DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT DRESDEN

Master’s Thesis in Informatics: Computational Modeling and Simulation

Analyzing Generative Factors of Functional Connectivity with Variational Autoencoder

Author:	Maksim Zhdanov
Supervisor:	Dr. Nico Hoffmann
Advisor:	Dr. Saskia Steinmann
Submission Date:	06.02.2022

I confirm that this master’s thesis is my own work and I have documented all sources and material used.

Dresden, 06.02.2022 Maksim Zhdanov

Abstract

Diagnosing neurological diseases is a challenging process hindered by the immense complexity of the human brain. Neurological examination generally consists of obtaining data from different modalities and recognising disease-indicated patterns in it. Every modality, however, provides a clinician with high-dimensional and often stochastic data that can be difficult to analyze for a human and make decisions regarding a patient's status. It encourages the application of machine learning algorithms that are able to operate with large volumes of data and efficiently perform classification.

Generally, the highly desired properties of a machine learning model deployed in a medical domain are robustness and interpretability. In the last decade, many algorithms were proposed to assist clinicians in recognising disease-related patterns on medical images and participating in the decision-making process. Those models were claimed to match or even surpass a human expert yet they generally underperformed when applied in the clinical environment to real-world data. It was due to their failure to learn causal relations from data instead relying on statistical associations evident in a large amount of data. Besides, interpretability is often omitted from the consideration which does not allow one to gain insights from data and hence improve knowledge about the system of interest. Combined with poor generalization, it leads to the rare application of machine learning in medicine.

We address the aforementioned problems by using the framework of generative modeling to develop an algorithm that is able to recover generative factors from data as well as use them for differentiating between multiple mental disorders. We demonstrate that learning such factors is equivalent to extracting causal mechanisms from data which is a prerequisite for robust classification. Since malfunctions are known to affect the neural activity of the brain, we incorporate them into the model as a supervision bias. It allows us to learn disorder-related causal mechanisms which can be further explored and possibly lead to gaining new knowledge about those diseases. The method demonstrates excellent classification performance and learns disease-related generative factors that are consistent with current domain knowledge.

Contents

Abstract iii

1. Introduction 1

1.1. The concept of causality 3

1.2. Causality for biomarkers search 4

1.3. Contributions 5

2. Probabilistic Modeling 7

2.1. Approximating a probability distribution 7

2.2. Graphical representation of probabilities 9

2.3. Bayesian networks 9

2.3.1. Learning in Bayesian networks 10

2.4. Latent variable models 11

2.4.1. Learning in latent variable models 12

2.5. Variational Autoencoder 12

2.5.1. Variational inference 12

2.5.2. The evidence lower bound 13

2.5.3. Auto-encoding Variational Bayes 14

2.5.4. Reparameterization trick 14

2.5.5. Gradient of ELBO 15

2.5.6. Variational autoencoder 15

2.5.7. Parameterizing distributions in VAE 16

2.5.8. Learning faces with VAE 17

2.6. Representation learning with VAE 18

2.6.1. Independent causal mechanisms 18

2.6.2. Learning a disentangled representation 19

2.6.3. Supervision bias 20

2.6.4. Capturing label characteristics in VAEs 21

3. Conditional Graph Variational Autoencoder 24

3.1. Introduction 24

3.2. Motivation 26

3.3. Conditional Graph Variational Auto-encoder 28
3.3.1. Model objective 29
3.3.2. Introducing β-regularization 30
3.3.3. Computational details 31
3.4. Exploring generative factors 32

4. Experimental Results 33
4.1. Experimental details 33
4.1.1. Parametrization of conditional distributions 33
4.1.2. Experimental data 34
4.1.3. Recovering functional connectivity 35
4.1.4. Optimization details 36
4.2. Results & Discussion 36
4.2.1. Classification of labels 36
4.2.2. The effect of β-regularization on latent space 37
4.2.3. Reconstruction from latent space 38
4.2.4. Most likely functional networks 39
4.2.5. Exploring generative factors of data 40
4.2.6. Limitations of the approach 42

5. Conclusion 43

A. Appendix 44
A.1. Notation 44
A.2. Derivation of ELBO 44
A.3. Model architecture details 45

B. Additional Results 46
B.1. Reconstruction ability 46
B.2. Intervening on labels 46

List of Figures 49

List of Tables 51

Bibliography 52

Acknowledgments 56

1. Introduction

It has been a while since computers started to be applied to fuel scientific research. They effectively managed to assist humans in the analysis of experimental data extracting meaningful patterns and insights that led to breakthroughs in multiple fields such as neuroscience [1], astrophysics [2] and genomics [3]. Eventually, computers became essential in scientific research due to the immense amount of data that is produced along the way - so large that human intelligence is not able to keep up with it anymore. Consequently, the need to operate with that information fueled the development of the new kind of intelligence - artificial intelligence - that would be able to utilize computational power to produce new knowledge. One subfield of artificial intelligence that was especially favoured by the new era of Big Data is machine learning which is designed to learn from experience by itself. Biomedicine has particularly benefited from the technologies. Medical institutions are constantly aggregating massive volumes of information which is produced in the form of high-resolution images, patients' records, DNA sequences and so on. It has created a fertile ground for the application of artificial intelligence which has gained extraordinary results in pattern recognition based fields like pathology and radiology [4]. Besides, the digitization of medicine accelerated the search for disease-related biomarkers which has the potential to drastically improve the quality of diagnosis and treatment.

The reason for the success is well summarized by neuroscientist Robert Darnell who once said [5]: "We can only do so much as biologists to show what underlies diseases like autism. The power of machines to ask a trillion questions where a scientist can ask just ten is a game-changer." However, this raises the concern as to what questions a computer is actually asking and how its answers to those questions can improve our understanding of real-world phenomena. To put it less abstractly, by a computer we understand a set of algorithms that a machine executes to solve a particular problem. Furthermore, by questions, we mean manipulations with input data that those algorithms perform while solving the problem. For example, we might use linear regression to quantify the relationship between the chance to have heart disease and multiple factors such as blood sugar, sex, age and so on. In this case, asking questions would correspond to estimating parameters of the linear model from each instance in a data set.

Most currently deployed models in machine learning are essentially pattern recognition algorithms trained on large scale data sets. The latter is often a prerequisite since those models aim at learning statistical associations that are not evident when the size of a data set is small. Those models are statistical at heart which means that asking "questions" by those "computers" boils down to simply inferring statistical dependence structures in data. Due to their statistical nature, the algorithms may rely on spurious associations instead of learning underlying causal mechanisms [6]. Hence, such models often underperform when dealing with unseen out-of-distribution data [7]. If experimental conditions are different from the original setting it could lead to inaccurate predictions with sometimes life-threatening consequences (e.g. misdiagnosis). Those limitations, namely poor robustness and out-of-distribution generalization, are key factors why artificial intelligence is not adapted in medicine as widely as in other data-heavy areas such as entertainment or marketing. Indeed, undiagnosed heart failure is much more dangerous than a wrongly advertised brand-new toothbrush.

Opposed to statistical models, causal models tend to acquire much more robust knowledge about a system (e.g. [8]). They seek a modular representation of a probability distribution corresponding to a real-world process. Those modules are essentially (coarse-grained) physical mechanisms whose robustness is guaranteed by laws of nature if defined correctly. Machine learning algorithms that incorporate causality tend to generalize better facing a new environment which is crucial for application in biomedical domain [6]. However, it is often the case that causal representation of a real-world process is unknown and thus has to be learned from data. This is a challenging task that was, in fact, shown to be fundamentally impossible to perform in an unsupervised manner without inductive biases on both model and data. At the same time, recovering causal mechanisms from data is truly fruitful as it allows one to ask questions about a modeled system that go beyond plain statistical correlations. Particularly, one can examine the outcome of an intervention, i.e. consequence of constraining these mechanisms. For example, one might wonder if changing the type of treatment would increase the probability of successful recovery for a patient.

One way to learn a causal model is via probabilistic modeling - a branch of machine learning that learns a joint probability distribution from data as a model of phenomena. It is capable of capturing the inherent uncertainty of real-world processes which makes probabilistic models preferable when dealing with stochastic data. Probabilistic graphical models are especially suitable for the task of learning a causal representation since a causal model can be represented as a graph. Indeed, incorporating causality applies constraints on a graphical representation such as independence of root variables and manipulability of causal mechanisms [6]. An approach that fulfils those requirements and can be effectively applied to large scale data sets is variational auto-encoder [9]. We consider probabilistic graphical models as well as variational autoencoders in Chapter 2. In the case when an observation comes with a set of its labels, we can utilize them as an inductive bias, namely supervision bias, to partially overcome the problem of uniqueness of a causal representation. Besides, linking certain generative factors to labels makes it possible to investigate causal relations between data and those labels. It gives a researcher an appealing opportunity to use generative modeling to perform interventions and investigate the system of interest. Inspired by the recently developed algorithm CCVAE [10], we demonstrate how such investigation can be done on a data set from neuroscience in Chapter 4.

We are particularly interested in tackling problems related to the mental health domain. The human brain is arguably one of the most complex structures in the universe whose principles of work were of great inspiration for the field of artificial intelligence. However, we are still far away from the complete understanding of the organ especially when its functionality is altered by a malfunction. The standard approach for neuroscientists to explore brain structures is via analyzing functional connectivity networks [11]. Those networks are constructed using brain imaging techniques such as EEG of fMRI and correspond to brain regions that demonstrate co-activation while performing a function. A network-like data is also an inductive bias that we incorporate in a model representing brain imaging data as graphs instead of images. It leads to the CCVAE model being modified to allow an application to data defined on a graph domain that we discuss in Chapter 3. We further apply this model to brain imaging data to investigate generative factors related to certain mental disorders. Analysis of those factors might reveal insights about causal mechanisms behind those disorders and potentially improve our understanding of mental health problems. We outline the contribution of the thesis in Section 1.3.

1.1. The concept of causality

For human beings, the causal way of thinking is quite natural. The power to ask "Why?" in response to a certain phenomenon has driven numerous amount of scientific discoveries. Understanding cause and effect relationships led to enormous progress in all fields of our activity and it is generally a key for developing knowledge about a system. However, before understanding always goes learning whether it is direct exploring the system of interest or indirect gaining insights about the physical laws governing nature. In any case, learning appears to be an essential step on the way to understanding causal relations.

While inferring cause-effect thinking is hard-wired in human intelligence, artificial intelligence is not developed enough to figure out why things happen. A lack of causal understanding leads to the general underperforming of AI systems when deployed in decision-making areas such as medicine, self-driving cars or banking [7]. The reason is that modern machine learning algorithms use statistical associations to learn from data instead of so-called causal reasoning. The approach is useful under certain conditions (such as independent and identically distributed data points) but does not allow to infer cause-effect relations. It is also vulnerable to statistical pitfalls such as the presence of spurious correlations in data. For example, there is a 99.79% correlation between the US science budget and the number of suicides by hanging [12]. It is obvious for a human that those variables are not causally related, however, a machine can decide to account for the budget to predict the count of suicides in the future.

Additionally, opposed to correlations, causal relations are not symmetric, meaning that if random variable X is a cause of random variable Y, the change of Y will not affect the value of X [13]. This difference can be depicted as follows:

$X \leftrightarrow Y$	$X \rightarrow Y$
Correlation between X and Y	Causal relation between X (cause) and Y

When we say symmetric, we imply that a statistical relationship can be written as an equation where a left-hand side and a right-hand side are connected via an equality sign that is symmetric. For example, in a linear case:

$$Y = A\,X + B$$

where the left-hand side corresponds to an effect and the right-hand side denotes predictor variables. From a machine's point of view, X plays a role of a cause and Y is an effect. However, it is possible to rewrite this equation such that

$$X = A^{-1}\,(Y - B)$$

where X would be an effect caused by Y. Therefore, causal relationships cannot be learned from data by calculating correlations due to their fundamental asymmetry.

Overall, the problem of causal reasoning, i.e. the ability to infer causes from observed phenomena [13], is crucial for artificial intelligence. Incorporating causality into a machine learning algorithm will potentially allow learning causal relations from data which will, in turn, significantly improve robustness and generalization of the algorithm to unseen data.

1.2. Causality for biomarkers search

Causal relations have always been of paramount importance in the field of medicine, where the main task for a doctor is to find a disease that causes a patient's symptoms.

As large volumes of multi-level information in the domain become accessible, machine learning algorithms were applied to derive new insights from that data. A relevant task in machine learning is to search for features in a patient's data called biomarkers that are also causally related to a disease. Therefore, a disease variable would be a confounder for both the features and symptoms, i.e. it would be causally related to both of them making features and symptoms statistically related. This search is important since symptoms are often subjective, they might not be apparent or spoken, e.g. in the mental health area. Meanwhile, biomarkers are meant to be highly reproducible and robust indicators of a disease and lead to diagnosis being more quantitative.

A classic approach for the search of biomarkers is to deploy a classifier that approximates a function f that maps a set of observed variables (e.g. gene expression data) X to a set of labels (e.g. health status) Y: $Y = f(X)$. The classifier is accompanied by a feature selection algorithm that filter out noise variable or non-informative features from the set X. Overall, many classification algorithms have been proposed in recent years claiming expert-level or above performance (e.g. [14]. As a rule, those models are trained on a sufficiently large amount of i.i.d. (independent and identically distributed) data samples and tested on data from the same distribution. Consequently, they generalized poorly to unseen data when clinical settings were different from experimental thus violating the i.i.d. assumption. Trustworthiness and robustness are, however, the key properties in decision-making areas such as medicine since inaccurate predictions can lead to dangerous consequences for people. The reason for the failure is the inability of machine learning algorithms to discover causal relations from data and instead identifying statistical correlations between elements of X and elements of Y. This can lead to relying on spurious associations or confounding factors when making predictions. For example, DeGrave et al. [8] demonstrated that deep learning algorithms predicting COVID-19 from chest X-ray images ignore medically relevant features and instead utilize text markers on scans - a feature that is correlated with a patient's status only due to the data collection process. Indeed text markers cannot be used as biomarkers, hence algorithms that discover causal relations have to be deployed to indicate robust disease-related features in data.

1.3. Contributions

In the thesis, we address the problem of learning causal mechanisms from data and present a representation learning algorithm that

a) learns a probabilistic model in the form of a disentangled representation;

b) incorporates supervised bias into the model, i.e. accounts for label information;

c) allows for interventions, e.g. varying learned generative factors;

d) can be applied to data defined on a graph domain.

Our model relies on recently developed characteristics capturing VAE that itself fulfils the first three points from the list above. Thus, the main contribution of the thesis is the adaptation of the algorithm to a graph domain (see Chapter 3). This is motivated by problems of neuroscience and brain imaging where graph data is the dominating form of data.

We further apply the model to electrical activity data taken from 3 groups of people: healthy people, patients suffering from schizophrenia with and without auditory verbal hallucinations. We learn generative factors of data related to a diagnosis and the presence of hallucinations to derive corresponding functional connectivity patterns (see Chapter 4).

2. Probabilistic Modeling

The immense complexity of the world as well as naturally limited cognitive abilities do not allow us to account for every possible variable while solving a real-world problem. It is opposed to Laplace's demon [15] - a vast intellect that knows the state of every object in the universe from which it is able to predict the future precisely as well as recover the past.

> *...for such an intellect nothing would be uncertain and the future just like the past would be present before its eyes.*
>
> Pierre Simon Laplace — A Philosophical Essay on Probabilities

Human beings are unfortunately far away from reaching the capabilities of the demon and usually possess limited knowledge of a system of interest. Nonetheless, we can convert our beliefs about the system into the language of mathematics to possibly extend our understanding as well as perform decision-making if those beliefs are sufficient. One effective way to do so while accounting for the inherent uncertainty is to use the framework of probabilistic modeling which will be the primary subject of the chapter. We provide a notation for this chapter and the following ones in Appendix A.1.

2.1. Approximating a probability distribution

As has been mentioned, we do not usually have complete knowledge about the system we want to model. Instead, what we usually have is a finite set $\mathbf{x}$ of observed variables from which we would like to deduce an outcome variable $\mathbf{y}$ which is also observed:

$$\mathbf{y} \approx f(\mathbf{x})$$

where f is an arbitrary complex deterministic function that we do not know. In mathematical modeling, the function f is further approximated with a chosen model f_θ with parameters θ such that for any observed $\mathbf{x}$:

$$f_\theta(\mathbf{x}) \approx f(\mathbf{x})$$

For example, we might develop a model that predicts if a person has a mental illness or not based on a psychiatrist assessment. Here, a set of tests that the doctor performs gives us the observed evidence $\mathbf{x}$ and $\mathbf{y}$ is a categorical variable indicating a brain malfunction.

The outcome of the assessment, however, is not deterministic since many factors might play a role such as the level of expertise of the doctor, quality of the medical facility and so on. Generally, it is often the case for real-world problems to involve some degree of uncertainty. Hence, it is more natural to handle this variability by considering a probability distribution instead of a deterministic relation:

$$p(\mathbf{x}, \mathbf{y})$$

Having the distribution, we could ask, for example, "what is the probability of the person to have schizophrenia?" and receive an answer as well as evaluate the confidence of the prediction.

However, the true distribution from which observed variables $\mathbf{x}$, $\mathbf{y}$ are sampled is unknown. In probabilistic modelling, we try to approximate it with a model distribution p_θ such that for any observed $\mathbf{x}$:

$$p_\theta(\mathbf{x}, \mathbf{y}) \approx p(\mathbf{x}, \mathbf{y})$$

Probabilistic modeling comprises 3 essential parts:

1. Representation

   At the representation step, one defines parameterization of the model p_θ by using prior knowledge about the system. As a rule, one usually seeks for an adequate yet simple and flexible expression for the probability distribution such that consequent fitting of the model to observed data is tractable.

