implying that the values of posterior means in these dimensions are activated, but their sign varies between observations. There are few coordinates with more stable mean value across images, for instance the dimensions 1, 5, 15 in TD-IWAE, $k = 50$. By looking at the average of the posterior standard deviation, all the coordinates are far away from the unit value, instead grouping around $\sigma = 0.4$, in each model.

The activity of dimensions can be identified by looking at the differences between their distribution and their prior, therefore activity is reflected not only in the relatively high values of the standard deviation of the posterior means, but also in smaller means of posterior standard deviations. The plots show that in each model, all of the $\mathbf{z}_2$ dimensions were activated, and none of them collapsed to the prior. Example distributions of each dimensions are shown in Figure

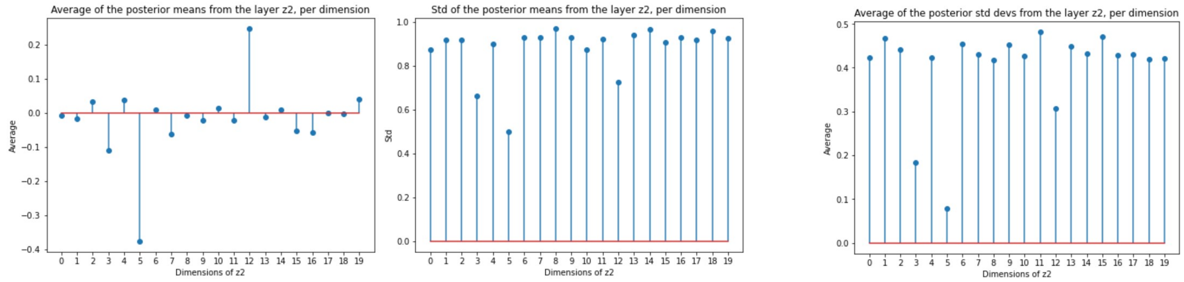
5.11 from TDVAE and from TD-IWAE trained with $k = 10$ samples, in comparison with their standard normal prior. Additionally, no visible differences can be seen when comparing the figures of TDVAE and all the TD-IWAEs.



(a)

(b)

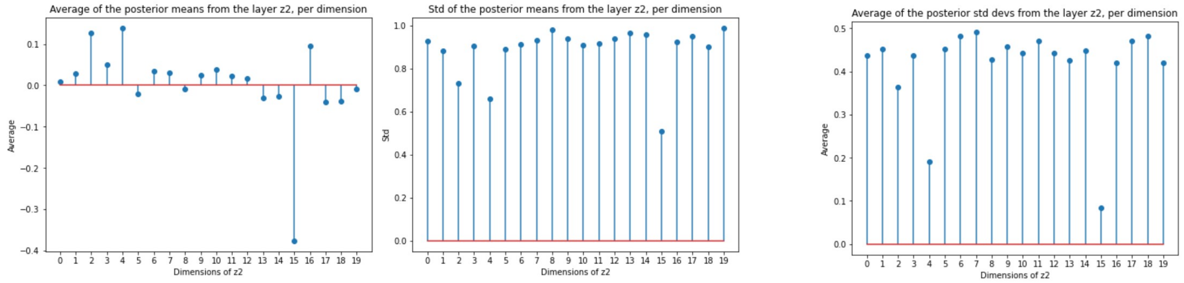
Figure 5.6: Examination of the mean and standard deviation values of the posterior $q_\phi(\mathbf{z}_2|\mathbf{x})$ in TDVAE



(a)

(b)

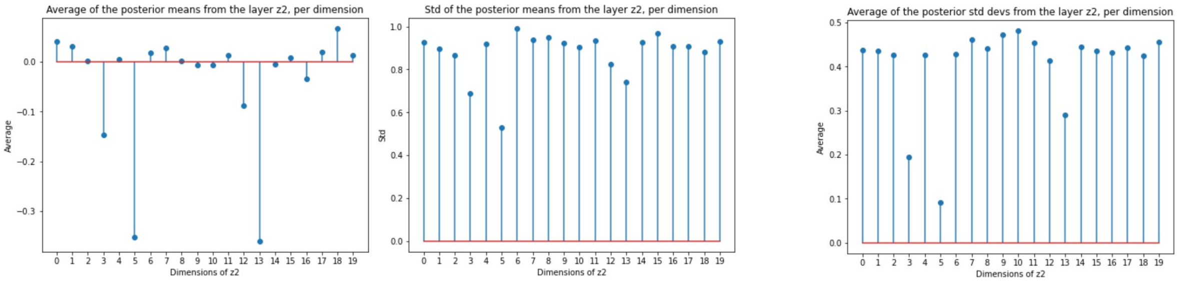
Figure 5.7: Examination of the mean and standard deviation values of the posterior $q_\phi(\mathbf{z}_2|\mathbf{x})$ in TD-IWAE, $k = 1$



(a)

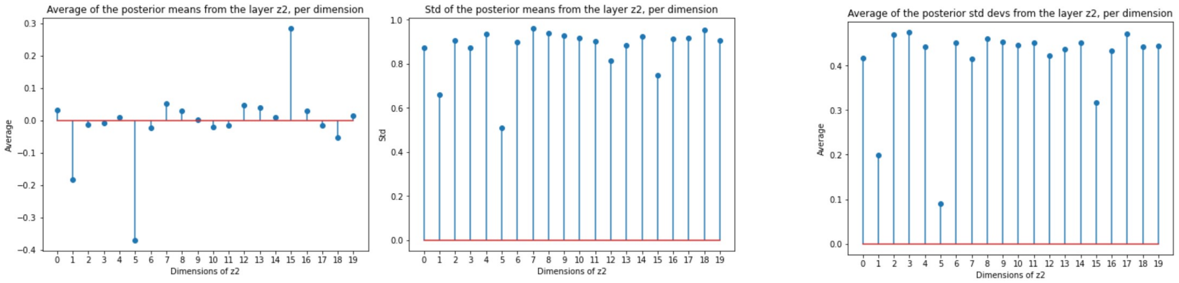
(b)

Figure 5.8: Examination of the mean and standard deviation values of the posterior $q_\phi(\mathbf{z}_2|\mathbf{x})$ in TD-IWAE, $k = 4$



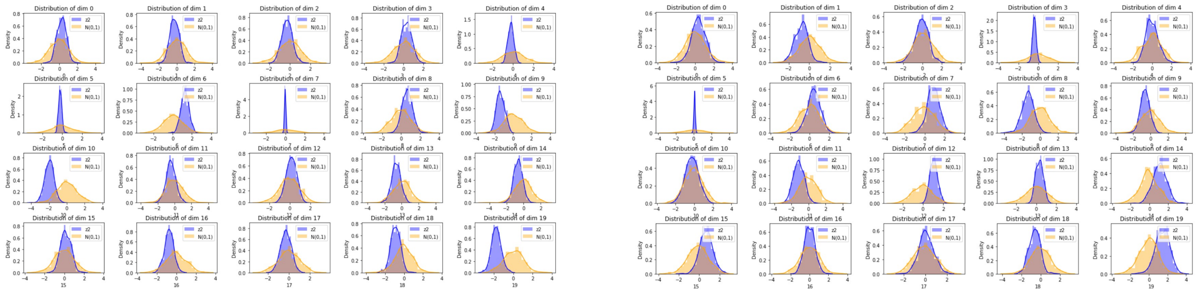
(a) (b)

Figure 5.9: Examination of the mean and standard deviation values of the posterior $q_\phi(\mathbf{z}_2|\mathbf{x})$ in TD-IWAE, $k = 10$



(a) (b)

Figure 5.10: Examination of the mean and standard deviation values of the posterior $q_\phi(\mathbf{z}_2|\mathbf{x})$ in TD-IWAE, $k = 50$

