Department of Physics and Astronomy
University of Heidelberg

Master Thesis in Physics
submitted by

Unai Fischer Abaigar

born in Berlin (Germany)

2022

Modeling Ordinal Mobile Data with sequential Variational Autoencoders

This Master Thesis has been carried out by
Unai Fischer Abaigar
at the Central Institute for Mental Health (ZI), Mannheim (Germany)
under the supervision of
Prof. Dr. Daniel Durstewitz

Abstract

Ordinal data, crucial for many scientific disciplines, consists of a discrete set of labels for which a natural ordering but no specified distance measure exists. In practice, this peculiarity of ordinal data is oftentimes overlooked, with many models making the simplified assumption that it can be interpreted as either metric or categorical. The rise of digital technologies allows the collection of ever larger data sets, facilitating the use of more powerful and expressive machine learning architectures. This thesis proposes and evaluates a deep probabilistic latent model for forecasting ordinal time series by integrating an ordered-logit model into a sequential variational autoencoder framework. The model is developed in the context of a multi-university research initiative (Living lab AI4U) with the goal to predict individuals' emotional trajectories using time series data collected from questionnaires on their smartphones to suggest personalized mental health interventions. The model is evaluated using empirical data collected during a psychiatric study and benchmark data matching this data is created to test the model's theoretical limitations. Hierarchical parameter estimation is implemented to deal with sparse and short time series. The findings identify future avenues for dealing with irregular time series, missing values and ways to integrate multimodal sensor data from smartphones.

Zusammenfassung

Ordinale Daten, welche für viele wissenschaftliche Disziplinen von entscheidender Bedeutung sind, bestehen aus einer diskreten Anzahl von Kategorien, für die eine natürliche Reihenfolge, aber kein spezifiziertes Abstandsmaß existiert. In der Praxis wird diese Besonderheit von ordinalen Daten oft übersehen, da viele Modelle vereinfachend davon ausgehen, dass sie entweder als metrisch oder nominal interpretiert werden können. Die zunehmende Verbreitung digitaler Technologien ermöglicht die Sammlung immer größerer Datensätze, was die Konstruktion von immer mächtigeren Machine Learning-Architekturen ermöglicht. In der vorliegenden Arbeit wird ein probabilistisches latentes Modell für die Vorhersage ordinaler Zeitreihen durch die Integration eines ordered-logit Modells in einen sequentiellen variational autoencoder entwickelt und evaluiert. Das Modell ist im Rahmen einer universitätsübergreifenden Forschungsinitiative (Living lab AI4U) entstanden, deren Ziel es ist, den Verlauf emotionaler Zustände von Individuen vorherzusagen, um personalisierte Interventionen zur Verbesserung der psychischen Gesundheit vorzuschlagen. Als Datenquelle dienen Zeitreihen, die über Smartphone-Fragebögen im Rahmen einer psychiatrischen Studie erhoben wurden. Mit diesen Daten werden übereinstimmende Benchmark-Daten erstellt, um die theoretischen Grenzen des Modells zu testen. Hierarchische Parameterschätzung wird implementiert, um mit kurzen Zeitreihen umzugehen. Die Ergebnisse eröffnen verschiedene Perspektiven für den zukünftigen Umgang mit unregelmäßigen Zeitreihen, fehlenden Werten und Möglichkeiten zur Integration multimodaler Sensordaten von Smartphones.

Contents

List of Figures

List of Tables

1 Introduction

It is difficult to make predictions, especially about the future.

— Danish proverb

Predictions of the future seem to have held a peculiar fascination for humanity since the dawn of history. From the carefully arranged fish guts beneath the knives of the augurs in ancient Rome to an octopus named Paul credited with correctly predicting the outcomes of all soccer games involving the German national team in the 2010 World Cup, humans have put their faith in an astounding variety of sources to find out what tomorrow holds in store.[1]

Being somewhat limited in scope, this thesis will solely concern itself with methods involving statistics. The focus lies on quantitative methods that rely on time series data collected in the past, without requiring specialized previous knowledge about the inner workings of the processes in question [Kantz and Schreiber 2004; Chatfield 2013; Lim and Zohren 2021]. This necessitates the assumption that the recorded data sufficiently captures the characteristic dynamics and the time scale of the system, and that we can expect these patterns to continue with some regularity in the future [Hyndman and Athanasopoulos 2018]. Although such forecasting methods are primarily informed by the data itself, it can still be very important to integrate knowledge from the respective application field, e.g. by choosing the most suited model class or formalizing insight from domain experts [Kantz and Schreiber 2004; Chatfield 2013].

In recent years, deep learning architectures, such as recurrent neural networks, have been employed with great success for a variety of forecasting problems [Lim and Zohren 2021], e.g. ranging from models for COVID-19 data [Ahmed et al. 2010], prediction of weather phenomena [Liu et al. 2015], power system planning [Guo et al. 2018] to financial time series [Sezer et al. 2020].