2. Learning

   Given a dataset consisting of pairs $(\mathbf{x}, \mathbf{y})$, we search for parameters θ of the model distribution p_θ such that it fits the data as good as possible.

3. Inference

   At the inference step, we estimate modes of marginal or conditional distributions of interest. It can be seen as evaluating maximally probable values of model variables (including its parameters) given evidence, e.g. assignment of a subset of other variables.

Those parts are all connected: the choice of a particular representation specifies algorithms for learning of the resulting model, while inference is used as a subroutine during learning.

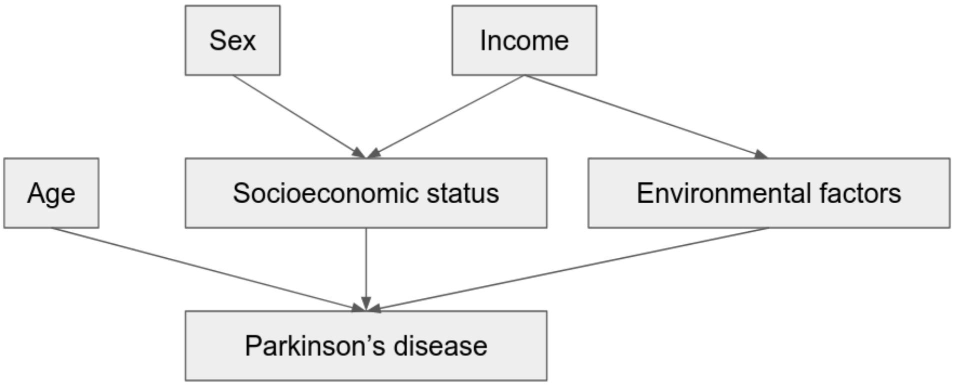


Figure 2.1.: Example of a graphical representation of a model for diagnosis of Parkinson's disease.

2.2. Graphical representation of probabilities

Probabilistic modeling can be combined with graph theory to describe joint probability distributions as graphical models, which can be beneficial for multiple reasons. First, the structure of a model topologically organized as a graph is easier to interpret. Second, the machinery of graph theory can be applied to effectively perform learning and inference tasks for a graphical model. Third, such representation can provide valuable insights into the properties of a probabilistic model, e.g. incorporated conditional independencies.

This method allows to effectively leverage prior knowledge regarding dependencies of variables, leading to an elegant yet powerful description of the system of interest. For example, consider a simple model of diagnosis of Parkinson's disease that is assumed to be dependent on multiple variables: (see Fig. 2.1). This is a directed acyclic graph with vertices denoting random variables and edges depicting dependency relations. For instance, the probability of a person to have Parkinson's disease depends on age which corresponds to conditional probability $p(\ \textit{Parkinson's disease} \mid \textit{Age}\ )$.

2.3. Bayesian networks

If a probabilistic graphical model has a structure of a directed acyclic graph, it falls into the family of Bayesian networks (or directed graphical models). In the case of a Bayesian network, nodes represent variables of a model and edges denote conditional dependencies. As the result, a modelled joint distribution is factorized as a product of conditional distributions:

$$p_\theta(\mathbf{x}_1, x_2, ..., \mathbf{x}_n) = \prod_{i=1}^{N} p_\theta(\mathbf{x}_i|\pi(\mathbf{x}_i)) \tag{2.1}$$

where $\pi(\mathbf{x}_i)$ is a set of ancestor variables of $\mathbf{x}_i$ in the graphical model. For example, in the model depicted at Fig. 2.1 variables *Sex* and *Income* are ancestors of variable *Socioeconomic status*. When a node is a root node, the set of parents is an empty set. In this case, the respective probability is unconditional and called a prior distribution.

Keeping a set of parent variables small for each non-root variable leads to a compact graphical representation of a joint distribution. Let us look at the graphical model shown in Fig. 2.1 which is a Bayesian network. It corresponds to a joint distribution which models a chance of a person to have Parkinson's disease d. This depends on the person's age a, socioeconomic status se and environmental factors e. The last two factors are, in turn, affected by sex s and income i. As the result, the joint distribution that the graph models factorizes as follows:

$$p(d, a, se, e, s, i) = p(d|a, se, e) \cdot p(se|s, i) \cdot p(e|i) \cdot p(a) \cdot p(s) \cdot p(i)$$

Generally, each conditional distribution $p(\mathbf{x}_i|\pi(\mathbf{x}_i))$ in a Bayesian network is a probability function that returns a probability of a variable given values of its ancestor variables. To parameterize such functions, an arbitrary approximation method can be chosen, e.g. a lookup table or a neural network.

In the case of neural networks, one parameterizes parameters of a distribution as deterministic functions with arbitrary complex neural networks [9]. For example, if $p(\mathbf{x}_i|\pi(\mathbf{x}_i))$ is assumed to be Gaussian, the parameterization is defined as follows:

$$\begin{cases} \mu(\pi(\mathbf{x}_i)) = NeuralNet_{\theta_1}(\pi(\mathbf{x}_i)) \\ \sigma(\pi(\mathbf{x}_i)) = NeuralNet_{\theta_2}(\pi(\mathbf{x}_i)) \\ p_{\theta_1,\theta_2}(\mathbf{x}_i|\pi(\mathbf{x}_i)) = \mathcal{N}(\mathbf{x}_i; \mu(\pi(\mathbf{x}_i)), \sigma(\pi(\mathbf{x}_i))) \end{cases} \tag{2.2}$$

where θ_1, θ_2 are parameters of neural networks optimized during learning.

2.3.1. Learning in Bayesian networks

Let us have a distribution p from which we sample to form a dataset D and which we want to model by p_θ. We assume that the data which we collected consists of observations that are independent and identically distributed (i.i.d.). Our goal is to find parameters θ of the model such that the true distribution is approximated as well as possible.

In other words, we want the model to assign high probabilities to the actual data points. It is equivalent to maximizing the following objective called log-likelihood:

$$log\, p_\theta(D) = \sum_{\mathbf{x} \in D} log\, p_\theta(\mathbf{x}) \tag{2.3}$$

The right-hand side is the sum over each data point which is the consequence of using i.i.d. samples in the dataset D. Maximizing log-likelihood can be also interpreted as minimizing the Kullback-Leibler (KL) divergence [16] between the data distribution and the model:

$$KL(p||p_\theta) = \int_{-\infty}^{+\infty} p(\mathbf{x})\, log \frac{p(\mathbf{x})}{p_\theta(\mathbf{x})}\, d\mathbf{x} \tag{2.4}$$

The right-hand side can be further rewritten as follows:

$$\int_{-\infty}^{+\infty} p(\mathbf{x})\, log \frac{p(\mathbf{x})}{p_\theta(\mathbf{x})}\, d\mathbf{x} = -H(p) - \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})}[log\, p_\theta(\mathbf{x})] \tag{2.5}$$

where $H(p)$ is the entropy of the true distribution p and the second term is the expected value of log-likelihood. Since the data distribution is unknown, the expected value can be estimated by the Monte-Carlo method:

$$\mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})}[log\, p_\theta(\mathbf{x})] \approx \frac{1}{|D|} \sum_{\mathbf{x} \in D} log\, p_\theta(\mathbf{x}) \tag{2.6}$$

Hence, in order to minimize the KL divergence between the true distribution and the model, one is aimed at maximization of the empirical log-likelihood.

2.4. Latent variable models

Bayesian networks that were introduced in the previous section can be extended to directed latent variable models by introducing latent variables. Opposed to observed variables that are part of a dataset, latent variables are never observed and hence are not included in the dataset but can be inferred. Incorporating a latent variable into a model can be seen as adding a new node to a directed graphical model along with the corresponding conditional dependency.

As a result, the probabilistic model with latent variables or a latent variable model approximates a joint probability distribution as follows:

$$p_\theta(\mathbf{x}, \mathbf{z}) \tag{2.7}$$

where $\mathbf{x}$ is a set of observed variables and $\mathbf{z}$ is a set of latent variables. It can also be seen that the probability over observed variables that we considered before is actually a marginal distribution now:

$$p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{x}, \mathbf{z})\, d\mathbf{z} \tag{2.8}$$

Latent variable models are more expressive compared to fully-observed models. They also allow to incorporate prior knowledge about the system without additional measurements. For example, it can be the case that data is clustered yet we do not know what categorical variable accounts for the clustering. Including discrete latent variable allows us to create a model capturing this phenomenon.

2.4.1. Learning in latent variable models

As was discussed in 2.3.1, to learn a model of a true probability distribution, one maximizes log-likelihood of the data (equation 2.3). The difference in the case of latent variable models is that the objective becomes the marginal log-likelihood. Indeed, by substituting $p_\theta(\mathbf{x})$ in equation 2.3 as in equation 2.8 we get the following expression for log-likelihood:

$$log\, p_\theta(D) = \sum_{\mathbf{x} \in D} log \int p_\theta(\mathbf{x}, \mathbf{z})\, d\mathbf{z} \tag{2.9}$$

The integral does not allow one to write the right-hand side as a sum of independent log probabilities nor does it have an analytic solution. This makes the computation of log-likelihood intractable and does not allow us to optimize the parameters of the model as efficiently as in the fully-observed case.

2.5. Variational Autoencoder

An efficient algorithm to perform learning of directed latent variable models is called Auto-encoding variational Bayes (AEVB) and was provided by Kingma and Welling in [9]. It is a stochastic variational inference and learning algorithm that gives rise to the variational autoencoders (VAE) framework. In this section, we will consider in detail the main ideas of AEVB and how a probabilistic model with latent variables can be learned with VAE.

2.5.1. Variational inference

Recall that the main obstacle for learning a Bayesian network with latent variables was the intractability of $p_\theta(\mathbf{x})$ (see 2.4.1). In turn, $p_\theta(\mathbf{x}, \mathbf{z})$ is easy to calculate when the model is defined. Hence, the intractability of marginal log-likelihood is equivalent to the intractability of posterior distribution $p_\theta(\mathbf{z}|\mathbf{x})$, which is the consequence of the following identity:

$$p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{z}|\mathbf{x})\ p_\theta(\mathbf{x}) \tag{2.10}$$

If one of them is known, the other can easily be computed. Therefore, an efficient algorithm for approximating either of them is required to learn a latent variable model by maximizing the marginal likelihood.

Variational inference [17] is a family of techniques that approximate intractable distributions in probabilistic modeling. The general idea of such algorithms is to treat the problem as an optimization problem. More precisely, variational inference tries to find a member q of a pre-defined family of tractable distributions Q such that KL divergence (equation 2.4) between q and the posterior distribution of interest is minimized:

$$\arg\min_{q \in Q} KL(q(\mathbf{z})||p_\theta(\mathbf{z}|\mathbf{x})) \tag{2.11}$$

As a result, so-called variational parameters of an approximate posterior q are learned at the optimization step. The family of tractable distributions Q is taken such that it is easy to optimize yet flexible enough to approximate the true posterior.

2.5.2. The evidence lower bound

What is exactly an optimization objective in variational inference? Note that by definition of KL divergence, optimizing it directly is not tractable since the computation of logarithm of posterior is required:

$$KL(q(\mathbf{z})||p_\theta(\mathbf{z}|\mathbf{x})) = \int_{-\infty}^{+\infty} q(\mathbf{z})\ log \frac{q(\mathbf{z})}{p_\theta(\mathbf{z}|\mathbf{x})} d\mathbf{z} = \mathbb{E}_{\mathbf{z} \sim q(\mathbf{z})}[log\ q(\mathbf{z}) - log\ p_\theta(\mathbf{z}|\mathbf{x})] \tag{2.12}$$

The right-hand side can be further rewritten using the identity 2.10:

$$KL(q(\mathbf{z})||p_\theta(\mathbf{z}|x)) = \mathbb{E}_{\mathbf{z} \sim q(\mathbf{z})}[log\ q(\mathbf{z}) - log\ p_\theta(x, \mathbf{z})] + log\ p_\theta(\mathbf{x}) \tag{2.13}$$

This yields an alternative objective for optimization that can be computed, which is called the evidence lower bound (ELBO) or variational lower bound:

$$ELBO(q(\mathbf{z})) = \mathbb{E}_{\mathbf{z} \sim q(\mathbf{z})}[log\ p_\theta(\mathbf{x}, \mathbf{z}) - log\ q(\mathbf{z})] \tag{2.14}$$

Note that KL divergence is equivalent to ELBO up to constant with respect to $q(\mathbf{z})$:

$$ELBO(q(\mathbf{z})) = log\ p_\theta(\mathbf{x}) - KL(q(\mathbf{z})||p_\theta(\mathbf{z}|\mathbf{x})) \tag{2.15}$$

Hence, maximizing ELBO one also minimizes KL divergence between approximate and true posteriors. Since $log\ p(\mathbf{x})$ is a constant and KL divergence is non-negative by definition, maximizing ELBO can be seen as decreasing the gap between marginal log-likelihood and ELBO itself thus "squeezing" KL divergence between them.

2.5.3. Auto-encoding Variational Bayes

AEVB employs ideas of variational inference to approximate the intractable posterior $p_\theta(\mathbf{z}|\mathbf{x})$. To do so, a parametric inference model is $q_\phi(\mathbf{z}|\mathbf{x})$ is introduced, where ϕ denotes variational parameters of the model. As was demonstrated in the previous section, the inference model can be learned by maximizing the evidence lower bound:

$$\mathcal{L}_{\theta,\phi}(\mathbf{x}) = \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[log\, p_\theta(\mathbf{x},\mathbf{z}) - log\, q_\phi(\mathbf{z}|\mathbf{x})] \tag{2.16}$$

where parameters of both the probability model and inference model are optimized simultaneously. It leads to concurrent maximization of the marginal log-likelihood $p_\theta(\mathbf{x})$ and minimization of KL divergence between approximate and true posteriors.

Given the objective, one can use stochastic gradient descent to jointly optimize parameters θ, ϕ of the models. However, the gradient of ELBO with respect to model parameters $\nabla_{\theta,\phi}\mathcal{L}_{\theta,\phi}(\mathbf{x})$ is often intractable due to the expected value operator. Note that the gradient is easy to compute for parameters θ:

$$\nabla_\theta \mathcal{L}_{\theta,\phi}(\mathbf{x}) = \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[\nabla_\theta\, log\, p_\theta(\mathbf{x},\mathbf{z})] \tag{2.17}$$

but a good unbiased estimator should be obtained for the gradient of ELBO with respect to inference parameters ϕ since gradient and expectation operators cannot be swapped anymore. There exist several Monte Carlo based approaches for estimating the gradient but they demonstrated high variance and were found to be impractical [9].

AEVB uses the reparameterization trick to obtain an unbiased estimator of the variational bound and its derivatives with respect to both parameters thus allowing one to optimize the objective 2.16 using gradient descent.

2.5.4. Reparameterization trick

Let $\mathbf{z}$ be a continuous random variable sample from a conditional distribution $q_\phi(z|\mathbf{x})$. Let ϵ be a random variable sampled from a distribution $p(\epsilon)$ that is independent of $\mathbf{x}$ and ϕ. One can often find a deterministic transformation g such that:

$$z = g(\epsilon, \mathbf{x}, \phi) \tag{2.18}$$

where g is a differentiable invertible function. Having such a function, one can express the inference model $q_\phi(\mathbf{z}|\mathbf{x})$ as a deterministic transformation of random noise generated by a simple distribution $\epsilon \sim p(\epsilon)$.

2.5.5. Gradient of ELBO

The reparameterization trick (equation 2.18) allows one to calculate the gradient of expectation with respect to variational parameters which was previously intractable:

$$\begin{aligned}\nabla_\phi \mathcal{L}_{\theta,\phi}(\mathbf{x}) &= \nabla_\phi \, \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[log\, q_\phi(\mathbf{z}|\mathbf{x})] \\ &= \nabla_\phi \, \mathbb{E}_{\epsilon\sim p(\epsilon)}[log\, q_\phi(\mathbf{z}|\mathbf{x})] \\ &= \mathbb{E}_{\epsilon\sim p(\epsilon)}[\nabla_\phi \, log\, q_\phi(\mathbf{z}|\mathbf{x})]\end{aligned} \tag{2.19}$$

Thus, combined with the gradient with respect to θ (see equation 2.17), the resulting Monte Carlo estimator is now differentiable and unbiased (see [9] for proof). It can be used for learning a Bayesian network with latent variables by maximizing the evidence lower bound.

2.5.6. Variational autoencoder

Variational autoencoders (VAE) is a framework for learning a directed latent variable model with latent variables $p_\theta(\mathbf{x}, \mathbf{z})$ also called generative model along with a corresponding inference model $q_\phi(\mathbf{z}|\mathbf{x})$, where parameters θ and ϕ of models are optimized with the AEVB algorithm. In the section we consider generative models that factorize as in equation 2.10 leading to the connection with auto-encoders.