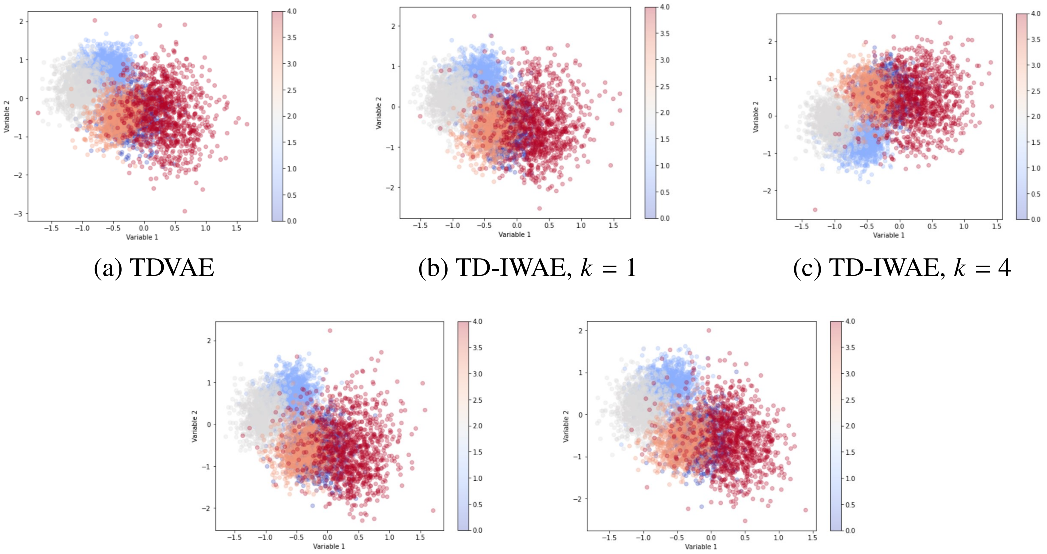


(a) TDVAE (b) TD-IWAE, $k = 10$

Figure 5.11: Visualizations of the distribution of $q_\phi(\mathbf{z}_2|\mathbf{x})$ and its comparison with the standard normal distribution, per dimension

In order to examine the texture-selectivity in the higher latent space, all the $\mathbf{z}_2$ representations corresponding to texture stimuli were plotted in two dimension and colored according to their texture classes. The 2D display in Figure 5.12 is performed by taking the two dimensions where the posterior variance was the lowest.

The experiments showed that the points are clustered according to their class labels, meaning that the five texture families formulate groups in the latent space, even though the training data contained natural patches and the models were not directly trained to recognize different texture types. The clusters are also present in the TD-IWAE models, meaning that this characteristic is steadily learnt by the top-down hierarchical models, like in [9] and in [8]. The clusters are formulated in a similar way in every model, not showing any meaningful difference between the baseline network and the importance weighted versions.



(a) TDVAE (b) TD-IWAE, $k = 1$ (c) TD-IWAE, $k = 4$

(d) TD-IWAE, $k = 10$ (e) TD-IWAE, $k = 50$

Figure 5.12: Texture families form clusters when plotting $\mathbf{z}_2$ in 2D. Colours indicate texture class labels, while individual points are the marginalized $\mathbf{z}_2$ latents of the texture test images

For the reason that texture families are clustered in the latent space of $\mathbf{z}_2$, it would be intriguing to explore how well the texture classes could be decoded from these latents. To investigate this, we fit a multinomial logistic regression on the $\mathbf{z}_2$ posterior means to predict the class labels of the observations.

To be more precise, the trained models were used to extract the means of $q_{\boldsymbol{\phi}}(\mathbf{z}_2|\mathbf{x})$ for each of the image patches in the train and test dataset, and then these were used as input to the classifier alongside with the corresponding class labels. The logistic regression was trained on the means belonging to the training patches, and then the fitted model was used for prediction on the means belonging to the test patches.

The training and evaluation methodology for the logistic regression was repeated 5 times, and the obtained accuracies and confusion matrices were averaged to decrease side effects coming from randomization. The above procedure was done for all the hierarchical models and the results are shown in Figure 5.13 and in the Table 5.1.

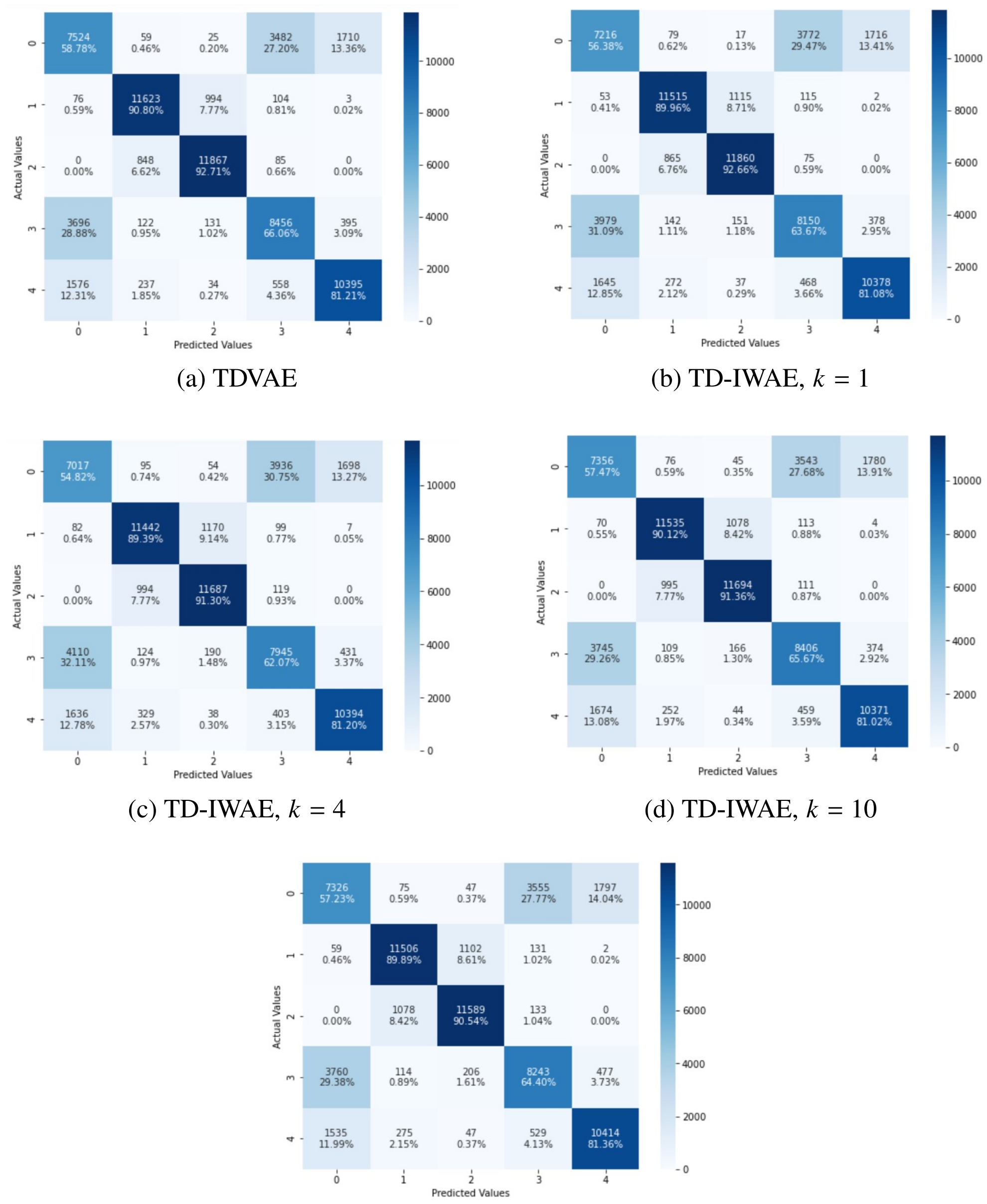
Generally, it can be stated for all the generative models that from the $\mathbf{z}_2$ posterior means the texture classes can be decoded quite confidently as every accuracy is at least 0.75. Due to the fact that in the training and test sets the ratio of the class labels are balanced, accuracy presents valid performance results. From the confusion matrices it can be said that the prediction for class labels 1, 2 and 4 are pretty good, in contrast to labels 0 and 3.

When comparing the importance weighted models with the baseline TDVAE, no significant difference can be seen in the confusion matrices. Regarding the accuracy scores, the one related to TDVAE posterior means seems to have the highest one. Additionally, concerning the accuracy scores in connection with TD-IWAE models, increasing the number of samples does not imply any gradual improvement or deterioration. The predictor fitted on the posterior means obtained from TD-IWAE with $k = 4$ had the lowest accuracy, while taking into consideration the TD-

IWAEs with more samples have higher scores as well as the $k = 1$ case which should be more or less identical with the baseline model.

Model	Average accuracy
TDVAE	0.779
TD-IWAE, $k = 1$	0.767
TD-IWAE, $k = 4$	0.757
TD-IWAE, $k = 10$	0.771
TD-IWAE, $k = 50$	0.766

Table 5.1: Accuracies of decoding texture families with logistic regression from $\mathbf{z}_2$ posterior means



(a) TDVAE

(b) TD-IWAE, $k = 1$

(c) TD-IWAE, $k = 4$

(d) TD-IWAE, $k = 10$

(e) TD-IWAE, $k = 50$

Figure 5.13: Confusion matrices of decoding texture families with logistic regression from $\mathbf{z}_2$ posterior means

An additional interesting question arises after the above results: all of the dimensions contribute equally to the texture family classification or some of them are encoding more information about texture types?