A common challenge encountered when constructing deep learning models for a diverse set of applications is how to handle the large variety of heterogeneous data modalities that get produced in different scientific contexts [Nazábal et al. 2020; Bommer et al. 2021; Shi et al. 2021]. Relatively little attention has so far been paid to the problem of forecasting ordinal time series by employing deep learning models, with most of the literature being concerned with classification and regression tasks

[1] For a history of human attempts to forecast the future, see the highly entertaining Minois 1998.

on non-sequential ordinal data [Cheng et al. 2008; Gutiérrez et al. 2016; Jaskari and Kivinen 2018; Lu et al. 2022], e.g. prediction of facial movements [Eleftheriadis et al. 2016] or imputation of missing values [Nazábal et al. 2020].

Ordinal data consists of a finite set of labels for which a natural ordering but no specified distance measure exists [Gutiérrez et al. 2016]. Thus, ordinal data can neither be assumed to exist on a metric interval scale nor is it truly nominal. Such data can be encountered in many different disciplines and application settings, ranging from assessing pain severity [Von Korff et al. 2000; Varin and Czado 2010], quality-of-life data in clinical trials [Goldberg et al. 2004; Lee and Daniels 2007], crash injury severity [Castro et al. 2013], corporate credit ratings [Hirk et al. 2018], ecological momentary assessments in psychiatry [Colombo et al. 2019], air quality [Kim 2017], anti-trafficking efforts [Wang et al. 2020] to brain computer interfacing [Yoon et al. 2011]. In general, ordinal variables tend to appear when collecting information on the preferences and opinions of people, for instance through questionnaires or other types of surveys, where they oftentimes take the form of so called Likert scales, e.g. responses ranging from "strongly disagree" to "strongly agree" [Likert 1932; Johnson and Albert 2006].

Longitudinal ordinal data has so far been primarily investigated outside of deep learning using a variety of modeling approaches, e.g. different Markov and generalized linear type models [Böckenholt 1999; Pruscha and Göttlein 2003; Lee and Daniels 2007; Lee and Daniels 2008], by leveraging the global odds ratio [Molenberghs and Lesaffre 1994; Williamson and Kim 1996], mixed autoregressive probit models [Varin and Czado 2010] or latent variable models [Todem et al. 2007; Cagnone et al. 2009; Tran et al. 2019].

Overall, many ordinal methods are based on the assumption that an underlying continuous latent variable exists that is segmented according to some threshold parameters into contiguous intervals that represent the different ordinal responses [Johnson and Albert 2006]. In practice, ordinal data is also oftentimes modeled by making the simplified assumption that it can be interpreted as either metric or categorical [Gutiérrez et al. 2016]. Such an approach has the convenient advantage that already existing powerful machine learning methods can simply be repurposed, but runs into the problem that converting ordinal labels to numerical values is theoretically unprincipled or that by alternatively treating ordinal prediction as a classification task the information about the ordering is lost [Gutiérrez et al. 2016; Lu et al. 2022].

In this thesis, I will build and evaluate a probabilistic latent variable model for ordinal time series data by leveraging a piecewise-linear RNN (PLRNN), a specific class of re-

current neural networks [Durstewitz 2017a], that will be trained through variational inference in the sequential variational autoencoder (SVAE) framework [Kingma and Welling 2014; Archer et al. 2015; Blei et al. 2017; Girin et al. 2021]. SVAEs extend the widely used Variational Autoencoder (VAE) [Kingma and Welling 2014] by including a state space model to account for the temporal dynamics. The SVAE framework has the great advantage that different observation modalities can be integrated in a straightforward way, which I will make use of by employing the proportional odds model (also called ordered logit model), one of the first threshold regression models specifically designed for ordinal data [McCullagh 1980].

As discussed, ordinal responses are oftentimes collected through questionnaires, which generally makes them difficult or costly to gather in large abundance. For instance, there are practical limitations to how often and how many participants are willing to answer a survey [Wen et al. 2017]. The rise of digital technologies somewhat alleviates this problem, as new ways to collect large data sources for social science research become more and more available, e.g. via smartphones, wearable technologies or social networks [Blazquez and Domenech 2018]. Still, ordinal data is generally much less readily available than data sets for other typical machine learning tasks, such as image classification [Dai et al. 2021]. Working with small datasets obviously presents a challenge from a modeling perspective, especially when wanting to apply expressive deep learning architectures that can require millions of data points and parameters to be successfully trained. Dealing with small and sparse data sets is a problem that also extends to many scientific disciplines and application contexts, ranging from material science [Zhang and Ling 2018] to psychiatry [Cearns et al. 2019; Koppe et al. 2021].

Here, as a first step towards solving this problem, I will implement and test a form of hierarchical parameter estimation. This approach is inspired by transfer learning [Pan and Yang 2010] and is an attempt to enable the model to leverage group-level information to strengthen predictions for short individual time series. Additionally, I will make use of the generative model capabilities to create tailored benchmark data to better test the theoretical limitations of the model for different sample sizes, and discuss and implement strategies to deal with irregular sampling, multimodal data and missing values.

1.1 Living Lab AI4U

The development of the model was carried out in the context of an interdisciplinary state-funded research initiative (Living lab AI4U), a multi-university collaboration exploring ways to use methods from artificial intelligence to promote mental health among young adults [Rauschenberg et al. 2021a]. Accordingly, the empirical investigation of this thesis is based upon data collected during a past psychiatric study [Rauschenberg et al. 2021b].