Within the framework, one optimizes the evidence lower bound albeit reformulated to reflect the dual nature of auto-encoders. The reformulation of equation 2.16 is following:

$$\begin{aligned}\mathcal{L}_{\theta,\phi}(\mathbf{x}) &= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[log\, (p_\theta(\mathbf{x}|\mathbf{z})\, p_\theta(\mathbf{z})) - log\, q_\phi(\mathbf{z}|\mathbf{x})] \\ &= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[log\, p_\theta(\mathbf{x}|\mathbf{z}) - (log\, q_\phi(\mathbf{z}|\mathbf{x}) - \ p_\theta(\mathbf{z}))] \\ &= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[log\, p_\theta(\mathbf{x}|\mathbf{z})] - KL(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}))\end{aligned} \tag{2.20}$$

This objective is especially efficient to optimize when KL divergence can be integrated analytically.

Let us deconstruct the resulted formulation step by step. Here, a generative model $p_\theta(\mathbf{x}, \mathbf{z})$ is factorized as the product of a log-likelihood $p_\theta(\mathbf{x}|\mathbf{z})$ of observed variables $\mathbf{x}$ given latent variables $\mathbf{z}$ and a prior distribution $p_\theta(\mathbf{z})$ over a latent space. It leads to the objective having two terms: the log-likelihood and KL divergence between the inference model and the prior over a latent space.

An inference model $q_\phi(\mathbf{z}|\mathbf{x})$ plays role of an encoder that maps an observed $\mathbf{x}$-space to a latent $\mathbf{z}$-space. The second term of the objective puts pressure on the conditional distribution to approximate pre-defined prior $p(\mathbf{z})$. It leads to the latent representation resembling the distribution of interest. The most common choice is to set the prior to be a Normal distribution. In this case, the second term influences the latent code to resemble Normal distribution as well.

A log-likelihood $p_\theta(\mathbf{x}|\mathbf{z})$, in turn, is a mapping in the opposite direction which corresponds to a decoder reconstructing an observed data point from its latent representation. The term reaches the maximum value when the reconstruction is exact. Hence, the term is called the negative reconstruction error as it penalizes low probability assigned to a data point $\mathbf{x}$ under the generative model.

2.5.7. Parameterizing distributions in VAE

As we have already discussed in the Bayesian network section (see section 2.3), a convenient and expressive choice for parameterization of distributions in the probabilistic models is neural networks. They are traditionally trained via stochastic gradient descent which is beneficial since the AEVB algorithm relies on stochastic optimization and no additional optimization techniques are required.

To parameterize a conditional distribution, one should assume an approximate distribution and define its parameters as real-valued deterministic functions of a given variable. Each function is further approximated with a deep neural network of arbitrary complexity. In this case, weights and biases of the neural network constitute parameters of the corresponding model.

For example, let a log-likelihood $p_\theta(\mathbf{x}|\mathbf{z})$ be a multivariate Normal distribution $\mathcal{N}(x;\mu,\sigma)$. We define μ, σ as functions of latent variables $\mathbf{z}$ which leads to the following parameterization:

$$\begin{cases} \mu(\mathbf{z}) = NeuralNet_{\theta_1}(\mathbf{z}) \\ \sigma(\mathbf{z}) = NeuralNet_{\theta_2}(\mathbf{z}) \\ p_{\theta_1,\theta_2}(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x};\mu(\mathbf{z}),\sigma(\mathbf{z})) \end{cases} \tag{2.21}$$

where θ_1, θ_2 are parameters of neural networks optimized during learning. For brevity, we further refer to separate parameters of a conditional distribution as joint parameters denoted with one symbol (e.g. θ_1, θ_2 are denoted as θ).

In the case of prior distributions over latent variables, only an approximating distribution has to be specified since the distribution is unconditional. In this case, prior $p_\theta(\mathbf{z})$ does not have parameters to optimize so the subscript θ can be further omitted.

2.5.8. Learning faces with VAE

One possible application of VAE is the generative modeling of images. Kingma and Welling [9] in the original VAE paper trained a VAE model on Frey Face dataset which contains multiple photos of a person's face demonstrating different emotions such as anger or happiness. In the case, the observed $\mathbf{x}$-space is continuous pixel space. Thus, a decoder $p_\theta(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}; \mu(\mathbf{z}), \sigma(\mathbf{z}))$ is designed as a multivariate Gaussian as the system 2.21 describes. Encoder $q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \mu(\mathbf{x}), \sigma(\mathbf{x}))$ has the identical parameterization and also approximates conditional Normal distribution. Prior $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; 0, \mathbf{I})$ is assumed to be a multivariate standard normal distribution. The resulting VAE model is trained by optimizing the objective 2.20 with stochastic gradient ascent.

One can use a learned model to visualize a low-dimensional data manifold to which high-dimensional data is projected by the encoder. For convenient visualization, we will consider a two-dimensional latent space. However, conclusions can be made to an arbitrary dimensional latent space without loss of generality.

A learned latent space is shown in Fig.2.2. To produce the image, coordinates in the latent space are sampled via inverse transform sampling with an evenly spaced unit square as input. A decoder is further applied to those coordinates to obtain a reconstruction in the pixel space.

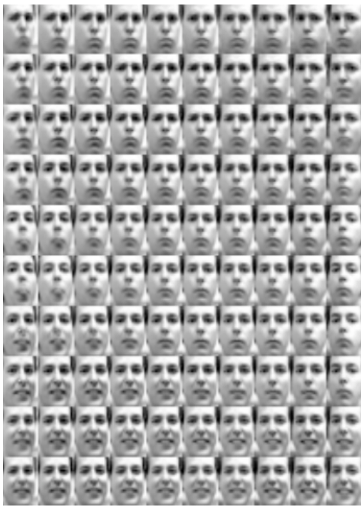


Figure 2.2.: Learned two-dimension latent space of Frey Face from [9].

It is also possible to interpolate between different categories of input data by interpolating in the latent space. For example, one can define sub-spaces in $\mathbf{z}$-space that corresponds to "smiling" and "serious" and move along the line in the data manifold that connects them. Projecting a point on the line from the latent space to the pixel space will yield in generating a face that the model considers to be an intermediate between "smiling" and "serious".

Another application of VAE is representation learning. It can be seen from Fig.2.2 that a generative model with only 2 latent dimensions is able to reconstruct a 20×28 pixels image with high accuracy. Therefore, VAE can be used to learn robust generative factors of the observational data which serve as an input for a classification model. This strategy was proved to be especially useful in semi-supervised learning where VAE-based models achieve state-of-the-art results [18].

2.6. Representation learning with VAE

As we figure out in the previous section, VAE is able to learn a low-dimensional representation of observed data when a generative model is deployed as follows:

$$p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{x}|\mathbf{z})\, p(\mathbf{z}) \tag{2.22}$$

where latent variables $\mathbf{z}$ are essentially features forming the latent representation.

This makes the framework especially suitable for the task of representation learning that is aimed at extracting statistically independent explanatory factors of variation from input data. In other words, given high-dimensional data, one seeks for a compact, faithful, explicit and interpretable [19] set of features that can serve as a bias for consecutive learning scenarios, e.g. classification. Having such representation is beneficial when performing tasks that demand generalization to unseen data, dealing with missing data or small dataset size (e.g. semi-supervised learning). In multiple approaches [20, 21, 22], generative modeling is incorporated as an auxiliary task providing a supervised predictor with a low-dimensional input. This strategy leads to more robust performance and better generalization to unseen data compared to pure discriminative models [23].

Learning human-interpretable explanatory factors of variation is particularly appealing and has gained a lot of interest in recent years. As we will show, if latent variables in variational auto-encoders are jointly independent, they are equivalent to causal variables allowing for learning independent causal mechanisms from data. We first define the causal mechanism and then provide an algorithm for learning them from data with variational auto-encoders which was described in [6].

2.6.1. Independent causal mechanisms

Mathematically, causality is described by a structural causal model or causal graph which entails a joint distribution over all the variables of interest $\mathbf{x}_1, ..., \mathbf{x}_n$. This structure is essentially a probabilistic graphical model combined with a set of equations describing a data-generating process. Each equation is technically an assignment of the following form:

$$\mathbf{x}_i := f_i(\pi(\mathbf{x}_i), \mathbf{u}_i) \quad \forall i : 1 \leq i \leq n \tag{2.23}$$

where $\pi(\mathbf{x}_i)$ denotes parents of $\mathbf{x}_i$ in the causal graph and $\mathbf{u}$ is a set of stochastic unexplained variables assumed to be jointly independent. From a representation learning point of view, it is more important that a causal graph implies factorization of the joint distribution into causal conditions (also called causal mechanisms):

$$P(\mathbf{x}_1, ..., \mathbf{x}_n) = \prod_{i=1}^{n} P(\mathbf{x}_i|\pi(\mathbf{x}_i)) \tag{2.24}$$

where conditionals correspond to the structural assignments. Such factorization is also called a disentangled factorization as it separates a generating process into governing causal mechanisms that are assumed to be independent [24]. It is summarized in the following principle [6]:

Independent Causal Mechanisms (ICM) Principle: *The causal generative process of a system's variables is composed of autonomous modules that do not inform or influence each other. In the probabilistic case, this means that the conditional distribution of each variable given its causes (i.e., its mechanism) does not inform or influence the other mechanisms.*

Hence, given a joint probability model in a disentangled factorization form, we can perform an intervention upon a mechanism of interest. According to the ICM principle, it will not affect other causal mechanisms in a system, i.e. they remain invariant. Hence, we can observe the effect of the intervention by sampling from the distribution and infer the nature of the causal mechanism that we manipulate.

If we assume that a real-world process is governed by multiple causal mechanisms, we expect that any change in distribution (e.g. observing healthy people instead of infected ones) is caused by a change in causal mechanisms (e.g. change of a health condition). Schölkopf et al. in [6] stated the following hypothesis:

Sparse Mechanism Shift (SMS) Hypothesis: *Small distribution changes tend to manifest themselves in a sparse or local way in the causal/disentangled factorization.*

If the hypothesis holds, changing one causal factor will lead to a sparse change in observed data which makes it much easier to make conclusions about the nature of the factor. It makes a disentangled representation much more preferable than an entangled one that allows for arbitrary many terms to be affected by a sparse intervention upon mechanisms.

2.6.2. Learning a disentangled representation

Let us now see how one can use variational autoencoders to learn a disentangled (causal) representation that corresponds to rendering a joint probability distribution of interest as a product of jointly independent causal mechanisms (see equation 2.24).

Assume that we are given high-dimensional observed data $\mathbf{x}$ and a pre-defined causal graph from which we want to reconstruct a set of independent causal variables $\mathbf{s}$. Each causal variable is defined through a corresponding causal mechanism describing the data-generating process (see equation 2.23):

$$\mathbf{s}_i := f_i(\pi(\mathbf{s}_i), \mathbf{u}_i) \quad \forall i : 1 \leq i \leq n \tag{2.25}$$

where $\pi(\mathbf{s}_i)$ denotes parents of $\mathbf{s}_i$ in the causal graph and $\mathbf{u}$ is a set of stochastic unexplained variables assumed to be jointly independent.

It can be done by learning a deep generative model of the form 2.22 within the framework of VAE. In this case, the causal graph is trivial, i.e. a set of parental nodes $\pi(\mathbf{s}_i) = \emptyset$ for every causal variable $\mathbf{s}_i$. Latent variables $\mathbf{z}$ correspond to unexplained noise variables $\mathbf{u}$ in equation 2.25. If they are jointly independent, then causal mechanisms learned by the deep probabilistic model are also independent by definition 2.25. This condition can be fulfilled by choosing an appropriate prior distribution, e.g. a multivariate Gaussian with diagonal covariance matrix.

Hence, the learned latent representation is decomposed such that factors of the representation are interpreted as conditional corresponding to causal mechanisms. Additionally, intuitive explanations for each dimension can be inferred by a sequence of interventions and reconstructions.

2.6.3. Supervision bias

In the section 2.1 we motivated studying probabilistic modeling by the need to account for inherent uncertainty appearing in many systems of human interest. Besides, we gave an example of such a system, namely a supervised learning problem with 2 sets of observed variables whose stochastic relationship is aimed to be modeled. More general task would be learning of a joint probability distribution $p(\mathbf{x}, \mathbf{y})$ over both a set of observation variables $\mathbf{x}$ and a set of label variables $\mathbf{y}$ from which conditional distribution $p(\mathbf{y}|\mathbf{x})$ can be computed.

Overall, if data appears as pairs $(\mathbf{x}, \mathbf{y})$ where $\mathbf{y}$ is a set of labels and $\mathbf{x}$ is observation data, it can be used as an inductive bias for learning meaningful and human-interpretable representations. Including label information in a generative model generally improves the quality of the model by providing insights about the structure of true data distribution. Indeed, a label variable can be interpreted as the context that partially governs the generation of an observed variable $\mathbf{x} \sim p(\mathbf{x}|\mathbf{y})$ which should be captured by a probabilistic model. Hence, generative factors that the model learns must encapsulate labels and characteristic information associated with them. Importantly, label-aware latent space allows one to perform an intervention, i.e. manipulating a latent representation before reconstructing it, to generate a sample with desired characteristics. It is of immense interest in the area of inverse design, e.g. generating molecules with desired properties.

There are multiple ways of incorporating labels into a deep generative model depending on the task and strength of constraints that labels are meant to put on a latent representation. In every case, however, it influences a latent space of a latent variable model. The most common approach to include labels in the latent space is by explicitly assigning a latent variable to each label [25, 18]. Such models have the advantage of learning a disentangled representation of data since each factor of variation is isolated by design. They were originally developed for the task of semi-supervised classification where the number of labeled data is low. If the label information is given, a model fixes values of latent variables to corresponding labels such that $\mathbf{z}_{\mathbf{y}_i} = \mathbf{y}_i$. Otherwise, latent representation is computed by an encoder. Even though these approaches proved to be effective and achieved state-of-the-art results in semi-supervised learning [18, 26], they are not suitable for representation learning. The main obstacle is that treating labels as latent variables allows the latter to capture only the label itself but not the information associated with the label. It can be harmful in the case when labels are abstract, e.g. "young" or "happy" for people's faces. Besides, the approach only allows interventions as changing values of $\mathbf{z}_{\mathbf{y}_i}$ which is too restrictive as it does not allow to manipulate latent variables continuously.

Another way is to use an auxiliary discriminative model that takes a latent representation as an input for a classification task [20]. The classifier imposes regularization of a latent space such that the latter is forced to capture the characteristics reflected in a label. However, it leads to label information being completely entangled within a latent space. In other words, the approach fails to isolate characteristics of different labels from each other in latent representation which makes it difficult to perform interventions, i.e. manipulating desired characteristics of data, or generating data with desired properties.

2.6.4. Capturing label characteristics in VAEs

An alternative approach called characteristic capturing VAE (CCVAE) has been introduced recently [10] that aims at capturing characteristic of labels from data by conditional latent variables $\mathbf{z} \sim p_\theta(\mathbf{z}|\mathbf{y})$. It effectively combines the advantages of both aforementioned approaches, namely disentanglement of characteristics of different labels and the ability to include rich information contained in them into a latent space.

The idea is to separate a latent space $\mathbf{z}$ into two sub-spaces: the characteristic sub-space $\mathbf{z}_c$ and the general sub-space $\mathbf{z}_{\setminus c}$. The first one is designed to encode characteristics associated with labels while the second one accounts for more general features of observed data. Furthermore, the characteristic sub-space is partitioned such that each label $\mathbf{y}^i$ has access only to its own part of the sub-space $\mathbf{z}_c^i$. This allows to prevent entanglement of label characteristics within the characteristic latent space $\mathbf{z}_c$ and leads to corresponding

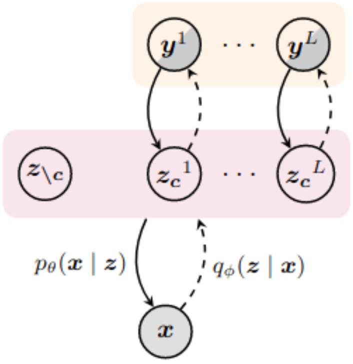


Figure 2.3.: Graphical model of CCVAE [10].

generative factors being interpretable.

The graphical model of CCVAE is shown in Fig. 2.3. It corresponds to the generative model factorized as a product of two separate generative models:

$$p_\theta(\mathbf{x}, \mathbf{y}, \mathbf{z}) = p_\theta(\mathbf{x}|\mathbf{z})\, p_\theta(\mathbf{z}|\mathbf{y})\, p(\mathbf{y}) \tag{2.26}$$

where the model $p_\theta(\mathbf{z}|\mathbf{y})$ forces the latent space to encapsulate label associated characteristics. Such formulation is particularly suitable for a conditional generation where one is aimed at generating a sample in $\mathbf{x}$-space with desired properties defined by $\mathbf{y}$.

In the supervised case, posterior distribution $p_\theta(\mathbf{z}|\mathbf{x}, \mathbf{y})$ is conditioned to both observation and label variables and approximated with the following inference model:

$$q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}) = \frac{q_\phi(\mathbf{y}|\mathbf{z}_c)\, q_\phi(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{y}|\mathbf{x})} \tag{2.27}$$

where conditional distribution $q_\phi(\mathbf{y}|\mathbf{x}) = \int q_\phi(\mathbf{y}|\mathbf{z}_c)\, q_\phi(\mathbf{z}|\mathbf{x})\, d\mathbf{z}$ reflects that observation variables $\mathbf{x}$ and label variables $\mathbf{y}$ are connected via latent variables $\mathbf{z}_c$.