To investigate the above question, the logistic regression was refitted by gradually adding a new dimension to its training dataset in every new fit. The experiment shows that most of the coordinates of the posterior means do not contribute to the finally accuracy significantly, but very few dimension has a huge impact on the increase in accuracy, matching with the findings in [9], [8]. After collecting these texture-selective coordinates, the classification was performed again by separating the dataset into two parts: to the posterior means considered only on the texture selective dimensions, and on the non-selective ones. Logistic regression was fitted 5 times on both datasets, and the results were averaged over the fits, to exclude randomization effects. The accuracies can be found in Table 5.3, along with the texture-selective dimensions listed in Table 5.2.

The results clearly show that the texture information is only encoded in few dimensions in all models, as these coordinates are enough to predict the texture classes with at least 0.75 accuracy, while using the other dimensions in themselves the accuracy is close to random guessing. The TD-IWAE models did not beat the vanilla TDVAE, all of them have lower, but close to baseline performance. Regarding the number of texture-selective dimensions, TD-IWAE with $k = 1$ and $k = 50$ has the same number of selective coordinates as TDVAE, while the models with $k = 4$, $k = 10$ have more such dimensions, meaning that the information about texture characteristics is spread over more coordinates. Additionally note that the texture-selective dimensions approximately matches the dimensions possessing smaller standard deviation of posterior means and smaller mean of the posterior standard deviations in the activity plots.

All things considered, the desired properties in the higher latent space of the top-down hierarchical network was present in all models, however, the results in the importance weighted models were quite similar to the results in the TDVAE, even with a slight performance degradation in the texture family decoding accuracy experiments.

Model	Texture-selective dimensions
TDVAE	[0, 4, 5, 7]
TD-IWAE, $k = 1$	[0, 3, 5, 12]
TD-IWAE, $k = 4$	[0, 1, 2, 4, 14, 15]
TD-IWAE, $k = 10$	[0, 1, 3, 5, 13]
TD-IWAE, $k = 50$	[0, 1, 5, 15]

Table 5.2: The texture-selective dimensions in $\mathbf{z}_2$ posterior means

	Average accuracy	
Models	Selective	Non-selective
TDVAE	0.775	0.270
TD-IWAE, $k = 1$	0.765	0.242
TD-IWAE, $k = 4$	0.754	0.242
TD-IWAE, $k = 10$	0.766	0.263
TD-IWAE, $k = 50$	0.764	0.273

Table 5.3: Accuracies of logistic regressions fitted on $\mathbf{z}_2$ means constrained on the texture-selective and non-selective dimensions

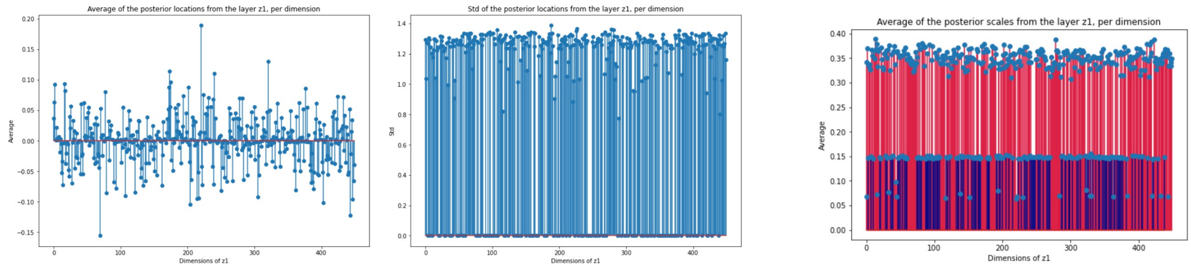
5.4 Examination of the latent space of $\mathbf{z}_1$

When studying the lower latent layer, we are interested in two things. Firstly, we would like to reproduce the properties of TDVAE presented in [9], [8], arising from the choices of specific architectural elements, and secondly investigating how importance weighting shapes the learnt representation. As it is stated in [9], [8], the top-down path, through the contextual prior $p(\mathbf{z}_1|\mathbf{z}_2)$, influences the posterior correlations in the level of $\mathbf{z}_1$, therefore we are particularly interested in the effects of importance weighting in these correlations.

The first step in the examination is to look at the activity of latent dimensions in our models. The latent space of $\mathbf{z}_1$ possesses 450 dimensions, but not all of them are active, unlike it was in the case of $\mathbf{z}_2$. Again, the activity can be seen by looking at the location and scale parameter of the posteriors, taking their mean and standard deviation over the test set. Many of the dimensions have relatively high standard deviation for the location parameter, but there are some being completely 0. The coordinates that own standard deviation different from 0 are defined to be active. The different models have nearly the same number of active dimensions: the TDVAE has 317 active ones while every importance weighted model have 314-315 active dimensions.

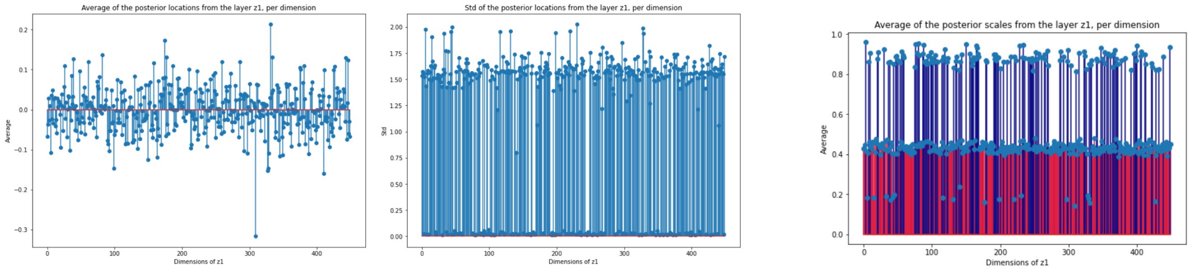
In the figure of the average scale parameters the dimensions are also isolated into two easily separable groups, and the ones which were considered active before are colored in red. This coloring help us indicate that the grouping of the dimensions regarding the scale parameter and the grouping corresponded to the std of location parameters are the same. This confirms that the active and non active dimensions can clearly be separated in all models.

Another observable phenomenon is that the active dimensions have their average posterior scale at around 0.4 in all models, while the magnitude of not active dimensions are different in the baseline TDVAE and in the importance weighted models.



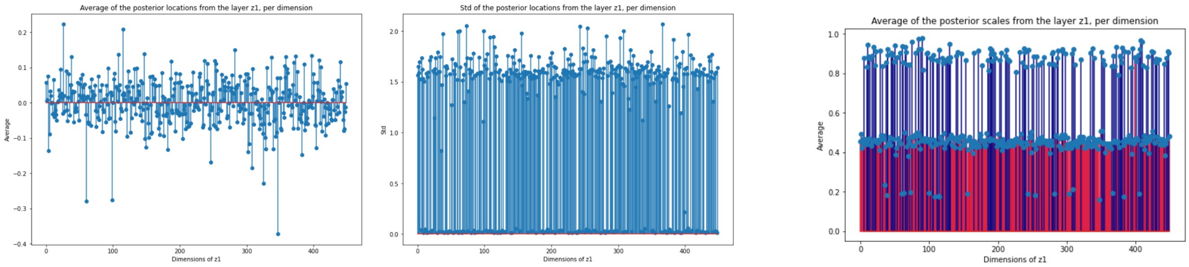
(a) (b)

Figure 5.14: Examination of the location and scale values of the posterior $q_\phi(\mathbf{z}_1|\mathbf{x})$ in TDVAE



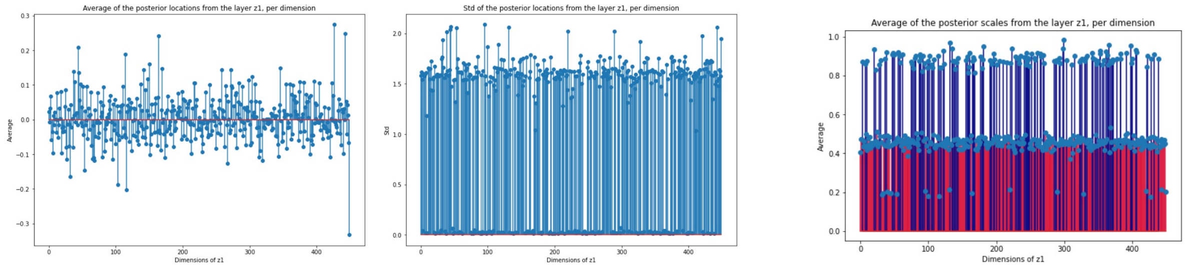
(a) (b)

Figure 5.15: Examination of the location and scale values of the posterior $q_\phi(\mathbf{z}_1|\mathbf{x})$ in TD-IWAE, $k = 1$



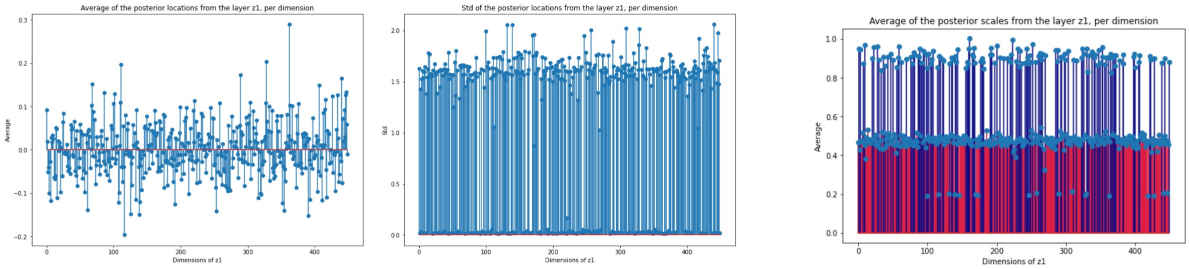
(a) (b)

Figure 5.16: Examination of the location and scale values of the posterior $q_\phi(\mathbf{z}_1|\mathbf{x})$ in TD-IWAE, $k = 4$



(a) (b)

Figure 5.17: Examination of the location and scale values of the posterior $q_\phi(\mathbf{z}_1|\mathbf{x})$ in TD-IWAE, $k = 10$



(a) (b)

Figure 5.18: Examination of the location and scale values of the posterior $q_\phi(\mathbf{z}_1|\mathbf{x})$ in TD-IWAE, $k = 50$

From now on, the analysis considers only the active dimensions in the models. The projective field of the model can be examined with latent traversal, which is a great tool to gain insight into what properties are encoded in the latent space. The traversal is performed in the following way: given a latent vector $\mathbf{z}_1$, one latent coordinate is changed by shifting it with the same constant value both by subtracting it and adding it to the latent coordinate, while the other dimensions kept fixed. The two new vectors are fed into the generative model, and the two resulted reconstructed images are subtracted from each other. The method indicates what sensitivities were encoded in the linear generative part of the model by observing the changes in the output image.

The obtained projective fields of the active units were mostly Gabor-like filters which are localized and oriented, such as in [9], [8]. The appearance of these linear filters is due to the linear relationship between $\mathbf{z}_1$ and $\hat{\mathbf{x}}$ as it is stated in [8]. In addition, some non-localized, more structured filters also emerged in all models. Examples of the learnt Gabor and non-Gabor filters are depicted in Figure 5.19. In all models, most of the active dimensions return Gabor-like filters, and they also learnt nearly the same number of localized filters: the TDVAE has 269 such filters, the TD-IWAE with $k = 1$, $k = 4$, $k = 10$, and $k = 50$ has 271, 270, 272, and 267 localized filters, respectively.

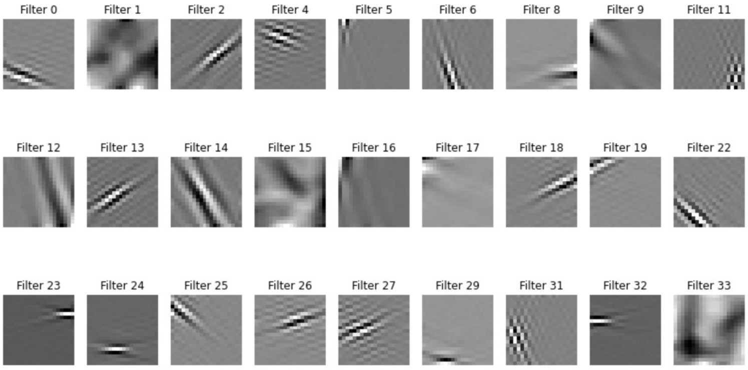


Figure 5.19: Example units of Gabor-like, and non-localized, more structured filters

Similarly to the examination of the posterior of $\mathbf{z}_2$, multinomial logistic regression was fit on the posterior locations, to investigate the decoding of texture families. Our expectation was that the texture classes can not be decoded properly from $\mathbf{z}_1$ locations, since texture-selectivity in thought to be encoded in $\mathbf{z}_2$.

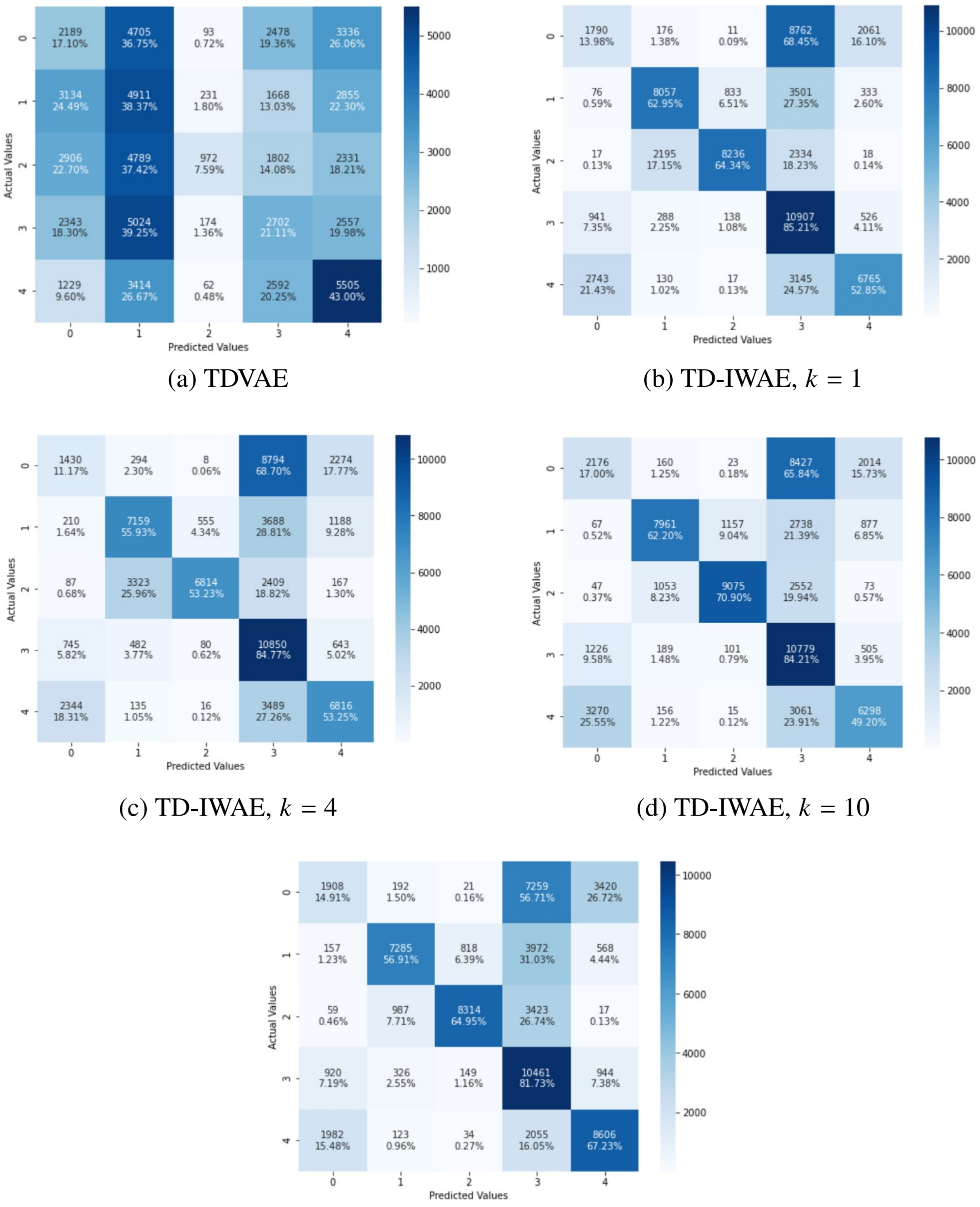
An individual logistic regression was built for all hierarchical models, these were fit on the labels and posterior locations corresponded to the training texture image patches, and were evaluated on the locations obtained from the test texture patches. For every top-down model, the classifier was fitted and evaluated 5 times, and the results were averaged to get rid of randomization effects. The accuracies measured on the test set are shown in Table 5.4 , and the corresponding confusion matrices can be seen in Figure 5.20.