As in the classical VAE, the model is optimized by maximizing the evidence lower bound. In the fully supervised case it is done by maximizing the following objective:

$$\mathcal{L}(\mathbf{x}, \mathbf{y}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\frac{q_\phi(\mathbf{y}|\mathbf{z}_c)}{q_\phi(\mathbf{y}|\mathbf{x})}\, log \frac{p_\theta(\mathbf{x}|\mathbf{z})\, p_\theta(\mathbf{z}|\mathbf{y})}{q_\phi(\mathbf{y}|\mathbf{z}_c)\, q_\phi(\mathbf{z}|\mathbf{x})}\right] + log\, q_\phi(\mathbf{y}|\mathbf{x}) + log\, p(\mathbf{y}) \tag{2.28}$$

The objective is similar to the one of vanilla VAE (see eq. 2.20). It also has the log-likelihood term $log\, p(\mathbf{x}|\mathbf{z})$ and a term $log\, q_\phi(\mathbf{z}|\mathbf{x}) - log\, p_\theta(\mathbf{z}|\mathbf{y})$ which resembles KL divergence between the inference model and the generative model. The latter term can intuitively be interpreted as the one that pressures the inference model $q_\phi(\mathbf{z}|\mathbf{x})$ to approximate the conditional distribution $p_\theta(\mathbf{z}|\mathbf{y})$ and thus enforcing capturing label characteristics from observed data. The key difference is the classification term $log\, q_\phi(\mathbf{y}|\mathbf{x})$ which is the consequence of the definition of the inference model (see eq. 2.27).

The approach allows one to analyze generative factors of data related to labels by forming the bridge between observations and its labels via latent variables. One can explore the relation by varying along the corresponding axis of a latent space while holding the remaining ones fixed, reconstructing the varied latent representation and observing resulting changes. It can be especially helpful when the relation is unknown thus giving insights about causal mechanisms in the system of interest. For example, in the case of a people's faces dataset, we perfectly know what label "hair color" is responsible for. Therefore, we would expect the model to vary only the hair color when the value of the label is changed. However, it might be more difficult when we consider, for example, labeled medical images where the effect of pathology is unknown. CCVAE makes it possible to investigate it via the analysis of corresponding generative factors. As a result, the approach is essentially a combination of a generative model and a classifier with inherent explainability which might be helpful when exploring the relationship between images and their classes.

3. Conditional Graph Variational Autoencoder

In this chapter, we generalize the framework of CCVAE to graph-structured data. The generalization is motivated by the problems of neuroimaging where data is traditionally represented with graphs. Precisely, we want to analyze electrical activity data but the approach is easily generalizable to other techniques. We define the generative and inference models with latent variables that are trained by maximizing the evidence lower bound with the AEVB algorithm. The models are factorized such that a part of latent variables accounts for encoding label characteristics in observed data resulting in learning label-related causal mechanisms. We describe an algorithm to investigate the meaning of those mechanisms in the chapter as well.

3.1. Introduction

Nowadays, multiple medical imaging techniques are used in clinical applications for diagnosing brain pathologies. One of the most widely utilized approaches is electroencephalography (EEG) which captures temporal behaviors of neural electrical activity. The general experimental setup for EEG consists of multiple interconnected electrodes located around a skull recording brain electrical activity. It has multiple advantages compared to other methods such as functional magnetic resonance imaging (fMRI) or positron emission tomography (PET), namely non-invasiveness, sufficient spatial resolution and low cost [11]. EEG also features high temporal resolution which makes the approach particularly suitable for studying temporal patterns in neural activity data. Specifically, it is used to investigate task-evoked neural synchronisation which manifests itself in the synchronised electrical activity of distal brain regions. The co-activation of spatially distributed brain regions (or functional connectivity) is tightly connected to multiple brain malfunctions as the latter often alter neural connections in the brain. For example, Alzheimer’s disease leads to progressive structural and functional cortical disconnection [27] which can be detected by analyzing functional connectivity alterations. Overall, the availability and the ability to reflect abnormalities in neural synchronization allows EEG to be an ideal method for early diagnosis testing.

However, the analysis of functional connectivity is difficult to perform for a human as the dimensionality of electrophysiological data is very high. This leads to the urge of developing automatic classification algorithms that are able to identify pathology-related characteristics in data and give a decision regarding a patient's status. Many machine learning classifying approaches were developed to tackle the problem by either using a set of hand-crafted pre-calculated features or explicitly deriving them from EEG data. A common approach to automatic feature extraction encompasses calculating temporal or frequency features from each electrode's recording separately thus neglecting the spatial information in the brain, e.g. via wavelet transform [28] or power spectral density [29]. Recently, several algorithms were developed that analyze functional connectivity matrices by examining corresponding network topologies and computing metrics from graph analysis [30, 31] or information theory [32]. These methods are more advantageous as they consider spatial information and account for interactions of distal brain regions. However, graph metrics provide a sub-optimal solution for feature extraction as they do not preserve the whole information hidden in neural activity data. Besides, they often require a substantial computation time that scale exponentially with the number of electrodes in the experimental setup. The overall advantage of handcrafted features is their interpretability which is often guaranteed by the fundamental theory behind them. For example, coefficients of wavelets indicate similarity between data signal, i.e. EEG recording, and a wavelet base function. It can be further used to, for example, gather insights about a pathology from salient features that a classification algorithm relies on.

Alternatively, one can deploy neural networks to automatically extract features from EEG data. Given a sufficient amount of data, they are able to recognise complex patterns in neural activity recordings that are related to a certain pathology, e.g. [33]. The most popular choice for a neural network-based classifier is convolutional neural networks (CNN) followed by deep belief networks and recurrent neural networks [34]. Convolution-based networks usually demonstrate higher classification accuracy as they can extract rich temporal information from EEG signals. However, they cannot handle topological information of brain networks as they are assumed to work with data defined on a regular grid. Recently, a new class of neural networks was developed - graph convolutional neural networks [35] - which is more suitable for the neuroimaging tasks as it learns spatio-temporal features from data. It led to a substantial interest in using those methods in computational neuroscience with respect to brain functional networks [36, 37]. Even though neural network-based classifiers are highly accurate when given enough data points, they are essentially black boxes as it is often impossible to recover the meaning of features that they use for classification. Besides, they rely on statistical associations in data which can be misleading, especially with the presence of spurious correlations.

One common drawback of classification models described above is their narrow specialization on the classification task. They often boil down to approximating conditional probability distribution $p(\mathbf{y}|\mathbf{x})$ where $\mathbf{x}$ is a set of predictors and $\mathbf{y}$ is a set of labels. Furthermore, they do not allow to perform inference tasks, for example, asking questions such as "what is the most likely $\mathbf{x}$ for a given $\mathbf{y}$". This limits the amount of knowledge one can extract from data and often does not allow gathering new insights about the system. Combined with little interpretability and vulnerability to statistical pitfalls (see 1.1), such approaches are barely applicable to real-world applications and fail to provide a reliable decision-making algorithm. In this chapter, we address the aforementioned problems proposing a novel approach for learning a generative model that takes both EEG data and functional connectivity matrices as input and learns independent causal mechanisms that govern the process of data generation. The model is based on the framework of CCVAE [10] which allows one to learn a generative model and a classification model simultaneously. It is able to successfully indicate an underlying malfunction of the brain as well as learn characteristics of data related to each malfunction. The robustness of learned features is enforced by the underlying causal model and the interpretability is achieved by manual exploring the meaning of each learned generative factor. Though we demonstrate the applicability of the proposed method to EEG data, one can essentially apply it to any brain imaging technique used to reconstruct functional connectivity, e.g. fMRI or PET.

3.2. Motivation

Performing electrophysiological recordings (EEG), we measure voltage fluctuations generated by neurons of the brain [38]. This is a stochastic process that can be seen as sampling from an unknown underlying process governing the generation of electrical activity. Thus, for a single region of the scalp on which we perform an EEG recording:

$$\mathbf{x} \sim p(\mathbf{x}) \tag{3.1}$$

where $\mathbf{x}$ is an observed random variable corresponding to a recorded EEG signal, $p(\mathbf{x})$ is a distribution that governs the generation of the signal. As a rule, measuring electrical activity is performed with multiple electrodes where each electrode records an EEG signal on an underlying scalp region. Hence, we will further consider a matrix $\mathbf{X} = \{\mathbf{X}_1, ..., \mathbf{X}_N\}^T$ of EEG recordings where $\mathbf{X}_i$ is the recording of the i-th electrode and N is the number of electrodes in the EEG setup.

Neural electrical activity is tightly connected with the cognitive function that the brain performs. There are regions of the brain that demonstrate task-induced neural activity which is different from the one measured during the resting state, i.e. when no cognitive function is performed. Therefore, a more complete description of the process (equation 3.1) that governs electrical activity should include a "function" random variable f such that $\mathbf{X} \sim p(\mathbf{X}|f)$. The function variable can denote, for example, whether or not the brain is performing listening of an external sound. Often, brain diseases such as schizophrenia correspond to structural and functional abnormalities in the brain which, in turn, affect neural activity. Hence, to account for brain malfunctions, we should incorporate a "disease" random variable d in the governing distribution yielding $\mathbf{X} \sim p(\mathbf{X}|f, d)$.

The altered neural activity allows one to diagnose brain diseases by analyzing neuroimaging data. Additionally, it has been observed that spatially distant brain regions tend to be synchronized in their activity. Moreover, the resulting synchronization patterns (so-called functional connectivity) are affected by a patient's condition as well, which gave rise to the development of the field of functional neuroimaging. As well as electrical activity, the co-activation networks also vary when performing different functions or being at a resting state. In the case of EEG, those networks are usually reconstructed [11] by measuring connectivity for each pair of electrodes in an EEG setup. Due to the importance of the functional connectivity networks for the diagnosis of brain diseases, we consider them as another observed variable:

$$\mathbf{X}, \mathbf{A} \sim p(\mathbf{X}, \mathbf{A}|f, d) \tag{3.2}$$

where we represent the network topology via adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$. This leads to defining the observed data as a graph $G = (V, E, \mathbf{X})$ where the set of nodes V and the set of edges E are derived from the adjacency matrix $\mathbf{A}$, and $\mathbf{X}$ is a node feature matrix. Hence, V represents the set of brain regions covered by electrodes of the EEG setup, E is the set of edges that connect nodes corresponding to neurally synchronised brain regions. Each node of the graph carries a vector of features $\mathbf{X}_i$ which is electrical recording in our case. Furthermore, the matrix $\mathbf{X}$ of EEG recordings defines the node feature matrix.

It should be noted that even though the functional connectivity is calculated based on the electrical activity, we assume that the causal relationship between these variables goes in the opposite direction. It is motivated by the fact that the co-activation of brain regions is governed by some underlying global process in the brain. In other words, there is a mechanism that defines which regions of the brain will be incorporated into performing a cognitive function and which will not. It can be seen as connecting the brain regions from the first cohort while keeping the second one disconnected.

3.3. Conditional Graph Variational Auto-encoder

We are interested in the whole process of data generation, thus we would like to find a model distribution $p_\theta(\mathbf{X}, \mathbf{A}, f, d)$ with parameters θ that approximates the true joint distribution $p(\mathbf{X}, \mathbf{A}, f, d)$ as well as possible. Besides, we aim at learning causal factorization of the joint distribution which will further allow us to investigate the factors. Finally, we want to utilize label information in the form of supervision bias to increase the quality of the approximation.

To fulfil each criterion, we expand the model to a latent variable model which we will learn in the framework of CCVAE. In other words, the resulting generative model will be trained along with the corresponding inference model approximating the intractable posterior with the AEVB algorithm. To account for each observed variable we partition the latent space into two subspaces $\mathbf{Z}$ and $\boldsymbol{\omega}$, where $\mathbf{Z}$ encodes electrical activity of the brain and the latent sub-space $\boldsymbol{\omega}$ encodes functional connectivity topology. As in the framework of CCVAE, we partition each of the sub-spaces into characteristic latent sub-space and general latent sub-space. Here, the characteristic latent sub-space captures characteristics of data associated with labels $\mathbf{y} = \{f, d\}$ and the general sub-space accounts for general features of data. Note that latent representation is calculated for each electrode's recording, hence we summarize the whole EEG recording in a matrix $\mathbf{Z}$. Thus, we define the factorization of our model distribution as follows:

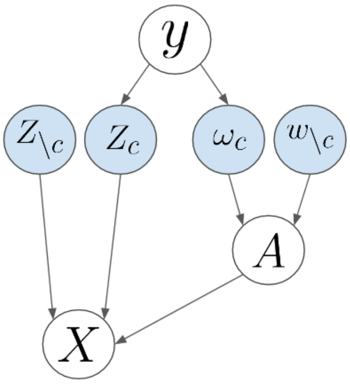


Figure 3.1.: Proposed graphical model. Latent variables are denoted with blue color and observed variables with white color.

$$p_\theta(\mathbf{X}, \mathbf{A}, \mathbf{y}, \mathbf{Z}, \boldsymbol{\omega}) = p_\theta(\mathbf{X}|\mathbf{Z}, \mathbf{A})\, p_\theta(\mathbf{A}|\boldsymbol{\omega})\, p_\theta(\mathbf{Z_c}|\mathbf{y})\, p_\theta(\boldsymbol{\omega_c}|\mathbf{y})\, p(\boldsymbol{\omega}_{\backslash \mathbf{c}})\, p(\mathbf{Z}_{\backslash \mathbf{c}})\, p(\mathbf{y}) \tag{3.3}$$

where θ are learnable parameters of the model. It corresponds to the probabilistic graphical model shown in Fig.3.1. Here, each label $\mathbf{y}^i$ is connected only to the corresponding subset of latent variables $\{\mathbf{Z_c}^i, \boldsymbol{\omega}_\mathbf{c}^i\}$. It allows for disentangling generative factors in the latent space by effectively isolating them from each other which is consistent with the ICM Principle discussed in Section 2.6.1.

As in the case of vanilla VAE, the posterior distribution $p_\theta(\mathbf{Z}, \boldsymbol{\omega}|\mathbf{X}, \mathbf{A}, \mathbf{y})$ is intractable. We approximate it with the following inference model:

$$q_\phi(\mathbf{Z},\boldsymbol{\omega}|\mathbf{X},\mathbf{A},\mathbf{y}) = \frac{q_\phi(\mathbf{y}|\mathbf{Z_c},\boldsymbol{\omega_c})\, q_\phi(\boldsymbol{\omega}|\mathbf{A})\, q_\phi(\mathbf{Z}|\mathbf{X})}{q_\phi(\mathbf{y}|\mathbf{X},\mathbf{A})} \tag{3.4}$$

where we define the classifier term $q_\phi(\mathbf{y}|\mathbf{X},\mathbf{A})$ as

$$q_\phi(\mathbf{y}|\mathbf{X},\mathbf{A}) = \int\int q_\phi(\mathbf{y}|\mathbf{Z_c},\boldsymbol{\omega_c})\, q_\phi(\boldsymbol{\omega}|\mathbf{A})\, q_\phi(\mathbf{Z}|\mathbf{X})\, d\mathbf{Z}\, d\boldsymbol{\omega} \tag{3.5}$$

3.3.1. Model objective

To simultaneously learn parameters θ and ϕ of generative and inference models, we use the AEVB algorithm and maximize the evidence lower bound:

$$\mathcal{L}_{\theta,\phi}(\mathbf{X},\mathbf{A},\mathbf{y}) = \mathbb{E}_{\mathbf{Z},\boldsymbol{\omega}\sim q_\phi(\mathbf{Z},\boldsymbol{\omega}|\mathbf{X},\mathbf{A},\mathbf{y})}\left[log\, p_\theta(\mathbf{X},\mathbf{A},\mathbf{y},\mathbf{Z},\boldsymbol{\omega}) - log\, q_\phi(\mathbf{Z},\boldsymbol{\omega}|\mathbf{X},\mathbf{A},\mathbf{y})\right]$$

which is equivalent to maximizing the following objective (see Appendix A.2 for derivation details):

$$\begin{aligned}\mathcal{L}(\mathbf{X},\mathbf{A},\mathbf{y}) = &\mathbb{E}_{\mathbf{Z}\sim q_\phi(\mathbf{Z}|\mathbf{X})}\Big[\mathbb{E}_{\boldsymbol{\omega}\sim q_\phi(\boldsymbol{\omega}|\mathbf{A})}\Big[\frac{q_\phi(\mathbf{y}|\mathbf{Z_c},\boldsymbol{\omega_c})}{q_\phi(\mathbf{y}|\mathbf{X},\mathbf{A})}\\ &\times\Big\{log\, p_\theta(\mathbf{X}|\mathbf{Z},\mathbf{A}) - \big(log\, q_\phi(\mathbf{Z}|\mathbf{X}) - log\, p_\theta(\mathbf{Z_c}|\mathbf{y}) - log\, p(\mathbf{Z}_{\setminus\mathbf{c}})\big)\\ &+ log\, p_\theta(\mathbf{A}|\boldsymbol{\omega}) - \big(log\, q_\phi(\boldsymbol{\omega}|\mathbf{A}) - log\, p_\theta(\boldsymbol{\omega_c}|\mathbf{y}) - log\, p(\boldsymbol{\omega}_{\setminus c})\big)\\ &- log\, q_\phi(\mathbf{y}|\mathbf{Z_c},\boldsymbol{\omega_c})\Big\}\Big]\Big] + log\, q_\phi(\mathbf{y}|\mathbf{X},\mathbf{A}) + log\, p(\mathbf{y})\end{aligned} \tag{3.6}$$

One can notice that expressions in the second and the third lines resemble the objective of vanilla VAE (see eq. 2.20). Particularly, the first term in both of them is a log-likelihood which is maximized when the reconstruction of a latent representation of a data point is exact. The second term plays the same role of KL divergence. Namely, it pressures an inference model to approximate a conditional generative model thus leading to latent representation either approximating a predefined prior distribution (for general latent variables) or capturing label-related features from data (for characteristic latent variables).