The accuracy values are less than the accuracies of the logistic regressions trained on $\mathbf{z}_2$ posterior means, as expected. Also, the TDVAE has near chance classification scores while all the TD-IWAE models achieved at least 0.51 accuracy, which came as a surprise. It was the case even with TD-IWAE $k = 1$. There is even some tendency appear, that the with $k > 1$, the more samples we have in the importance weigthing, the higher this accuracy is. The phenomenon indicates that there are more texture-related information in the lower latent layer in the TD-IWAE models than in the lower latent space of vanilla TDVAE.

Building on this observation, it is worth taking a look at the $\mathbf{z}_1$ samples visualized in two dimension. These illustrations are generated in the same way as with $\mathbf{z}_2$ samples, and can be observed in Figure 5.21. The obvious difference with the plots of $\mathbf{z}_2$ samples is that in the lower latent layer the texture families do not form clusters at all. The clusters are not even present in the figures of importance weighted models, therefore this visualization does not provide an additional evidence that texture information is presenting in the $\mathbf{z}_1$ latent space.

Model	Accuracy
TDVAE	0.254
TD-IWAE, $k = 1$	0.558
TD-IWAE, $k = 4$	0.516
TD-IWAE, $k = 10$	0.567
TD-IWAE, $k = 50$	0.571

Table 5.4: Accuracies of decoding texture families with logistic regression from $\mathbf{z}_1$ posterior locations



(a) TDVAE

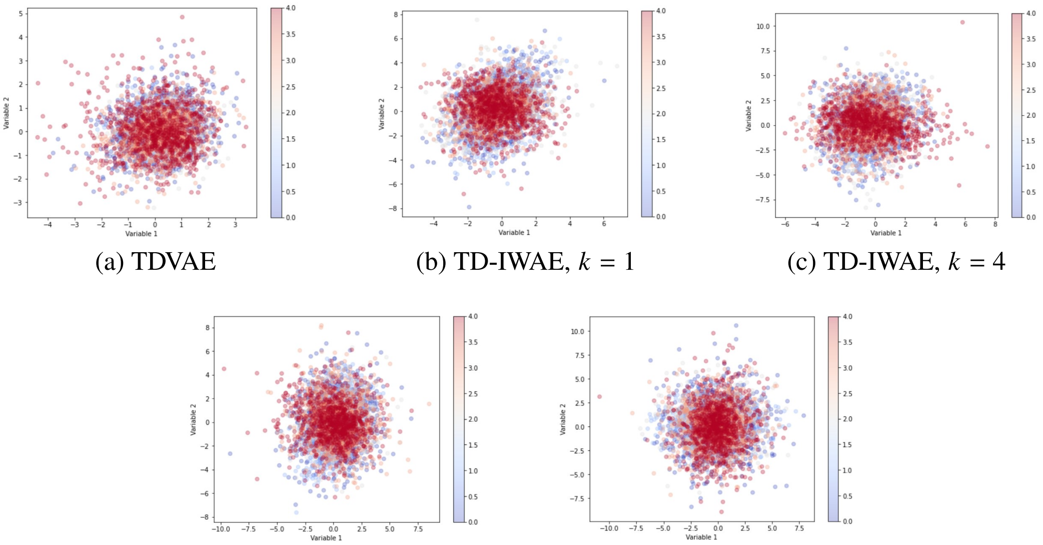
(b) TD-IWAE, $k = 1$

(c) TD-IWAE, $k = 4$

(d) TD-IWAE, $k = 10$

(e) TD-IWAE, $k = 50$

Figure 5.20: Confusion matrices of decoding texture families with logistic regression from $\mathbf{z}_1$ posterior locations



(a) TDVAE (b) TD-IWAE, $k = 1$ (c) TD-IWAE, $k = 4$

(d) TD-IWAE, $k = 10$ (e) TD-IWAE, $k = 50$

Figure 5.21: Plotting $\mathbf{z}_1$ posterior samples in 2D. Colours indicate texture families.

We examined further the phenomenon of better texture family classification in $\mathbf{z}_1$ provided by importance weighted models by fitting additional logistic regression predictors on $\mathbf{z}_1$ posterior locations, but this time only the active dimensions of the posterior parameter is used and these are separated into localized filters and noon-localized filters. The intuition behind splitting up the coordinates is that the active dimensions formulate only two types of filters, the Gabor-like ones, and the more abstract ones as the visualization showed, probably indicating that different types of information were captured in these two forms of filters. Moreover, surprisingly the number of the non-localized filters is more or less matches the number of dimensions in $\mathbf{z}_2$ as it can be seen in the Table 5.5.

For all hierarchical models two dataset was provided to the logistic regressions, one is for the active units returning localized filters and one is for the active units providing the abstract filters. For each model, on both datasets 5 fitting and evaluation of the predictors were performed and the results were averaged over the 5 fittings in the Table 5.6. Unlike in the case of $\mathbf{z}_2$, no clear distinction can be seen in the predictors' performance on the separated datasets. The classifiers fitted the localized filters obtained by importance weighted models seem to be slightly more powerful in decoding classes than the one fitted on locations coming from TDVAE, while it is totally the opposite with the classifiers fitted on the non-localized filters.

To sum up, the logistic regression trained on posterior locations coming from importance weighted models outperformed the ones fitted on posterior locations given by TDVAE, indicating that information about texture families were present more in the lower layer representation of these models than of TDVAE. However, no clear conclusion can be drawn about the role of different filters in improved prediction obtained by importance weighted models from this experiment.

5.4.1 Posterior correlations of $\mathbf{z}_1$

For the reason that both hierarchical VAE and importance weighting aims to enrich the latent spaces, studying the relationships of the latent dimensions through computing correlation values

Model	Non-localized filters
TDVAE	21
TD-IWAE, $k = 1$	18
TD-IWAE, $k = 4$	18
TD-IWAE, $k = 10$	18
TD-IWAE, $k = 50$	20

Table 5.5: Number of non-localized filters in $\mathbf{z}_1$ posterior locations

	Average accuracy	
Models	Localized	Non-localized
TDVAE	0.279	0.305
TD-IWAE, $k = 1$	0.311	0.275
TD-IWAE, $k = 4$	0.297	0.293
TD-IWAE, $k = 10$	0.315	0.282
TD-IWAE, $k = 50$	0.323	0.264

Table 5.6: Accuracies of logistic regressions fitted on $\mathbf{z}_1$ locations constrained on the localized filters and non-localized ones

would provide a great insight about the learnt representations. Also, the analysis will be carried out on individual patches besides considering several input patches and aggregating the results, since it was stated in [9] that contextual priors influence the posterior correlation values of $\mathbf{z}_1$ causing it to depend on the input patch.

To carry out an analysis for the posterior correlations, the test set of natural images was used, and we selected the 1000 highest contrast patches from this set to conduct the study on them.

Firstly, to have a full picture of the correlations between $\mathbf{z}_1$ dimensions, we took natural patches, and drew 1000 samples from the marginalized posterior $q_{\boldsymbol{\phi}}(\mathbf{z}_1|\mathbf{x})$ returned by the hierarchical models for the underlying image patch, and then the correlation coefficients were calculated between the latent dimensions over the drawn samples. As before, in case of importance weighted models, the samples are drawn according to the resampling method. Note that the correlation coefficients depend on the input image patch. Examples of such correlation matrices are shown in Figure 5.23. Note that the coefficients corresponding the cases where a unit is paired with itself, hence yielding a coefficient value of 1, are excluded in the analyses. The distributions of correlation values follow a normal distribution concentrated around 0 with heavy tails, meaning that among a lot of low absolute correlation values we can found some higher absolute ones in all models. For this image patch, these higher absolute values are around 0.4-0.5. Comparing the standard TDVAE and its importance weighted versions for this natural patch, the distribution of coefficients are similar, but in the TD-IWAE models some values stronger then the ones in TDVAE (around 0.5) can be found as well.