The formulation of the inference model as in equation 3.4 leads to the classification term $log\, q_\phi(\mathbf{y}|\mathbf{X},\mathbf{A})$ arising in the final objective. When computed as in equation 3.5, the term is effectively the composition of mappings forming a learnable mapping from input data $(\mathbf{X},\mathbf{A})$ to labels $\mathbf{y}$ that goes through the characteristic part of latent space.

It is also possible to adjust the strength of the classifier term by multiplying it by a constant α. The constant would either increase or decrease the pressure that the model applies to the latent space. We experimented with it and found that high values of α decrease the time of learning by additionally forcing the encoder to obtain label-related information from data.

3.3.2. Introducing β-regularization

Motivated by β-VAE [39], we introduce additional constrains on terms that mimic the KL divergence between an approximate posterior and a conditional prior of corresponding latent variables. Precisely speaking, we scale them by a factor $\beta > 1$ yielding the final objective of CGVAE:

$$\begin{aligned}\mathcal{L}(\mathbf{X},\mathbf{A},\mathbf{y}) = &\,\mathbb{E}_{\mathbf{Z}\sim q_\phi(\mathbf{Z}|\mathbf{X})}\Big[\mathbb{E}_{\boldsymbol{\omega}\sim q_\phi(\boldsymbol{\omega}|\mathbf{A})}\Big[\frac{q_\phi(\mathbf{y}|\mathbf{Z_c},\boldsymbol{\omega}_c)}{q_\phi(\mathbf{y}|\mathbf{X},\mathbf{A})}\\ &\times\Big\{log\, p_\theta(\mathbf{X}|\mathbf{Z},\mathbf{A}) - \beta\cdot\ \big(log\, q_\phi(\mathbf{Z}|\mathbf{X}) - log\, p_\theta(\mathbf{Z_c}|\mathbf{y}) - log\, p(\mathbf{Z}_{\backslash\mathbf{c}})\big)\\ &+ log\, p_\theta(\mathbf{A}|\boldsymbol{\omega}) - \beta\cdot\ \big(log\, q_\phi(\boldsymbol{\omega}|\mathbf{A}) - log\, p_\theta(\boldsymbol{\omega}_c|\mathbf{y}) - log\, p(\boldsymbol{\omega}_{\backslash c})\big)\\ &- log\, q_\phi(\mathbf{y}|\mathbf{Z_c},\boldsymbol{\omega}_c)\Big\}\Big]\Big] + \alpha\cdot log\, q_\phi(\mathbf{y}|\mathbf{X},\mathbf{A}) + log\, p(\mathbf{y})\end{aligned} \tag{3.7}$$

When $\beta > 1$, it applies additional pressure on an inference model to resemble a prior distribution. The extra pressure is assumed to lead to the objective encouraging conditional independence in the inference models and hence enforces higher disentanglement of the latent representation [39]. It is, however, done at the cost of reduced capacity of latent space and thus neglecting high-frequency details that are not able to pass through the constrained latent bottleneck. The optimal trade-off thus leads to learning the most efficient representation that is able to sufficiently preserve information while encapsulating characteristics of independent factors of variation.

We apply the additional pressure on an inference model as we are going to use the generative model for tasks that include sampling, e.g. conditional sampling from $p_\theta(\boldsymbol{\omega}|\mathbf{A})$. Hence, we want the inference model to capture those features of data that we would be able to reliably reproduce afterwards with a generative model. For example, it can be the case that a learned approximate posterior distribution $q_\phi(\boldsymbol{\omega}|A)$ reproduces high-frequency details from data but is not structured as a Gaussian distribution, e.g. it has multiple peaks. In this case, using a mode of the prior distribution during sampling can lead to misleading reconstruction results.

3.3.3. Computational details

In the original CCVAE approach, $\frac{q_\phi(\mathbf{y}|\mathbf{Z_c},\boldsymbol{\omega_c})}{q_\phi(\mathbf{y}|\mathbf{X},\mathbf{A})}$ is estimated via logarithms of corresponding probabilities:

$$\frac{q_\phi(\mathbf{y}|\mathbf{Z_c},\boldsymbol{\omega_c})}{q_\phi(\mathbf{y}|\mathbf{X},\mathbf{A})} = e^{log\, q_\phi(\mathbf{y}|\mathbf{Z_c},\boldsymbol{\omega_c}) - log\, q_\phi(\mathbf{y}|\mathbf{X},\mathbf{A})} \tag{3.8}$$

where logarithm of the classifier term 3.5 is estimated by the Monte-Carlo method:

$$\begin{aligned} log\, q_\phi(\mathbf{y}|\mathbf{X},\mathbf{A}) &= log \int\int q_\phi(\mathbf{y}|\mathbf{Z_c},\boldsymbol{\omega_c})\, q_\phi(\boldsymbol{\omega}|\mathbf{A})\, q_\phi(\mathbf{Z}|\mathbf{X})\, d\mathbf{Z}\, d\boldsymbol{\omega} \\ &\approx log\ \mathbb{E}_{\boldsymbol{\omega}\sim q_\phi(\boldsymbol{\omega}|\mathbf{A}),\mathbf{Z}\sim q_\phi(\mathbf{Z}|\mathbf{X})} \left[q_\phi(\mathbf{y}|\mathbf{Z_c},\boldsymbol{\omega_c})\right] \\ &= log\ \mathbb{E}_{\boldsymbol{\omega}\sim q_\phi(\boldsymbol{\omega}|\mathbf{A}),\mathbf{Z}\sim q_\phi(\mathbf{Z}|\mathbf{X})} \left[e^{log\, q_\phi(\mathbf{y}|\mathbf{Z_c},\boldsymbol{\omega_c})}\right] \end{aligned}$$

Though straightforward, this formulation leads to the gradients of the classifier parameters suffering from a high variance during training. In [10], authors managed to reduce the problem by not reparameterizing label-related latent variables $\boldsymbol{Z_c}, \boldsymbol{\omega_c}$ when estimating the fraction.

In our case, the trick did not work and the high variance of parameters' gradients remained an issue which did not allow one to train a model. To tackle it, we assume that for a trained model the following holds:

$$log\, q_\phi(\mathbf{y}|\mathbf{Z_c},\boldsymbol{\omega_c}) \approx log\ \mathbb{E}_{\boldsymbol{\omega}\sim q_\phi(\boldsymbol{\omega}|\mathbf{A}),\mathbf{Z}\sim q_\phi(\mathbf{Z}|\mathbf{X})} \left[q_\phi(\mathbf{y}|\mathbf{Z_c},\boldsymbol{\omega_c})\right]$$

Thus, we can write the Taylor expansion of exponent in equation 3.8 as follows:

$$\begin{aligned} \frac{q_\phi(\mathbf{y}|\mathbf{Z_c},\boldsymbol{\omega_c})}{q_\phi(\mathbf{y}|\mathbf{X},\mathbf{A})} &= e^{log\, q_\phi(\mathbf{y}|\mathbf{Z_c},\boldsymbol{\omega_c}) - log\, q_\phi(\mathbf{y}|\mathbf{X},\mathbf{A})} \\ &\approx \sum_{n=0}^{+\infty} \frac{(log\, q_\phi(\mathbf{y}|\mathbf{Z_c},\boldsymbol{\omega_c}) - log\, q_\phi(\mathbf{y}|\mathbf{X},\mathbf{A}))^n}{n!} \end{aligned} \tag{3.9}$$

We experimented with the number of terms N during training and found that estimating the term with $N = 0$, i.e.

$$\frac{q_\phi(\mathbf{y}|\mathbf{Z_c},\boldsymbol{\omega_c})}{q_\phi(\mathbf{y}|\mathbf{X},\mathbf{A})} = 1, \tag{3.10}$$

not surprisingly yields the lowest gradient norm of the classifier parameters. Moreover, it still leads to the model successfully learning expected structure of the latent space (see Section 4.2.2) along with the classifier term's weights. Hence, we substitute the term with 1 when optimizing the objective 3.6.

3.4. Exploring generative factors

Generally, in the field of neuroimaging, one is interested in disease-associated characteristics in data, e.g. how schizophrenia affects functional connectivity during listening. If the mechanism is unknown, it can be inferred by exploring learned generative factors that are related to the disease label.

One possible way to discover the meaning of a learned generative factor is through manipulating it and observing realization in electrical activity space and functional connectivity space. Assume we are aimed at investigating the meaning of a label $\mathbf{y}_i$ with respect to functional connectivity. To do so, we fix values of every other label in $\mathbf{y}$ and randomly vary the value of $\mathbf{y}_i$, e.g. setting it equal to 0 or 1 if $\mathbf{y}_i$ is binary. Afterwards, a latent representation is sampled $\boldsymbol{\omega}^{\boldsymbol{i}} \sim p_\theta(\boldsymbol{\omega}_c^{\boldsymbol{i}}|\mathbf{y}^{\mathbf{i}})$ and modes of remaining prior distributions are used to derive the values of other latent variables. We further reconstruct the latent representation by sampling from $p_\theta(\mathbf{A}|\boldsymbol{\omega})$. Doing it multiple times allows one to calculate variation for each connection in $\mathbf{A}$ which would indicate whether a set of connections is relevant for the label or not, i.e. is affected by the label or not.

4. Experimental Results

In this chapter, we demonstrate how the proposed model can be used for studying functional connectivity reconstructed from EEG data. In this case, electrical activity is recorded for 3 groups of people: healthy controls, schizophrenia patients and people suffering from schizophrenia with AVH. Each group was performing a dichotic listening task, i.e. auditory signals were given simultaneously in each ear.

First, we define experimental details: parametrization of conditional distributions, the model architecture, optimization details and experimental data. Then, we show the reconstruction ability of the model and explore learned generative factors of variation with the algorithm described in 3.4.

4.1. Experimental details

4.1.1. Parametrization of conditional distributions

Figuratively, we can sort every distribution in generative 3.3 and inference 3.4 models into two groups. Each group corresponds to the family of distributions which we assume a distribution to belong to:

Multivariate Gaussian: $p_\theta(\mathbf{X}|\mathbf{Z},\mathbf{A})$, $q_\phi(\mathbf{Z}|\mathbf{X})$, $p_\theta(\mathbf{Z_c}|\mathbf{y})$, $p_\theta(\mathbf{A}|\boldsymbol{\omega})$, $q_\phi(\boldsymbol{\omega}|\mathbf{A})$, $p_\theta(\boldsymbol{\omega_c}|\mathbf{y})$, $p(\mathbf{Z}_{\setminus\mathbf{c}})$, $p(\boldsymbol{\omega}_{\setminus c})$

Multivariate Bernoulli: $q_\phi(\mathbf{y}|\mathbf{X},\mathbf{A})$, $q_\phi(\mathbf{y}|\mathbf{Z_c},\boldsymbol{\omega_c})$, $p(\mathbf{y})$

In the case of conditional distributions, every parameter (μ,σ for the first group and a λ for the second group) is defined as a deterministic function of corresponding input variables which we approximate with neural networks.

Encoder and Decoder (Z-space): We implement the likelihood distribution $p_\theta(\mathbf{X}|\mathbf{Z},\mathbf{A})$ $= \mathcal{N}(\mathbf{X};\boldsymbol{\mu}_\theta(\mathbf{Z},\mathbf{A}), diag(\boldsymbol{\sigma}^2_\theta))$ using graph neural networks. Precisely, deterministic function $\boldsymbol{\mu}(\mathbf{X}|\mathbf{Z},\mathbf{A})$ takes the computational graph given by the adjacency matrix $\mathbf{A}$ with the matrix of latent node features $\mathbf{Z}$ and calculates the matrix of EEG recordings $\mathbf{X}$ via multiple steps of learned message-passing. Each round of message-passing is computed using the graph convolutional operator from Morris et al. [40]. We initialize variance of each signal point $\sigma_\theta \in \mathbb{R}$ as a separate parameter optimized during training with SGD. We assume the inference model $q_\phi(\mathbf{Z}|\mathbf{X}) = \mathcal{N}(\mathbf{Z};\boldsymbol{\mu}_\phi(\mathbf{X}), diag(\boldsymbol{\sigma}^2_\phi(\mathbf{X})))$

to be a multivariate Gaussian. We approximate its parameters with with 1D convolutional neural networks, which help us to calculate low-dimensional representation $\mathbf{Z}_i$ for each node with feature vector $\mathbf{X}_i$ independently from other nodes.

Encoder and Decoder (ω-space): We use 2D convolutional neural networks when implementing the generative and inference models which we represent with Normal distributions $p_\theta(\mathbf{A}|\boldsymbol{\omega}) = \mathcal{N}(\mathbf{A}; \boldsymbol{\mu}_\theta(\boldsymbol{\omega}), diag(\boldsymbol{\sigma}^2_\theta))$ and $q_\phi(\boldsymbol{\omega}|\mathbf{A}) = \mathcal{N}(\boldsymbol{\omega}; \boldsymbol{\mu}_\phi(\mathbf{A}), diag(\boldsymbol{\sigma}^2_\phi(\mathbf{A})))$ hence treating an adjacency matrix as an image. In the case of the inference model, 2D transposed convolution operators are utilized to reconstruct the image from a low-dimensional vector representation.

Classifier: We represent the label predictive distribution $q_\phi = Ber(\mathbf{y}; \lambda_\phi(\mathbf{Z_c}, \boldsymbol{\omega_c}))$ as a multivariate Bernoulli distribution with $\lambda_\phi(\mathbf{Z_c}, \boldsymbol{\omega_c})$ implemented as a diagonal transformation. It enforces the factorization of the model $q_\phi(\mathbf{y}|\mathbf{Z_c}, \boldsymbol{\omega_c}) = \prod_i q^i_\phi(\mathbf{y}_i|\mathbf{Z_c}{}^i, \boldsymbol{\omega}^i_c)$ which corresponds to disentangling causal variables related to labels. We concatenate $\vec{\mathbf{z}}_\mathbf{c}$ and $\boldsymbol{\omega_c}$ to form an input for the diagonal transformation, where $\vec{\mathbf{z}}_\mathbf{c}$ is a vectorized form of a latent feature matrix $\mathbf{Z}$.

Conditional Priors: The conditional priors are represented as multivariate Gaussians $p_\theta(\mathbf{Z_c}|\mathbf{y}) = \mathcal{N}(\mathbf{Z_c}; \boldsymbol{\mu}_\theta(\mathbf{y}), \mathbb{1})$ and $p_\theta(\boldsymbol{\omega_c}|\mathbf{y}) = \mathcal{N}(\boldsymbol{\omega_c}; \boldsymbol{\mu}_\theta(\mathbf{y}), \mathbb{1})$, where $\mathbb{1}$ is an identity matrix forcing independence of unlabeled latent variables. We approximate mean values by look-up tables, where each table returns $\mathbf{2}$ if $\mathbf{y}^i = 1$ and $-\mathbf{2}$ otherwise. We parameterize conditional priors such that latent variables corresponding to different labels are independent. It enforces disentanglement of label-related characteristic in latent space and is equivalent to factorization, e.g. in the case of $\boldsymbol{\omega}$, $p_\theta(\boldsymbol{\omega_c}|\mathbf{y})$ $= \prod_i p_\theta(\boldsymbol{\omega}^{\boldsymbol{i}}_{\boldsymbol{c}}|\mathbf{y^i})$.

Priors (unlabeled latents): Prior distributions of the unlabeled latent variables are assumed to follow multivariate Gaussian $p(\mathbf{Z}_{\backslash\mathbf{c}}) = \prod_i p(\mathbf{Z}^\mathbf{i}_{\backslash\mathbf{c}}) = \prod_i \mathcal{N}(\mathbf{Z}^\mathbf{i}_{\backslash\mathbf{c}}; \mathbf{0}, \mathbb{1})$ and $p(\boldsymbol{\omega}_{\backslash c}) = \mathcal{N}(\boldsymbol{\omega}_{\backslash c}; \mathbf{0}, \mathbb{1})$, where $\mathbb{1}$ is an identity matrix forcing independence of unlabeled latent variables. Prior distribution over labels is represented as a multivariate Bernoulli $p(\mathbf{y}) = \prod_i Bern(\mathbf{y}_i; \boldsymbol{\lambda})$. Parameters $\boldsymbol{\lambda}$ of $p(\mathbf{y})$ are calculated according to the proportion of corresponding label values in the data set.

Detailed description of the model architectures is given in Appendix A.3.

4.1.2. Experimental data

The study included 29 patients diagnosed with schizophrenia and 52 healthy controls (HC). Every participant was right-handed. Among 29 schizophrenia patients, 14 subjects belonged to the AVH group, i.e. schizophrenia patients hearing voices with no external stimuli presented. The remaining 15 people are assigned to the SZ group. Each participant was given one of six different syllables (/ba/, /da/, /ka/, /ga/, /pa/, /ta/) for 500 ms simultaneously to each ear after 1 s silence period. During the task, the EEG recording was conducted with 64 electrodes including 4 EOG channels to monitor eye movements. At the preprocessing step, the data was filtered from 1 to 20 Hz and downsampled to 500 Hz according to a protocol described in [41]. Afterwards, the common average was taken with further re-referencing of all channels. Besides, muscle and visual artefacts were identified and removed. Each recording was then divided into 2 parts: a resting state (first 500 ms with no syllable given) and a listening state (initial 500 ms when syllables were presented). Detailed information on data acquisition and preprocessing can be found in [41].