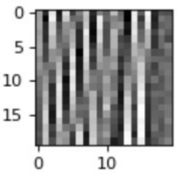
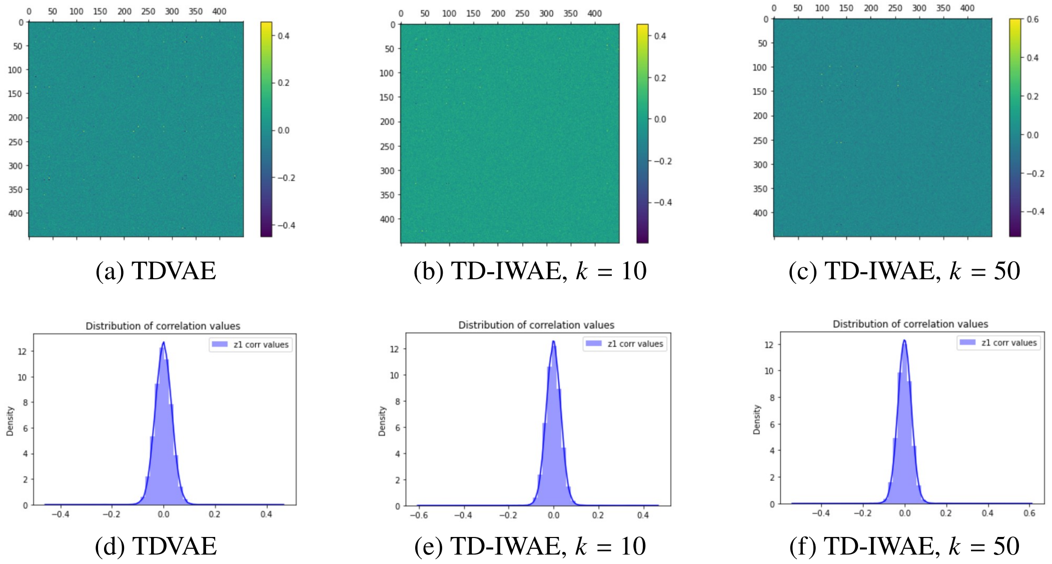


Figure 5.22: Natural image patch used for individual image posterior correlation analysis



(a) TDVAE (b) TD-IWAE, $k = 10$ (c) TD-IWAE, $k = 50$

(d) TDVAE (e) TD-IWAE, $k = 10$ (f) TD-IWAE, $k = 50$

Figure 5.23: Correlation matrices and distribution of correlation values of $\mathbf{z}_1$ on a natural patch

After gaining insight into the correlation matrices of the whole latent $\mathbf{z}_1$ obtained on some natural image patches, we seek to examine them in more details by investigating the correlations between some interesting dimensions and also by inspecting the strongest correlations values in general.

Firstly, we chose some dimensions from the active units which yielded Gabors with their centers located approximately in the middle of the patch, their sizes presented within some predefined range, and their spatial frequency and wavelength was higher than a predefined threshold. Examples of such Gabor-filters can be seen in Figure 5.24 along with the defined thresholds. The selection process were performed with the same thresholds in all the models.

The distribution of the absolute value of the correlation coefficients of the selected Gabor-coordinates were calculated for 500 image patches, and these are shown in Figure 5.25. From the results, we can claim that these units do not present meaningful correlations, most of the coefficients are almost 0, and only few have their absolute value between 0.1 and 0.2 in every model. This can be caused by the patch sizes which are relatively small compared to the size of the filters, as it can also be observed by looking at the examples of the selected filters in Figure 5.24.

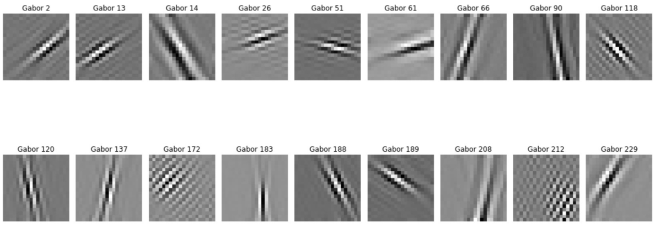
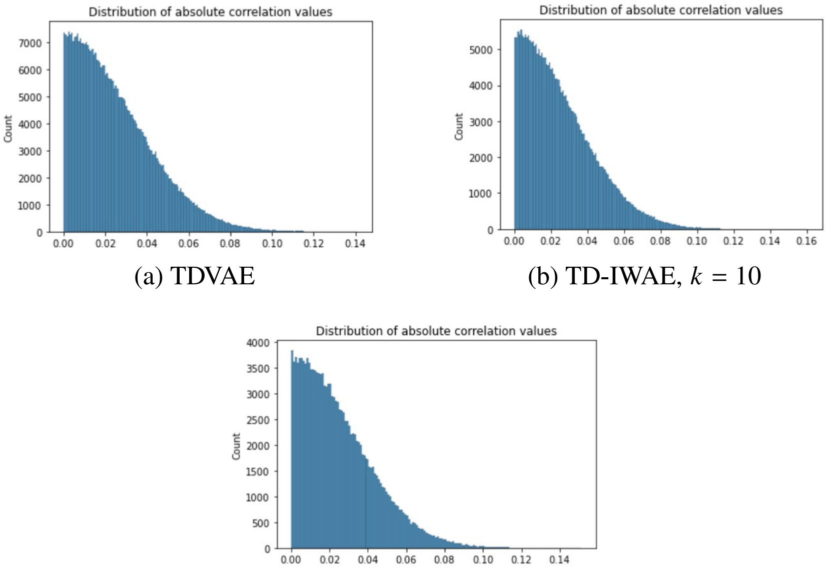


Figure 5.24: Example Gabor-filters selected according to predefined relative threshold values: with their center localized between the 6th and 14th pixels both vertically and horizontally, their scale falling between the values 3 and 6, their wavelengths being > 2, and their spatial frequency being > 0.18



(a) TDVAE

(b) TD-IWAE, $k = 10$

(c) TD-IWAE, $k = 50$

Figure 5.25: Distributions of absolute correlation values averaged over 500 image patch, between the selected Gabor-filters

Another way to examine the differences in correlation values between the models, we can simply investigate the highest correlations in absolute value. Firstly, we observed on the image patch shown previously how the 5 highest correlation values change between the models, as it can be seen in Figure 5.26. In this image, the TD-IWAE with $k = 50$ has the strongest correlations, beating TDVAE, while the model where we sample only once, and which should be more or less identical to the baseline model, has the lowest. Generally, no tendency can be seen in the direction of the highest correlations.

Secondly, we searched for the 10 coordinates which have the highest absolute correlation coefficients in the underlying image, and also plotted the coordinates yielding the 3 strongest among them in two dimension by marginalizing the $\mathbf{z}_1$ samples on these coordinate pairs. For the image patch presented above, the average and standard deviation of the 10 highest correlations are summarized in Table 5.7, which shows that for this input the importance weighted models seem to have less or the same correlation as the baseline model, except for the TD-IWAE, $k = 50$. The filters corresponding the units with the 3 strongest correlations are visualized in Figures 5.28, 5.29, 5.30 along with their distribution plots. Surprisingly, in all models the non-localized, more exotic filter pairs gave the strongest correlation relations. Also, the two-dimension plots corresponding to the examined unit pairs show that the distribution of these dimensions do not concentrate around 0, indicating the activation of these units.

	TDVAE	TD-IWAE, $k = 1$	TD-IWAE, $k = 4$	TD-IWAE, $k = 10$	TD-IWAE, $k = 50$
Average	0.439	0.371	0.445	0.401	0.516
Std	0.029	0.022	0.037	0.062	0.045

Table 5.7: The 10 highest absolute correlations averaged per model on an image patch

For the reason that the study of highest absolute correlation coefficients was performed only on one image patch, we would like to also extend this survey to more patches to get a clearer

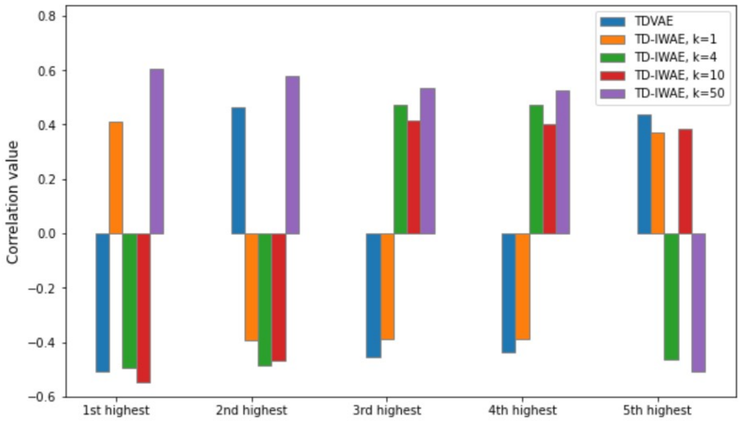


Figure 5.26: The 5 strongest correlations measured on our example image patch in every model

picture about these coordinates. Hence, we searched for the 5 highest correlations in absolute value on 500 image patches, and averaged the results in Figure 5.27. It is clearly visible that on average the importance weighted models do not introduce stronger correlation structure than the one found in TDVAE. Moreover, the TD-IWAE with $k = 1$ experienced a huge drop in its highest absolute correlations compared to the TDVAE highest ones. The plot also shows us that the standard deviations, indicated by error lines on top of the bars, are larger in the importance weighted versions than in the standard TDVAE.

In addition, in order to examine if stronger relations between posterior units are also strong in other image patches or not, we took 3 random image patches from the dataset, calculated the 5 highest absolute correlation values for each of them and then returned these values along with the filter pairs that produced them. After that for each of the 3 patches and coordinate pairs we examined what correlation coefficients these pairs give on another 500 patches, and then we averaged the results. To put it simply, we are interested in if these pairs produce consistently high absolute values for a bunch of other image patches as well or not.

The results of the consistency study are present in Table 5.8. For every coordinate pair, the mean and standard deviation of the 500 obtained correlation coefficients were computed, and we measured the ratio of the difference between this mean and the referred coefficient in absolute value, and the standard deviation of the 500 coefficients. This measure outlined a picture about how far away the referred coefficient from the sample mean relatively to the sample standard deviation for this coordinate pair. These ratios were averaged over coordinates and the 3 image patches and depicted in Table 5.8. One can observe that TDVAE seems to be more consistent for strong correlation values then the TD-IWAE models. Also, incorporating importance weighting altered this consistency since with $k = 1$ the ratio is much higher than with the baseline model. On the other hand, taking many samples provides some improvement as the consistency of TD-IWAE with $k = 50$ is better than with $k = 4$ and $k = 10$.

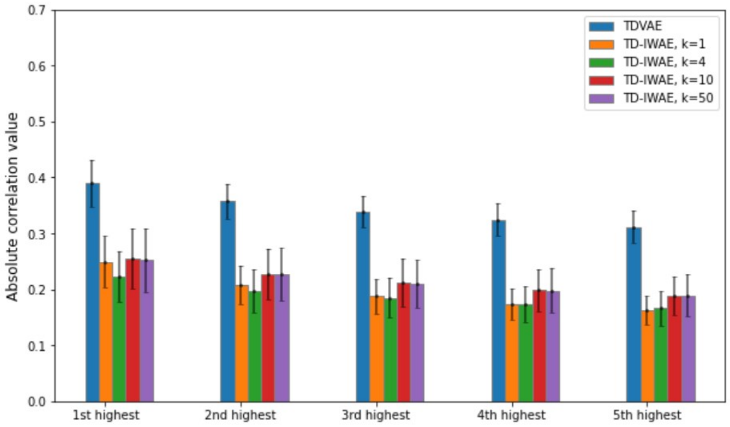
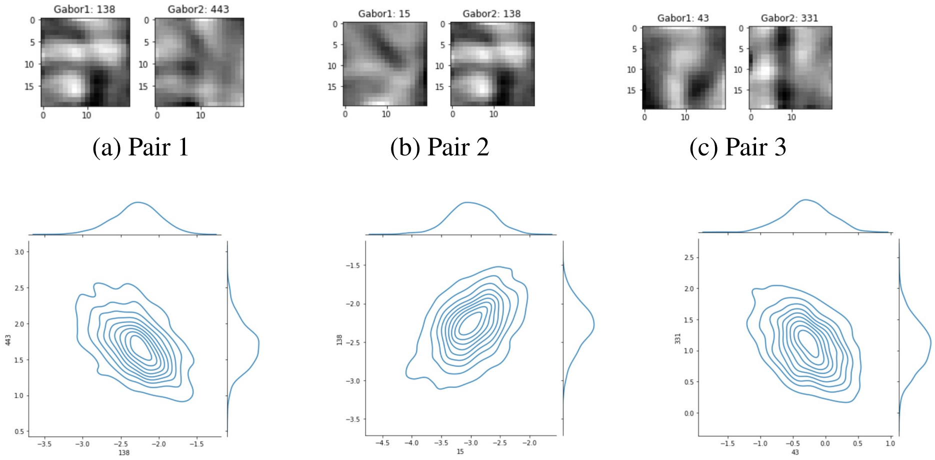


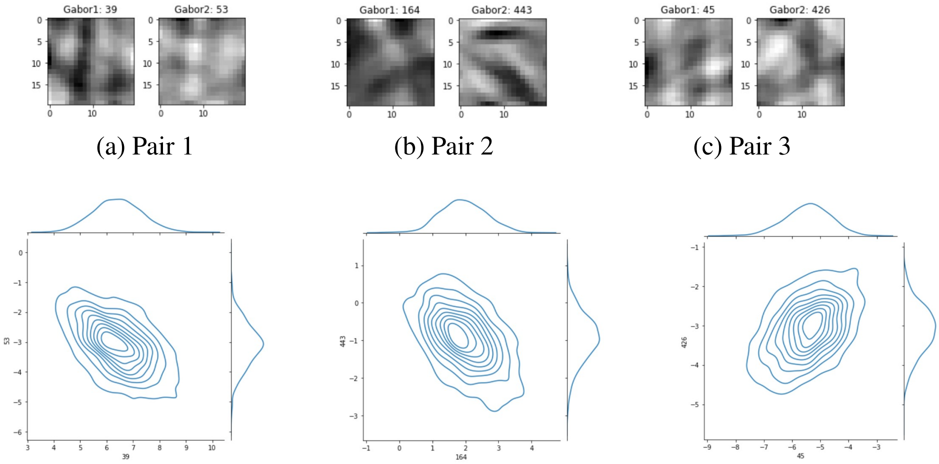
Figure 5.27: The 5 highest absolute correlation values averaged over 500 natural patches in every model. Height of the bars indicate mean of the measured correlations, while the standard deviation also depicted as error lines on top of the bars



(a) Pair 1 (b) Pair 2 (c) Pair 3

(d) Pair 1 (e) Pair 2 (f) Pair 3

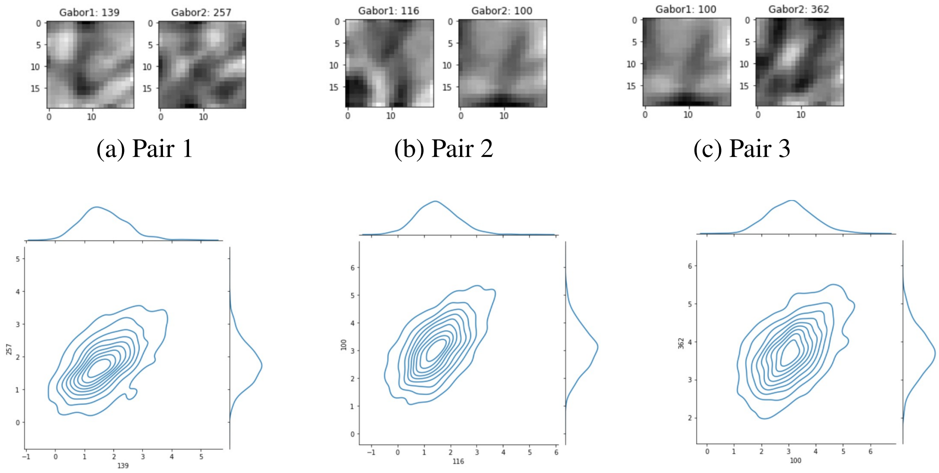
Figure 5.28: The Gabor pairs which provide the 3 strongest correlation values on an example natural patch, and their 2D distribution for TDVAE



(a) Pair 1 (b) Pair 2 (c) Pair 3

(d) Pair 1 (e) Pair 2 (f) Pair 3

Figure 5.29: The Gabor pairs which provide the 3 strongest correlation values on an example natural patch, and their 2D distribution for TDIWAE, $k = 10$



(a) Pair 1 (b) Pair 2 (c) Pair 3

(d) Pair 1 (e) Pair 2 (f) Pair 3

Figure 5.30: The Gabor pairs which provide the 3 strongest correlation values on an example natural patch, and their 2D distribution for TDIWAE, $k = 50$

TDVAE	TD-IWAE, $k = 1$	TD-IWAE, $k = 4$	TD-IWAE, $k = 10$	TD-IWAE, $k = 50$
0.8936	1.5453	1.8718	1.7489	1.5443

Table 5.8: The consistency of Gabor pairs with higher absolute correlation

To summarize the the study of the posterior correlations of $\mathbf{z}_1$, the importance weighted models did not introduce additional or stronger correlation relations in general. For individual natural image patches, we could observe that in TD-IWAE with $k = 50$ the mean of 10 strongest coefficients were higher than of TDVAE, as well as the individual 5 highest absolute coeffiecients, but considering multiple patches TDVAE had higher absolute values, these were more stable and also more consistent regarding the strong coefficients given by filter-pairs.

Chapter 6

Summary

The aim of this thesis was to to give an outline of Importance Weighted Variational Autoencoders [4], discuss their properties, and apply their methodology in a hierarchical variational autoencoder, TDVAE proposed in [9], [8]. Since TDVAE is a top-down hierarchical VAE with two latent layers, formulating an extension of the IWAE scheme is needed in order to adapt it to the TDVAE model. Both the baseline TDVAE and its generalized, importance-weighted version, TD-IWAE was discussed in detail. In addition, an attempt to generalize to TDVAE the properties of the reinterpretation of single latent layer IWAE models proposed in [7], [6] was also included.