Initially, each data point is assigned to 2 classes: state (listening/resting) and disorder (HC/SZ/AVH). We reformulate the disorder classification to yield binary label description (see Table 4.1). Namely, we use the following set of labels: *state* $\in$ {resting, listening}, *schizophrenia* $\in \{0, 1\}$, *hallucinations* $\in \{0, 1\}$.

disorder	schizophrenia	hallucinations
HC	0	0
SZ	1	0
AVH	1	1

Table 4.1.: Reformulation of the disorder class (left column) to yield binary representation.

To create a dataset for training the model, we randomly sample from each group of people and each state group such that the number of instances of each class is equal. As the result, the final dataset contains 12000 EEG recordings which we split into training and test partitions with split ratios 0.8 and 0.2 respectively.

4.1.3. Recovering functional connectivity

We recover functional connectivity by calculating the Pearson correlation coefficient for each pair of electrodes which yields a weighted adjacency matrix $A \in [-1, 1]$. To filter out insignificant connections, we preprocess each element of an adjacency matrix as follows:

$$\hat{a}_{ij} = -0.1 + 0.009 \cdot 121^{a_{ij}} \tag{4.1}$$

This maps $A \in [-1, 1] \rightarrow \hat{A} \in [-0.1, 1]$. It can be seen as increasing the resolution for high-value range while significantly decreasing it for low-value range. We use it to make the model to focus on the edges with high connectivity measure value.

Generally, the framework does not apply any limitation on the choice of connectivity measure. We use the covariance adjacency matrices as they are easy to obtain without additional software. However, one can also use more advanced approaches such as Lagged Phase Synchronization or Phase-Locking Value [11] which is one possible direction for further work.

4.1.4. Optimization details

The parameters of every model are trained via optimizing the objective 3.6. We use Adam optimizer with a learning rate of 10^{-4}. The training was performed in mini-batches of size 32 for 100 epochs. All models are trained on a NVIDIA Tesla V100 GPU from the Taurus HPC system of the Technical University of Dresden.

4.2. Results & Discussion

For the experimental part, we trained a CGVAE model with $|\boldsymbol{\omega}| = 10$ and $|\mathbf{Z}| = 61 \times 10$. Each label was assigned with one dimension of each partition, thus $|\boldsymbol{\omega}_\mathbf{c}^{\boldsymbol{i}}| = 1, |\mathbf{Z}_\mathbf{c}^\mathbf{i}| = 61 \times 1 \ \forall i \in \{\textit{state}, \textit{schizophrenia}, \textit{hallucinations}\}$.

4.2.1. Classification of labels

We first demonstrate the ability of the approach to classify electrical activity and functional activity data with respect to an underlying brain malfunction. To do, we train a model with the non-regularized objective (See Table 4.2, $\beta_\omega = 1$, $\beta_Z = 1$). As one can see, it is able to reliably separate healthy people from schizophrenia patients with or without AVH yielding an exceptionally high performance. At the same time, it is not able to tell apart the resting state from the listening state as the resulting accuracy is not different from the one of a random guess. We hypothesize that it can be related to heavily entangled characteristics of the diagnosis labels and the state label in the data space. Thus, the model cannot identify a causal variable that would account for a state and be independent of variables related to diagnosis.

We further analyze the influence of β-regularization on the performance of the model. To do so, we gradually increase the values of both β_ω and β_Z and see how it effect performance metrics (See Table 4.2). As expected, the growth of β is followed by the increase of mean square error between a true data point and its reconstruction. It is not surprising since we know that by varying β we basically regulate the capacity of a latent representation, i.e. its capability to store high-frequency features of data. Along with the capacity, the difference between an approximate posterior distribution and a prior distribution is regulated which is measured by KL divergence (see 3rd and 4th columns). Overall, the model behaves according to our expectations when changing the value of β during training. In order to still use the generative model for recovering causal mechanisms, for further experiments we limit ourselves to only the listening part of data, i.e. recordings of the brain performing the auditory processing function.

	$\overline{(X-\hat{X})^2}$	$\overline{(A-\hat{A})^2}$	KL_ω	KL_Z	Accuracy (diagnosis)	Accuracy (state)
$\beta_\omega = 1, \beta_Z = 1$	12.6	6.5	27.2	702.6	1	0.5
$\beta_\omega = 2, \beta_Z = 2$	14.6	6.8	22.4	255.0	1	0.5
$\beta_\omega = 5, \beta_Z = 2$	27.0	10.7	4.0	245.0	1	0.5
$\beta_\omega = 5, \beta_Z = 5$	27.5	10.6	4.3	220.0	0.98	0.5
$\beta_\omega = 10, \beta_Z = 10$	28.0	11.5	2.0	37.8	0.65	0.5

Table 4.2.: Comparison on the test partition of the dataset over different values of β for multiple metrics. The first two columns correspond to data point-wise mean square error between true data points and reconstructed ones. They are followed by KL divergence between the inference model and a prior distribution for both latent variables $\boldsymbol{\omega}$ and $\mathbf{Z}$. The last two columns demonstrate the accuracy of classification.

4.2.2. The effect of β-regularization on latent space

Let us consider the latent space $\boldsymbol{\omega}$ corresponding to functional connectivity data. It is separated into two sub-spaces: the general sub-space and the characteristic sub-space. The first one follows the prior of choice, in our case a multivariate Gaussian: $\boldsymbol{\omega}_{\backslash c} \sim \mathcal{N}(\omega_{\backslash c}; \mathbf{0}, \mathbf{1})$. Meanwhile, the second one is governed by the conditional prior, e.g. for $\boldsymbol{\omega}$: $\boldsymbol{\omega_c} \sim p_\theta(\boldsymbol{\omega_c}|\mathbf{y})$, which causes the characteristic sub-space to reflect labels. CGVAE forces the approximate posterior distribution to resemble the prior distribution over latent space which leads to the encoder capturing label-related information from data. To demonstrate that, we sample from both distributions and depict resulting densities in Figure 4.1 (Left). Note that everything we said above is also applied to the latent space $\mathbf{Z}$, though we focus on $\boldsymbol{\omega}$ as it is easier to visualize.

One can immediately see the effect of incorporating label information into the latent space. Here, the dimension corresponding to binary labels demonstrate the decomposition into two different Gaussian distributions: $p_\theta(\boldsymbol{\omega}_c^i|\mathbf{y^i} = \mathbf{0})$ and $p_\theta(\boldsymbol{\omega}_c^i|\mathbf{y^i} = \mathbf{1})$ (see red distributions). In turn, it forces the approximate posterior to decompose accordingly (see blue distributions), thus encoding label-related characteristics in data. Note that the dimensions that belong to the general sub-space are unaffected by labels. It is a desired property since we want to disentangle generative factors related to labels from others.

It is also crucial to apply additional pressure to the inference model to resemble the prior which is demonstrated in Figure 4.1. One can see that with the high value of $\beta = 5$, the learned approximate posterior $q_\phi(\boldsymbol{\omega}|\mathbf{A})$ is much smoother than in the case of $\beta = 1$. This heavily affects the quality of performing inference tasks.

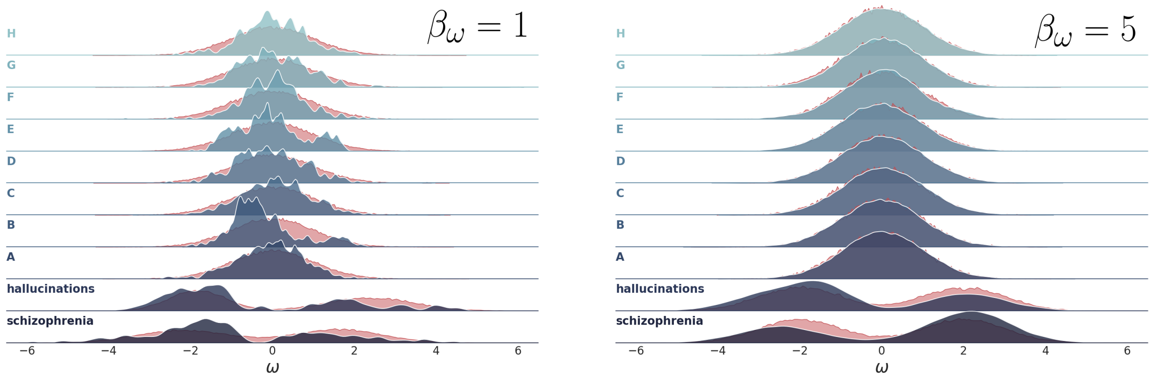


Figure 4.1.: The effect of the β-regularization of the CGVAE objective on the learned approximate posterior distribution $q_\phi(\boldsymbol{\omega}|\mathbf{A})$ (blue). Prior distribution over latent space $p_\theta(\boldsymbol{\omega}|\mathbf{y})$ is depicted with red color. Indicators on the left denote whether a dimension is assigned by a label (*schizophrenia*, *hallucinations*) or not (capital letters).

Imagine, for example, if we want to estimate the most probable value of $\boldsymbol{\omega}$ given a certain combination of labels $\mathbf{y}$. It is equivalent to calculating a mode of the conditional distribution $p_\theta(\boldsymbol{\omega}|\mathbf{y})$. However, if we simply take the mean value of the distribution, it will generate a wrong result for the model trained with $\beta = 1$. Indeed, since the generative model is trained to reconstruct a data point from its latent representation learned by an inference model, a high difference in the approximate posterior and the prior will manifest itself in an unexpected reconstruction. In this sense, a regularized model is superior as its inference model generates a latent distribution closely resembling the prior distribution.

Overall, the high values of β during training lead to learning more generalizable and hence robust latent representation. As we have discussed, it will significantly improve the quality of sampling and inference tasks hence we utilize a model with $\beta_\omega = 5$ and $\beta_Z = 2$ for further experiments.

4.2.3. Reconstruction from latent space

We also demonstrate the ability of the model to reconstruct a data point in the form of a graph $G = (\mathbf{X}, \mathbf{A})$. To do so, we first obtain a latent representation of the graph $G_{latent} = (\mathbf{Z}, \boldsymbol{\omega})$ via the encoders $q_\phi(\mathbf{Z}|\mathbf{X})$ and $q_\phi(\boldsymbol{\omega}|\mathbf{A})$. Afterwards, we reconstruct the latent graph back with the decoders $p_\theta(\mathbf{X}|\mathbf{Z}, \mathbf{A})$ and $p_\theta(\mathbf{A}|\boldsymbol{\omega})$. Results of reconstruction are shown in Figure 4.2 (for more examples see Appendix B.1, Figures B.1, B.2).

Speaking about electrical activity data, the model is able to capture low-frequency

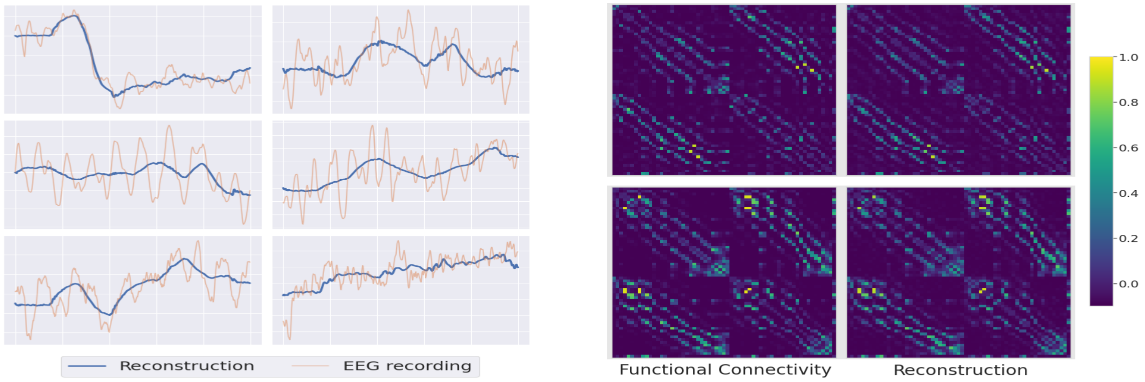


Figure 4.2.: The ability of the model to reconstruct EEG recordings (Left) and functional connectivity matrices (Right) based on their latent representations $\mathbf{Z}$ and $\boldsymbol{\omega}$. Functional connectivity is reconstructed with Pearson correlation coefficient between corresponding node signals. We further highlight strong connections by applying a non-linear mapping 4.1.

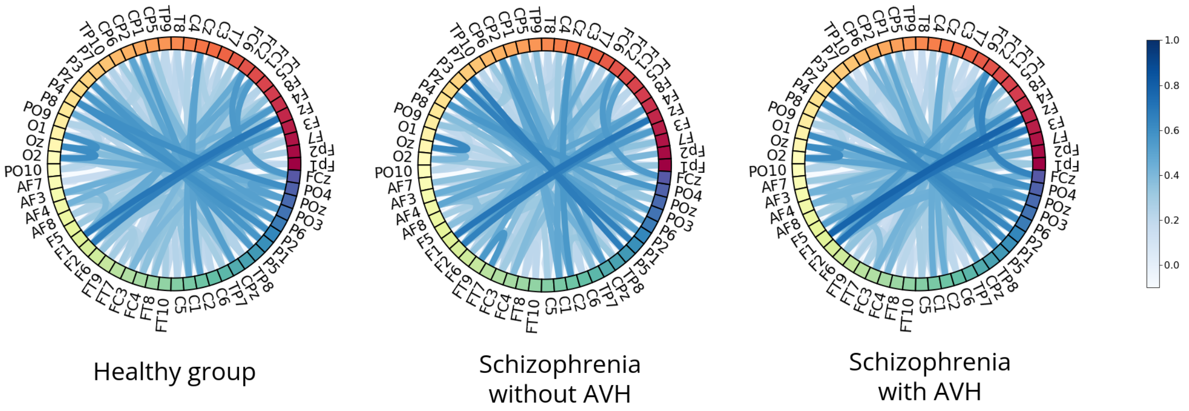
modes in a recording. It is sufficient to reproduce auditory evoked potentials, i.e. responses of a brain on auditory stimulation. For example, one can see the sharp decrease in the signal in the left upper corner of Figure 4.2. It corresponds to auditory $N100$ - a negative evoked potential peaking at around 100 ms which the model is able to capture.

Functional connectivity matrices are also reconstructed precisely from their latent representations. Even though we treat those matrices as images and implement a decoder with convolutional neural networks, there is no smearing of the high-frequency signal. It is particularly important for inference tasks, as the model should reproduces connections with high connectivity score (depicted in yellow).

Overall, the model is able to encapsulate key characteristics of both EEG data and functional connectivity matrices in its latent space. If the KL divergence between an approximate posterior and a prior distribution is small, we now can sample from a latent space and produce reconstructions in data space that will share features of data points from the original data set.

4.2.4. Most likely functional networks

A trained generative model allows one to solve inference tasks. For example, we are interested in the following question: given a brain disorder and a function the brain performs, what is the most likely functional connectivity? From the probabilistic point of view, the question boils down to estimating a mode of the posterior distribution $\hat{\mathbf{A}} = \arg\max_{\mathbf{A}}\, p_\theta(\mathbf{A}|\mathbf{y})$. Since the distribution can be factorized as $p_\theta(\mathbf{A}|\boldsymbol{\omega})\, p_\theta(\boldsymbol{\omega_c}|\mathbf{y})$, it is equivalent to estimating modes of each factor: $\hat{\omega} = \arg\max_{\omega} p_{\theta}(\omega|\mathbf{y})$ and $\hat{\mathbf{A}} = \arg\max_{\mathbf{A}} p_{\theta}(\mathbf{A}|\hat{\omega})$. We assume each of the distribution to be a multivariate Gaussian, hence each mode is simply the mean value of the corresponding conditional distribution. It is crucial for the task to have a conditional prior that is resembled by an inference model as close as possible. It guarantees the reliability of latent representation via learning the features of data that are distributed according to our expectations (i.e. multivariate Gaussians in our case). We demonstrate the most likely functional connectivity for each disorder in Figure 4.3.



Healthy group

Schizophrenia without AVH

Schizophrenia with AVH

Figure 4.3.: Most likely functional connectivity for healthy people and patients suffering from schizophrenia with or without AVH. Functional connectivity is reconstructed with Pearson correlation coefficient between corresponding node signals. We further highlight strong connections by applying a non-linear mapping 4.1.

Overall, the correlation matrices are similar and demonstrate high correlation between electrodes located in frontal, parietal and occipital lobes. It is yet difficult to see the alterations that the malfunctions induce on the functional connectivity with respect to the healthy group.

4.2.5. Exploring generative factors of data

We further try to reconstruct meaning of label related latent-variables by using the algorithm described in Section 3.4. The algorithm boils down to intervening upon a single label multiple times, sampling a latent representation, reconstructing it and calculating variance of correlation value for each connection. We argue that high variance indicates those connection that are controlled by the corresponding generative factor. For instance, to obtain an explanation for the *schizophrenia* label, we fix the value of $\mathbf{y}^{hallucinations} = 0$. Besides, we keep every dimension of the latent space fixed but randomly sample $\boldsymbol{\omega}^{schizophrenia} \sim p_\theta(\boldsymbol{\omega}^{schizophrenia}|\mathbf{y}^{schizophrenia})$. The same approach is used for the *hallucinations* label where we keep $\mathbf{y}^{schizophrenia} = 1$. We provide connections with the highest variance for each group in Table 4.3 (see Appendix B.2, Figure B.3 for full visualization).

Intervening on *schizophrenia*			Intervening on *hallucinations*		
connection	brain regions	σ^2	connection	brain regions	σ^2
FT7 ↔ FT9	BA47L*, BA20L*	0.046	PO3 ↔ O1	BA19L, BA18L	0.044
T7 ↔ FT9	BA42L*, BA20L*	0.045	PO3 ↔ POz	BA19L, BA17L	0.040
P7 ↔ P5	BA37L, BA39L*	0.034	P3 ↔ P5	BA39L, BA39L	0.035
P3 ↔ P5	BA39L*, BA39L*	0.033	POz ↔ Oz	BA17L, BA17R	0.029
T7 ↔ FT7	BA42L*, BA47L*	0.024	PO3 ↔ Oz	BA19L, BA17R	0.028

Table 4.3.: Variance in reconstructions of functional connectivity matrices when intervening on a single label: *schizophrenia* (Left) and *hallucinations* (Right). For each label, 5 connections are taken that have the highest variance of the correlation value. Each connection is provided with a pair of brain regions that it encompasses. Brain regions that are found to be involved in auditory tasks for healthy people are denoted with *.

In the tables, we denote regions of the brain that were previously found to demonstrate auditory-evoked activity with *. It allows us to do a sanity check as we expect those regions to be highlighted by the model. According to our expectations, intervening on *schizophrenia* label causes variance in connections between electrodes that cover either the temporal lobe including the auditory cortex ($T7$, $FT7$) or the part of the parietal lobe that encompass Wernicke's area ($P5$, $P3$). Both regions play the central role in auditory processing and comprehension which proves the ability of the model to learn meaningful label-related generative factors from data. It is worth noting that every connection listed is located in the left hemisphere. It is not surprising as it is known that the auditory function is left-lateralized for right-handed people.

It is, however, more difficult to judge the sanity of intervening on the *hallucinations* label, as underlying mechanisms of AVH are mostly unknown. The model is explaining it by alterations of functional connectivity in parietal and occipital lobes which current evidence in the field of neuroimaging does not support. At the same time, EEG might be not the best modality to study the phenomena. There is currently growing evidence supporting the interhemispheric miscommunication theory that point toward abnormalities in interhemispheric auditory pathways that lead to the emergence of AVH. Yet those pathways cross regions that are located deep in the brain thus making it difficult to capture by recording electrical activity from the cortex only. It gives another direction for further work, namely to apply the framework for different neuroimaging data, e.g. fMRI that allow studying neural synchronization on full depth.

4.2.6. Limitations of the approach

We would like to mention limitations in the proposed approach that should be addressed in future work to increase its reliability and interpretability. First of all, in spite of the excellent classification performance with respect to a mental disorder, the model was not able to differentiate resting state from listening state. It does not allow the application of the model to the analysis of temporal dynamics of functional connectivity which is more challenging yet more informative compared to static connectivity.

Second, one has to manually enforce regularization of the objective 3.6 to ensure that the latent space learned by an inference model is close to a pre-defined prior. The point is crucial to perform inference tasks with the generative model. However, the current formulation of the objective forces one to visually estimate the "goodness" of the fit which is burdensome and gets more complicated when the dimensionality of latent space is high.

The next point we have already mentioned before - electrical activity can be at most a sub-optimal data source for the analysis of some brain malfunctions. Even though the technique does have its advantages including the superior temporal resolution, it only records the neural activity of the brain cortex. Consequently, it does not allow to analyse those regions of the brain that are located beneath the cortex and whose signals are lost in surrounding noise. It might lead to extracting misleading explanations for diseases that are caused by alterations in functional connectivity in those regions. To address this problem, we should carefully choose the appropriate modality for a given disease. Yet the model is easily generalizable for many neuroimaging techniques such as fMRI or PET.

The last, we approached exploring learned generative factors of functional connectivity data by estimating variation for each connection in response to an intervention upon a label of choice. This approach is straightforward and easy to implement, but it does not give the full information about the meaning of a factor. It gives another direction of research, namely interpreting causal variables that should positively affect the applicability of the model for decision-making support.

5. Conclusion

We presented a novel approach for a joint analysis of neuroimaging data and functional connectivity networks based on the framework of variational auto-encoders. Its development was mainly motivated by the data-driven search of reliable biomarkers for robust diagnosing brain diseases. The method efficiently combines a classification model and a generative model which allows one to simultaneously differentiate between multiple diseases and recover causal mechanisms related to them. We further demonstrate the applicability of the proposed model to the identification of schizophrenia either followed or not by auditory verbal hallucinations based on EEG recordings. The approach yields exceptionally high accuracy on the test data set. Furthermore, we analyze disorder-related learned factors of data generation and illustrate that they manifest themselves in functional connectivity data according to our expectations.

There are multiple possible directions for further work:

- We did not manage to reliably classify EEG recordings taken from different time points thus working with static functional connectivity data. If implemented, it opens the door for application in the analysis of dynamic functional connectivity.
- The framework can be applied to different neuroimaging techniques such as fMRI or PET that might be more suitable for studying particular brain diseases.
- We explored the meaning of each generative factor by simply observing what regions of a data point are affected when intervening on a corresponding latent variable. It, however, provides limited information and more advanced approaches from probability theory can be applied to the problem, e.g. mutual information.
- We apply regularization on the framework objective to enforce encapsulating reliable features of data in latent variables. It makes the process of learning burdensome as one has to manually vary the strength of the regularization and observe the effect imposed on latent space. Developing more advanced regularization techniques might help to avoid it as well as improve the interpretability of learned data generative factors.

A. Appendix

A.1. Notation

Random scalars are denoted with lower case letters, e.g. x, y, z. We denote random vectors with bold lower case letters, e.g. $\mathbf{x}, \boldsymbol{\omega}$. Random matrices are written with bold and capitalized letters such as, for example, $\mathbf{X}$, $\mathbf{Z}$. The i-th row of a random matrix is denoted with subscript, e.g. $\mathbf{X_i}$.

A.2. Derivation of ELBO

We are interested in obtaining an expression for a evidence lower bound when a generative function is given as in equation 3.3 and an inference model as in equation 3.4. Using the definition of ELBO 2.16 yields

$$\mathcal{L}_{\theta,\phi}(\mathbf{X},\mathbf{A},\mathbf{y}) = \mathbb{E}_{\mathbf{Z},\boldsymbol{\omega}\sim q_\phi(\mathbf{Z},\boldsymbol{\omega}|\mathbf{X},\mathbf{A},\mathbf{y})}\left[log\, p_\theta(\mathbf{X},\mathbf{A},y,\mathbf{Z},\boldsymbol{\omega}) - log\, q_\phi(\mathbf{Z},\boldsymbol{\omega}|\mathbf{X},\mathbf{A},\mathbf{y})\right]$$

We further factorize the model distributions according to equations 3.3 and 3.4:

$$\mathcal{L}_{\theta,\phi}(\mathbf{X},\mathbf{A},\mathbf{y}) = \mathbb{E}_{\mathbf{Z},\boldsymbol{\omega}\sim q_\phi(\mathbf{Z},\boldsymbol{\omega}|\mathbf{X},\mathbf{A},\mathbf{y})}\left[log\, \frac{p_\theta(\mathbf{X}|\mathbf{Z},\mathbf{A})\; p_\theta(A|\boldsymbol{\omega})\; p_\theta(\mathbf{Z}|\mathbf{y})\; p_\theta(\boldsymbol{\omega}|\mathbf{y})\; p(\mathbf{y})}{\frac{q_\phi(\mathbf{y}|\mathbf{Z_c},\boldsymbol{\omega}_c)\; q_\phi(\boldsymbol{\omega}|\mathbf{A})\; q_\phi(\mathbf{Z}|\mathbf{X})}{q_\phi(\mathbf{y}|\mathbf{X},\mathbf{A})}}\right]$$

where $p_\theta(\mathbf{Z}|\mathbf{y}) = p_\theta(\mathbf{Z_c}|\mathbf{y})\; p(\mathbf{Z}_{\backslash\mathbf{c}})$ and $p_\theta(\boldsymbol{\omega}|\mathbf{y}) = p_\theta(\boldsymbol{\omega}_c|\mathbf{y})\; p(\boldsymbol{\omega}_{\backslash c})$. By the definition of expected value we obtain

$$\begin{aligned}
\mathcal{L}_{\theta,\phi}(\mathbf{X},\mathbf{A},\mathbf{y}) &= \int_{\boldsymbol{\omega}}\int_{\mathbf{Z}} \frac{q_\phi(\mathbf{y}|\mathbf{Z_c},\boldsymbol{\omega}_c)\; q_\phi(\boldsymbol{\omega}|\mathbf{A})\; q_\phi(\mathbf{Z}|\mathbf{X})}{q_\phi(\mathbf{y}|\mathbf{X},\mathbf{A})} \times \\
&\qquad\qquad log\, \frac{p_\theta(\mathbf{X}|\mathbf{Z},\mathbf{A})\; p_\theta(\mathbf{A}|\boldsymbol{\omega})\; p_\theta(\mathbf{Z}|\mathbf{y})\; p_\theta(\boldsymbol{\omega}|\mathbf{y})\; p(\mathbf{y})}{\frac{q_\phi(\mathbf{y}|\mathbf{Z_c},\boldsymbol{\omega}_c)\; q_\phi(\boldsymbol{\omega}|\mathbf{A})\; q_\phi(\mathbf{Z}|\mathbf{X})}{q_\phi(\mathbf{y}|\mathbf{X},\mathbf{A})}} d\mathbf{Z}\, d\boldsymbol{\omega} \\
&= \mathbb{E}_{\mathbf{Z}\sim q_\phi(\mathbf{Z}|\mathbf{X})}\Bigg[\mathbb{E}_{\boldsymbol{\omega}\sim q_\phi(\boldsymbol{\omega}|\mathbf{A})}\Bigg[\frac{q_\phi(\mathbf{y}|\mathbf{Z_c},\boldsymbol{\omega}_c)}{q_\phi(\mathbf{y}|\mathbf{X},\mathbf{A})} \times \\
&\qquad\qquad log\, \frac{p_\theta(\mathbf{X}|\mathbf{Z},\mathbf{A})\; p_\theta(\mathbf{A}|\boldsymbol{\omega})\; p_\theta(\mathbf{Z}|\mathbf{y})\; p_\theta(\boldsymbol{\omega}|\mathbf{y})\; p(\mathbf{y})}{\frac{q_\phi(\mathbf{y}|\mathbf{Z_c},\boldsymbol{\omega}_c)\; q_\phi(\boldsymbol{\omega}|\mathbf{A})\; q_\phi(\mathbf{Z}|\mathbf{X})}{q_\phi(\mathbf{y}|\mathbf{X},\mathbf{A})}}\Bigg]\Bigg]
\end{aligned}$$

Since terms $log\, q_\phi(\mathbf{y}|\mathbf{X},\mathbf{A})$ and $log\, p(\mathbf{y})$ are independent from $\boldsymbol{\omega}$ and $\mathbf{Z}$, they can be moved out of the expectation operators. Re-writing logarithm of product as sum of logarithms yields the final objective as in 3.6:

$$
\begin{aligned}
\mathcal{L}(\mathbf{X},\mathbf{A},\mathbf{y}) = \mathbb{E}_{\mathbf{Z}\sim q_\phi(\mathbf{Z}|\mathbf{X})}\Big[\mathbb{E}_{\boldsymbol{\omega}\sim q_\phi(\boldsymbol{\omega}|\mathbf{A})}\Big[&\frac{q_\phi(\mathbf{y}|\mathbf{Z_c},\boldsymbol{\omega}_c)}{q_\phi(\mathbf{y}|\mathbf{X},\mathbf{A})} \\
&\times \Big\{log\, p_\theta(\mathbf{X}|\mathbf{Z},\mathbf{A}) - \big(log\, q_\phi(\mathbf{Z}|\mathbf{X}) - log\, p_\theta(\mathbf{Z_c}|\mathbf{y}) - log\, p(\mathbf{Z}_{\setminus\mathbf{c}})\big) \\
&+ log\, p_\theta(\mathbf{A}|\boldsymbol{\omega}) - \big(log\, q_\phi(\boldsymbol{\omega}|\mathbf{A}) - log\, p_\theta(\boldsymbol{\omega}_c|\mathbf{y}) - log\, p(\boldsymbol{\omega}_{\setminus c})\big) \\
&- log\, q_\phi(\mathbf{y}|\mathbf{Z_c},\boldsymbol{\omega}_c)\Big\}\Big]\Big] + log\, q_\phi(\mathbf{y}|\mathbf{X},\mathbf{A}) + log\, p(\mathbf{y})
\end{aligned}
$$

A.3. Model architecture details

Detailed information about the architectures of the model is given in Table A.1.

Encoder $q_\phi(\mathbf{Z}|\mathbf{X})$	Encoder $q_\phi(\boldsymbol{\omega}|\mathbf{A})$
Input 61 × 500 × 1 channel image	Input 61 × 61 × 1 channel image
Conv1D 8 × 7 (stride 3) & ReLU	Conv2D 32 × 3 × 3 (stride 2) & ReLU
Conv1D 16 × 7 (stride 3) & ReLU	Conv2D 64 × 5 × 5 (stride 2) & ReLU
Conv1D 16 × 7 (stride 3) & ReLU	Conv2D 32 × 5 × 5 (stride 2) & ReLU
Conv1D 16 × 5 (stride 3) & ReLU	Conv2D 16 × 5 × 5 (stride 2) & ReLU
Linear 96 × (2 × $|\mathbf{Z}|$)	Linear 256 × (2 × $|\mathbf{!}|$)

Decoder $p_\theta(\mathbf{X}|\mathbf{Z},\mathbf{A})$	Decoder $p_\theta(\mathbf{A}|\boldsymbol{\omega})$
Input 61 × 500 × 1 channel image	Input 61 × 61 × 1 channel image
GraphConv $|\mathbf{Z}|$ × 32 & ReLU	Linear $|\boldsymbol{\omega}|$ × 256 & ReLU
GraphConv 32 × 32 & ReLU	ConvT2D 32 × 5 × 5 (stride 2) & ReLU
GraphConv 32 × 32 & ReLU	ConvT2D 64 × 5 × 5 (stride 2) & ReLU
Conv1D 8 × 3 & Upsample(2) & ReLU	ConvT2D 32 × 5 × 5 (stride 2) & ReLU
Conv1D 16 × 5 & Upsample(2) & ReLU	ConvT2D 1 × 3 × 3 (stride 2)
Conv1D 16 × 5 & Upsample(2) & ReLU	
Conv1D 4 × 5 & Upsample(2) & ReLU	
Conv1D 1 × 5 & Upsample(2) & ReLU	

Classifier $q_\phi(\mathbf{y}|\mathbf{Z_c},\boldsymbol{\omega}_c)$	Conditional Prior $p_\theta(\mathbf{Z_c}|\mathbf{y})$	Conditional Prior $p_\theta(\boldsymbol{\omega}|\mathbf{y})$
Input 61 × $\mathbb{R}^3$, $\mathbb{R}^3$	Input $\{0,1\}^3$	Input $\{0,1\}^3$
Linear 61 × 1	Diagonal 3 × (61 × 3)	Diagonal 3 × 3
Diagonal 3 × 3		

Table A.1.: Architecture of the model.

B. Additional Results

B.1. Reconstruction ability

Here we report more examples demonstrating high reconstruction ability of the model in both EEG data (see Figure B.1) and functional connectivity domains (see Figure B.2).

B.2. Intervening on labels

We demonstrate the effect of intervening on labels *schizophrenia* and *hallucinations* on functional connectivity matrices in Figure B.3.

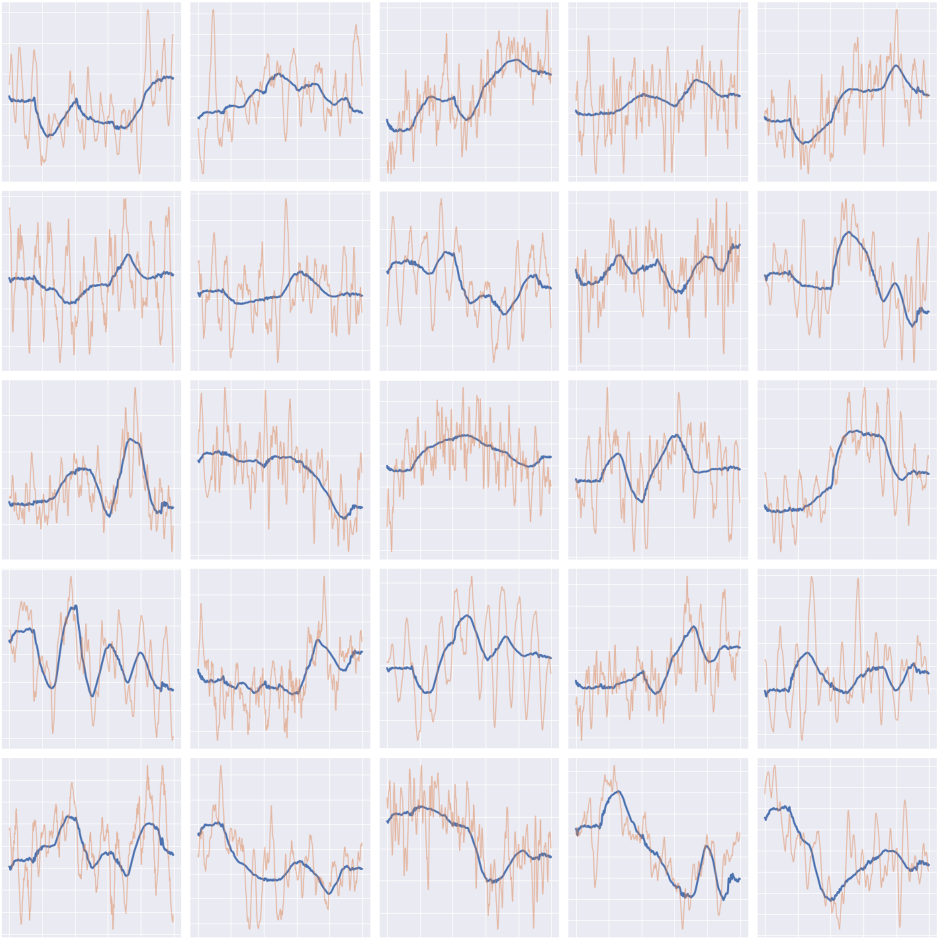


Figure B.1.: The ability of the model to reconstruct EEG recordings based on their latent representation **Z**. EEG recordings are depicted with orange while bold blue curves correspond to their reconstruction.