The thesis is complemented by implementing both TDVAE and TD-IWAE, training them on a dataset consisted of natural image patches, and carrying out an analysis on the learnt representations. The single latent layer VAE enhanced with importance weighting was also implemented and examined, but in this thesis I focused only on my main contribution, the extended version of this model.

The TD-IWAE model was trained in four variations, each of them with different number of samples drawn from the variational posterior: with $k = 1$, $k = 4$, $k = 10$, and $k = 50$. The evaluation of the trained models was done by measuring their average ELBO loss obtained on both the natural and texture image test set. It was observed that the importance weighted models achieved lower loss value than TDVAE on both test data, and as the training was performed with higher k, the lower this averaged loss value was, which matched with our expectations.

Owing to the appealing properties of IWAE models, with which we wished to obtain more expressive variational posteriors, it was particularly interesting to study how the importance weighting shaped the learnt representations. The analysis was conducted on both latent layers, building on the examinations carried out in [9], [8] and the results were compared between the TDVAE and TD-IWAE models.

In the higher latent layer the focus of interest was the information encoded in $\mathbf{z}_2$ about textures. Visualizing these higher layer representations of input patches revealed that texture classes form clusters. No qualitative difference could be seen between the different models. Also, logistic regression was fit to classify the input images into the five texture families based on posterior means. The predictors of fitting on data obtained from models with lower k were slightly worse than the ones trained on TDVAE posterior means, but with $k = 50$ the results were again close to the performance on data given by TDVAE.

The lower latent layer was also examined from the aspect of texture encoding, and surprisingly the classifiers which were fit on $\mathbf{z}_1$ posterior locations coming from importance weighted models performed better then the ones fitted on TDVAE posterior locations. The analysis was extended by other investigations to find out the reason behind this phenomenon but no clear answer was found. Furthermore, the posterior correlations were also studied. We concluded that while individual natural patches can be found where the importance weighted models yield some stronger correlation values, these models could not provide stronger relations when averaging over several patches.

Future research directions include the investigation of texture-related information captured by the lower latent layer in TD-IWAE models, as the reason for why fitting multinomial logistic regression on posterior locations of importance weighted models achieved unexpectedly high prediction accuracy is still unanswered. Related questions also emerge about the relationship of the latent layer of $\mathbf{z}_2$ and $\mathbf{z}_1$, and its different nature in TDVAE and TD-IWAE, because we have seen that information about texture classes are quite reliably present in the higher latent layer in all models. Studying these properties would provide us a deeper understanding of the differences between the learnt representations in TDVAE and TD-IWAE.

Bibliography

[1] Philip Bachman and Doina Precup. “Training deep generative models: Variations on a theme”. In: *NIPS Approximate Inference Workshop*. 2015.

[2] Yoshua Bengio, Aaron Courville, and Pascal Vincent. *Representation Learning: A Review and New Perspectives*. 2014. arXiv: `1206.5538 [cs.LG]`.

[3] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006. ISBN: 0387310738.

[4] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. *Importance Weighted Autoencoders*. 2016. arXiv: `1509.00519 [cs.LG]`.

[5] Rewon Child. *Very Deep VAEs Generalize Autoregressive Models and Can Outperform Them on Images*. 2021. arXiv: `2011.10650 [cs.LG]`.

[6] Chris Cremer. “Approximate Inference in Variational Autoencoders”. PhD thesis. University of Toronto, 2020.

[7] Chris Cremer, Quaid Morris, and David Duvenaud. “Reinterpreting importance-weighted autoencoders”. In: *arXiv preprint arXiv:1704.02916* (2017).

[8] Ferenc Csikor, Balazs Meszena, and Gergo Orban. “Top-down perceptual inference shaping the activity of early visual cortex”. In: *bioRxiv* (2023), pp. 2023–11.

[9] Ferenc Csikor et al. *Top-down inference in an early visual cortex inspired hierarchical Variational Autoencoder*. 2022. arXiv: `2206.00436 [q-bio.NC]`.

[10] Hien Dang et al. *Beyond Vanilla Variational Autoencoders: Detecting Posterior Collapse in Conditional and Hierarchical Variational Autoencoders*. 2024. arXiv: `2306.05023 [stat.ML]`.

[11] Jeremy Freeman et al. “A functional and perceptual signature of the second visual area in primates”. In: *Nature neuroscience* 16.7 (2013), pp. 974–981.

[12] Victor Geadah et al. “Sparse-coding variational auto-encoders”. In: *BioRxiv* (2018), p. 399246.

[13] Charles D Gilbert and Wu Li. “Top-down influences on visual processing”. In: *Nature Reviews Neuroscience* 14.5 (2013), pp. 350–363.

[14] GM Harshvardhan et al. “A comprehensive survey and analysis of generative models in machine learning”. In: *Computer Science Review* 38 (2020), p. 100285.

[15] Matthias Hennig and Theoklitos Amvrosiadis. *Neural Computation - Practical 5: Simple and Complex cells in visual cortex*. `https://www.inf.ed.ac.uk/teaching/courses/nc/NClab5.pdf`. Oct. 2018.

[16] Trevor Huff, Navid Mahabadi, and Prasanna Tadi. "Neuroanatomy, visual cortex". In: (2018).

[17] Diederik P Kingma and Max Welling. *Auto-Encoding Variational Bayes*. 2022. arXiv: `1312.6114 [stat.ML]`.

[18] Diederik P. Kingma and Max Welling. "An Introduction to Variational Autoencoders". In: *Foundations and Trends® in Machine Learning* 12.4 (2019), pp. 307–392. issn: 1935-8245. doi: `10.1561/2200000056`. url: `http://dx.doi.org/10.1561/2200000056`.

[19] Hung Le and Ali Borji. "What are the receptive, effective receptive, and projective fields of neurons in convolutional neural networks?" In: *arXiv preprint arXiv:1705.07049* (2017).

[20] S. R. Lehky. "Projective field". In: *Scholarpedia* 7.10 (2012). revision #137419, p. 10114. doi: `10.4249/scholarpedia.10114`.

[21] Calvin Luo. *Understanding Diffusion Models: A Unified Perspective*. 2022. arXiv: `2208.11970 [cs.LG]`.

[22] Wenjie Luo et al. "Understanding the effective receptive field in deep convolutional neural networks". In: *Advances in neural information processing systems* 29 (2016).

[23] Lars Maaløe et al. *BIVA: A Very Deep Hierarchy of Latent Variables for Generative Modeling*. 2019. arXiv: `1902.02102 [stat.ML]`.

[24] David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[25] Bruno A Olshausen and David J Field. "Emergence of simple-cell receptive field properties by learning a sparse code for natural images". In: *Nature* 381.6583 (1996), pp. 607–609.

[26] Javier Portilla and Eero P Simoncelli. "A parametric texture model based on joint statistics of complex wavelet coefficients". In: *International journal of computer vision* 40 (2000), pp. 49–70.

[27] V Shiv Naga Prasad and Justin Domke. "Gabor filter visualization". In: *J. Atmos. Sci* 13 (2005), p. 2005.

[28] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. "Stochastic Back-propagation and Variational Inference in Deep Latent Gaussian Models". In: ().

[29] Davide Scaramuzza. *Sparse Codes for Natural Images*. Tech. rep. ETH Zurich, 2005.

[30] Øivind Skare, Erik Bølviken, and Lars Holden. "Improved samping-importance resampling and reduced bias importance sampling". In: *Scandinavian Journal of Statistics* 2003 (2003), pp. 719–738.

[31] Casper Kaae Sønderby et al. *Ladder Variational Autoencoders*. 2016. arXiv: `1602.02282 [stat.ML]`.

[32] Arash Vahdat and Jan Kautz. *NVAE: A Deep Hierarchical Variational Autoencoder*. 2021. arXiv: `2007.03898 [stat.ML]`.

[33] J Hans Van Hateren and Arjen van der Schaaf. "Independent component filters of natural images compared with simple cells in primary visual cortex". In: *Proceedings of the Royal Society of London. Series B: Biological Sciences* 265.1394 (1998), pp. 359–366.

[34] Alan Yuille and Daniel Kersten. "Vision as Bayesian inference: analysis by synthesis?" In: *Trends in cognitive sciences* 10.7 (2006), pp. 301–308.

[35] Corey M Ziemba et al. “Selectivity and tolerance for visual texture in macaque V2”. In: *Proceedings of the National Academy of Sciences* 113.22 (2016), E3140–E3149.