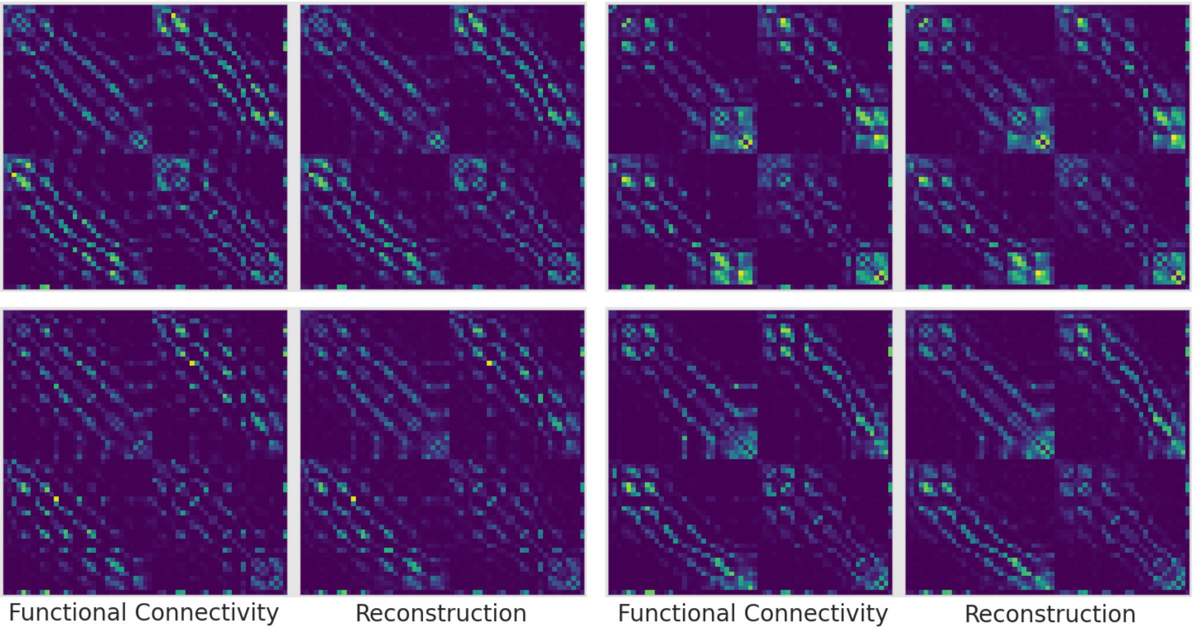


Figure B.2.: The ability of the model to reconstruct functional connectivity matrices based on their latent representation ω. Functional connectivity is reconstructed with Pearson correlation coefficient between corresponding node signals. We further highlight strong connections by applying a non-linear mapping 4.1.

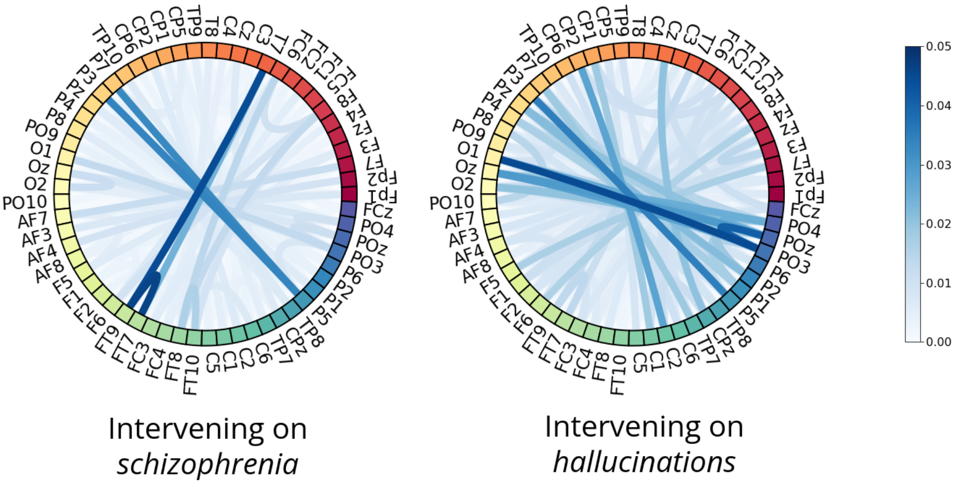


Figure B.3.: Variance in reconstructions of functional connectivity matrices when intervening on a single label: *schizophrenia* (Left) and *hallucinations* (Right).

List of Figures

2.1. Example of a graphical representation of a model for diagnosis of Parkinson's disease. 9
2.2. Learned two-dimension latent space of Frey Face from [9]. 17
2.3. Graphical model of CCVAE [10]. 21

3.1. Proposed graphical model. Latent variables are denoted with blue color and observed variables with white color. 28

4.1. The effect of the β-regularization of the CGVAE objective on the learned approximate posterior distribution $q_\phi(\boldsymbol{\omega}|\mathbf{A})$ (blue). Prior distribution over latent space $p_\theta(\boldsymbol{\omega}|\mathbf{y})$ is depicted with red color. Indicators on the left denote whether a dimension is assigned by a label (*schizophrenia, hallucinations*) or not (capital letters). 38
4.2. The ability of the model to reconstruct EEG recordings (Left) and functional connectivity matrices (Right) based on their latent representations $\mathbf{Z}$ and $\boldsymbol{\omega}$. Functional connectivity is reconstructed with Pearson correlation coefficient between corresponding node signals. We further highlight strong connections by applying a non-linear mapping 4.1. . 39
4.3. Most likely functional connectivity for healthy people and patients suffering from schizophrenia with or without AVH. Functional connectivity is reconstructed with Pearson correlation coefficient between corresponding node signals. We further highlight strong connections by applying a non-linear mapping 4.1. 40

B.1. The ability of the model to reconstruct EEG recordings based on their latent representation $\mathbf{Z}$. EEG recordings are depicted with orange while bold blue curves correspond to their reconstruction. 47
B.2. The ability of the model to reconstruct functional connectivity matrices based on their latent representation $\boldsymbol{\omega}$. Functional connectivity is reconstructed with Pearson correlation coefficient between corresponding node signals. We further highlight strong connections by applying a non-linear mapping 4.1. 48

B.3. Variance in reconstructions of functional connectivity matrices when intervening on a single label: *schizophrenia* (Left) and *hallucinations* (Right). 48

List of Tables

4.1. Reformulation of the disorder class (left column) to yield binary representation. 35

4.2. Comparison on the test partition of the dataset over different values of β for multiple metrics. The first two columns correspond to data pointwise mean square error between true data points and reconstructed ones. They are followed by KL divergence between the inference model and a prior distribution for both latent variables $\boldsymbol{\omega}$ and $\mathbf{Z}$. The last two columns demonstrate the accuracy of classification. 37

4.3. Variance in reconstructions of functional connectivity matrices when intervening on a single label: *schizophrenia* (Left) and *hallucinations* (Right). For each label, 5 connections are taken that have the highest variance of the correlation value. Each connection is provided with a pair of brain regions that it encompasses. Brain regions that are found to be involved in auditory tasks for healthy people are denoted with *. 41

A.1. Architecture of the model. 45

Bibliography

[1] A. Banino, C. Barry, B. Uria, C. Blundell, T. P. Lillicrap, P. W. Mirowski, A. Pritzel, M. J. Chadwick, T. Degris, J. Modayil, G. Wayne, H. Soyer, F. Viola, B. Zhang, R. Goroshin, N. C. Rabinowitz, R. Pascanu, C. Beattie, S. Petersen, A. Sadik, S. Gaffney, H. King, K. Kavukcuoglu, D. Hassabis, R. Hadsell, and D. Kumaran. "Vector-based navigation using grid-like representations in artificial agents". In: *Nature* 557 (2018), pp. 429–433.

[2] V. Morello, E. Barr, M. Bailes, C. Flynn, E. F. Keane, and W. van Straten. "SPINN: a straightforward machine learning solution to the pulsar candidate selection problem". In: *Monthly Notices of the Royal Astronomical Society* 443 (2014), pp. 1651–1662.

[3] J. M. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Zídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. A. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis. "Highly accurate protein structure prediction with AlphaFold". In: *Nature* 596 (2021), pp. 583–589.

[4] M. A. Mazurowski, M. Buda, A. Saha, and M. R. Bashir. "Deep learning in radiology: An overview of the concepts and a survey of the state of the art with focus on MRI". In: *Journal of Magnetic Resonance Imaging* 49 (2019).

[5] E. J. Topol. "Deep Medicine: How Artificial Intelligence Can Make Healthcare Human Again". In: 2019.

[6] B. Scholkopf, F. Locatello, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal, and Y. Bengio. "Toward Causal Representation Learning". In: *Proceedings of the IEEE* 109 (2021), pp. 612–634.

[7] J. M. Bishop. "Artificial Intelligence Is Stupid and Causal Reasoning Will Not Fix It". In: *Frontiers in Psychology* 11 (2020).

[8] A. J. DeGrave, J. D. Janizek, and S.-I. Lee. "AI for radiographic COVID-19 detection selects shortcuts over signal". In: *medRxiv* (2020).

[9] D. P. Kingma and M. Welling. "Auto-Encoding Variational Bayes". In: *CoRR* abs/1312.6114 (2014).

[10] T. Joy, S. M. Schmon, P. H. S. Torr, N. Siddharth, and T. Rainforth. “Capturing Label Characteristics in VAEs”. In: *ICLR*. 2021.

[11] G. C. O’Neill, P. Tewarie, D. Vidaurre, L. Liuzzi, M. W. Woolrich, and M. J. Brookes. “Dynamics of large-scale electrophysiological networks: A technical review”. In: *NeuroImage* 180 (2018), pp. 559–576.

[12] *Spurious Correlations*. `https://www.tylervigen.com/spurious-correlations`. Accessed: 2022-01-31.

[13] J. Pearl. “Causality: Models, Reasoning and Inference”. In: 2000.

[14] P. Rajpurkar, J. A. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Y. Ding, A. Bagul, C. Langlotz, K. S. Shpanskaya, M. P. Lungren, and A. Ng. “CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning”. In: *ArXiv* abs/1711.05225 (2017).

[15] P. S. de Laplace. “A Philosophical Essay On Probabilities”. In.

[16] S. Kullback and R. A. Leibler. “On Information and Sufficiency”. In: *Annals of Mathematical Statistics* 22 (1951), pp. 79–86.

[17] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. “Variational Inference: A Review for Statisticians”. In: *Journal of the American Statistical Association* 112 (2016), pp. 859–877.

[18] L. Maaløe, C. K. Sønderby, S. K. Sønderby, and O. Winther. “Auxiliary Deep Generative Models”. In: *ArXiv* abs/1602.05473 (2016).

[19] K. Ridgeway. “A Survey of Inductive Biases for Factorial Representation-Learning”. In: *ArXiv* abs/1612.05299 (2016).

[20] R. Gómez-Bombarelli, D. K. Duvenaud, J. M. Hernández-Lobato, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, and A. Aspuru-Guzik. “Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules”. In: *ACS Central Science* 4 (2018), pp. 268–276.

[21] K. Sohn, H. Lee, and X. Yan. “Learning Structured Output Representation using Deep Conditional Generative Models”. In: *NIPS*. 2015.

[22] J. Walker, C. Doersch, A. K. Gupta, and M. Hebert. “An Uncertain Future: Forecasting from Static Images Using Variational Autoencoders”. In: *ECCV*. 2016.

[23] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman. “Building machines that learn and think like people”. In: *Behavioral and Brain Sciences* 40 (2016).

[24] B. Schölkopf, D. Janzing, J. Peters, E. Sgouritsa, K. Zhang, and J. M. Mooij. “On causal and anticausal learning”. In: *ICML*. 2012.

[25] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling. “Semi-supervised Learning with Deep Generative Models”. In: *ArXiv* abs/1406.5298 (2014).

[26] X. Zhang, L. Yao, and F. Yuan. “Adversarial Variational Embedding for Robust Semi-supervised Learning”. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2019).

[27] M. Wegrzyn, S. J. Teipel, I. Oltmann, A. Bauer, J. Thome, A. Grossmann, K. Hauenstein, and J. Höppner. “Structural and functional cortical disconnection in Alzheimer’s disease: A combined study using diffusion tensor imaging and transcranial magnetic stimulation”. In: *Psychiatry Research: Neuroimaging* 212 (2013), pp. 192–200.

[28] N. Kumar, K. Alam, and A. H. Siddiqi. “Wavelet Transform for Classification of EEG Signal using SVM and ANN”. In: *Biomedical and Pharmacology Journal* 10 (2017), pp. 2061–2069.

[29] M. J. Hasan, D. Shon, K. Im, H.-K. Choi, D. Yoo, and J.-M. Kim. “Sleep State Classification Using Power Spectral Density and Residual Neural Network with Multichannel EEG Signals”. In: *Applied Sciences* 10 (2020), p. 7639.

[30] G. F. González, D. J. A. Smit, M. J. W. van der Molen, J. Tijms, C. J. Stam, E. J. C. de Geus, and M. W. van der Molen. “EEG Resting State Functional Connectivity in Adult Dyslexics Using Phase Lag Index and Graph Analysis”. In: *Frontiers in Human Neuroscience* 12 (2018).

[31] G. Li, Y. Jiang, W. Jiao, W. Xu, S. Huang, Z. Gao, J. Zhang, and C. Wang. “The Maximum Eigenvalue of the Brain Functional Network Adjacency Matrix: Meaning and Application in Mental Fatigue Evaluation”. In: *Brain Sciences* 10 (2020).

[32] S. Nobukawa, T. Yamanishi, S. Kasakawa, H. Nishimura, M. Kikuchi, and T. Takahashi. “Classification Methods Based on Complexity and Synchronization of Electroencephalography Signals in Alzheimer’s Disease”. In: *Frontiers in Psychiatry* 11 (2020).

[33] X. Bi and H. Wang. “Early Alzheimer’s disease diagnosis based on EEG spectral images using deep learning”. In: *Neural networks : the official journal of the International Neural Network Society* 114 (2019), pp. 119–135.

[34] A. Craik, Y. He, and J. L. Contreras-Vidal. “Deep learning for electroencephalogram (EEG) classification tasks: a review.” In: *Journal of neural engineering* 16 3 (2019), p. 031001.

[35] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, and M. Sun. “Graph Neural Networks: A Review of Methods and Applications”. In: *ArXiv* abs/1812.08434 (2020).

[36] S. Arslan, S. I. Ktena, B. Glocker, and D. Rueckert. “Graph Saliency Maps through Spectral Convolutional Networks: Application to Sex Classification with Brain Connectivity”. In: *GRAIL/Beyond-MIC@MICCAI*. 2018.

[37] B.-H. Kim and J. C. Ye. “Understanding Graph Isomorphism Network for rs-fMRI Functional Connectivity Analysis”. In: *Frontiers in Neuroscience* 14 (2020).

[38] A. Patil. “Applications of Electroencephalogram (EEG): Review”. In: *Political Economy - Development: Health eJournal* (2019).

[39] I. Higgins, L. Matthey, A. Pal, C. P. Burgess, X. Glorot, M. M. Botvinick, S. Mohamed, and A. Lerchner. “beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework”. In: *ICLR*. 2017.

[40] C. Morris, M. Ritzert, M. Fey, W. L. Hamilton, J. E. Lenssen, G. Rattan, and M. Grohe. “Weisfeiler and Leman Go Neural: Higher-order Graph Neural Networks”. In: *ArXiv* abs/1810.02244 (2019).

[41] S. Steinmann, G. Leicht, C. Andreou, N. Polomac, and C. Mulert. “Auditory verbal hallucinations related to altered long-range synchrony of gamma-band oscillations”. In: *Scientific Reports* 7 (2017).

Acknowledgments

First of all, I would like to thank Nico for the constant support and encouragement that he gave me throughout my time in Dresden. I am particularly grateful to him for the opportunity to work on the research problems that I was interested in. He has always been available to contact and our frequent discussions were valuable for the development of the thesis.

I also wish to thank Saskia who has guided me in the world of neuroimaging and was always open to giving me advice.

I am extremely thankful to my parents who made it possible for me to study in Germany. I couldn’t wish for better parents. I am grateful for all the emotional and financial support that they gave me on the way.

I would like to thank Clara for all the support that she gave me in the last two years. She has made the pandemic time much more pleasant and greatly helped me to overcome all the difficulties that I stumbled upon when moving to Germany.