As discussed, ordinal data plays a large role in psychiatric research, as Likert scales are very commonly used on questionnaires collecting self-reported information about emotional states or behavior [Likert 1932]. There is some evidence to show that retrospective self-reporting can often lead to heavily biased results [Ben-Zeev et al. 2009], which is why ecological momentary assessments (EMAs) emerged as an alternative approach to evaluate the dynamics of mental health [Csikszentmihalyi and Larson 1987; Colombo et al. 2019]. The general idea behind this assessment method is to capture information on mood, behavior or thoughts of participants as close as possible to the moment they occur [Myin-Germeys et al. 2018]. Repeated and real-time measurements reduce recall bias and allow researchers to better understand the temporal and context-specific dependencies of emotional and behavioral symptoms [Ebner-Priemer and Trull 2009]. This is especially important, as it has been argued that investigating the temporal dynamics of psychiatric phenomena might be crucial for gaining a deeper understanding of mental illnesses [Bystritsky et al. 2012; Durstewitz et al. 2021].

Traditionally, this was oftentimes realized through written self-report diaries or questionnaires, in which participants could record their mood in a structured way several times a day over longer time periods [Verbrugge 1980; Stone et al. 2007]. Nowadays, the data collection process can be greatly facilitated through the use of digital technology in the context of mobile health (mHealth) [Myin-Germeys et al. 2016; Colombo et al. 2019]. For instance, the widespread usage of mobile devices now allows for EMAs to be administered through the use of smartphone applications [Seppälä et al. 2019], which makes is easier to store and collect the relevant information. Moreover, users can now be regularly reminded to respond to an assessment through mobile notifications and the process of filling out a questionnaire is usually quicker and more convenient than on paper [Colombo et al. 2019]. Mobile devices can also collect additional data without requiring direct user input, e.g. through the embedded sensors of a smartphone, ranging from step

counts, GPS tracking to general data on phone usage [Koppe et al. 2019b; Seppälä et al. 2019].
Ecological momentary interventions (EMIs) seek to apply similar principles to the treatment of psychiatric patients and general mental health promotion [Myin-Germeys et al. 2016]. In the same way that EMAs assume that emotions and behavior are best measured directly in the everyday context, EMIs seek to bring treatment options into the daily life of participants via the use of mobile devices [Colombo et al. 2019]. For instance, during the EMIcompass study adaptive interventions were administered through the smartphone of participants by making use of compassion-focused intervention techniques, such as positive imagery or self-compassionate writing [Rauschenberg et al. 2021b].
The rapid development of mobile healthcare services in psychiatry, but also in other medical contexts, e.g. the increasing number of electronic health records [Wang et al. 2018; Kim and Chung 2019], allows for the collection of larger data sets than what previously was possible [Donker et al. 2013; Durstewitz et al. 2019; Koppe et al. 2021]. This creates a growing potential for the deployment of powerful deep learning architectures to provide new types of treatment solutions and opens up new avenues for understanding mental health [Durstewitz et al. 2019; Koppe et al. 2019b]. On the other hand, while many types of mobile mental health applications do exists, much work still needs to be done to provide them with a solid scientific foundation [Donker et al. 2013]. Specifically in the context of time series forecasting, RNNs have already been employed for a variety of health applications, e.g. predicting depressed moods from self-reported histories and sleeping logs [Suhara et al. 2017], providing forecasts on users physical activity levels [Kim and Chung 2019], predicting self-reported emotional states [Sükei et al. 2021], forecasting stress levels [Mikelsons et al. 2017; Umematsu et al. 2019] or sleep quality prediction [Sathyanarayana et al. 2016].

The Living lab AI4U project seeks to further explore the potential for machine learning methods for general mental health promotion for adolescents and young adults [Rauschenberg et al. 2021a]. More specifically, the goal is to predict an individual's emotional trajectories using time series data collected from their smartphone via questionnaires and sensory data to suggest personalized digital mental health interventions. Therefore, the focus lies not only on forecasting the emotional dynamics of participants, but also on leveraging a machine learning

model to provide personalized treatment interventions.[2] This thesis is primarily focused on the first component; generating accurate forecasts of the ordinal EMA trajectories. This is arguably the key modeling challenges, as interventions can at least in principle be easily included into the modeling framework, as will be discussed later. Additionally, the models will be trained on ordinal trajectories alone, leaving the empirical investigation of sensor data for future work. To do so, the model will be evaluated on an empirical data set collected during the psychiatric EMIcompass study [Rauschenberg et al. 2021b]. Figure 1, presents a schematic overview for how RNNs could be used for mobile mental health treatment.

[2] Of course, the potential consequences, advantages and risks of using machine learning models for real-world (mental) health applications needs to be discussed in a much more in depth way than it is possible in the context of this thesis. Here, the primary focus lies on the methodological development, see the following review papers for a discussion on the future of machine learning approaches in mental health, and their potential societal and ethical ramifications [Dwyer et al. 2018; Thieme et al. 2020].

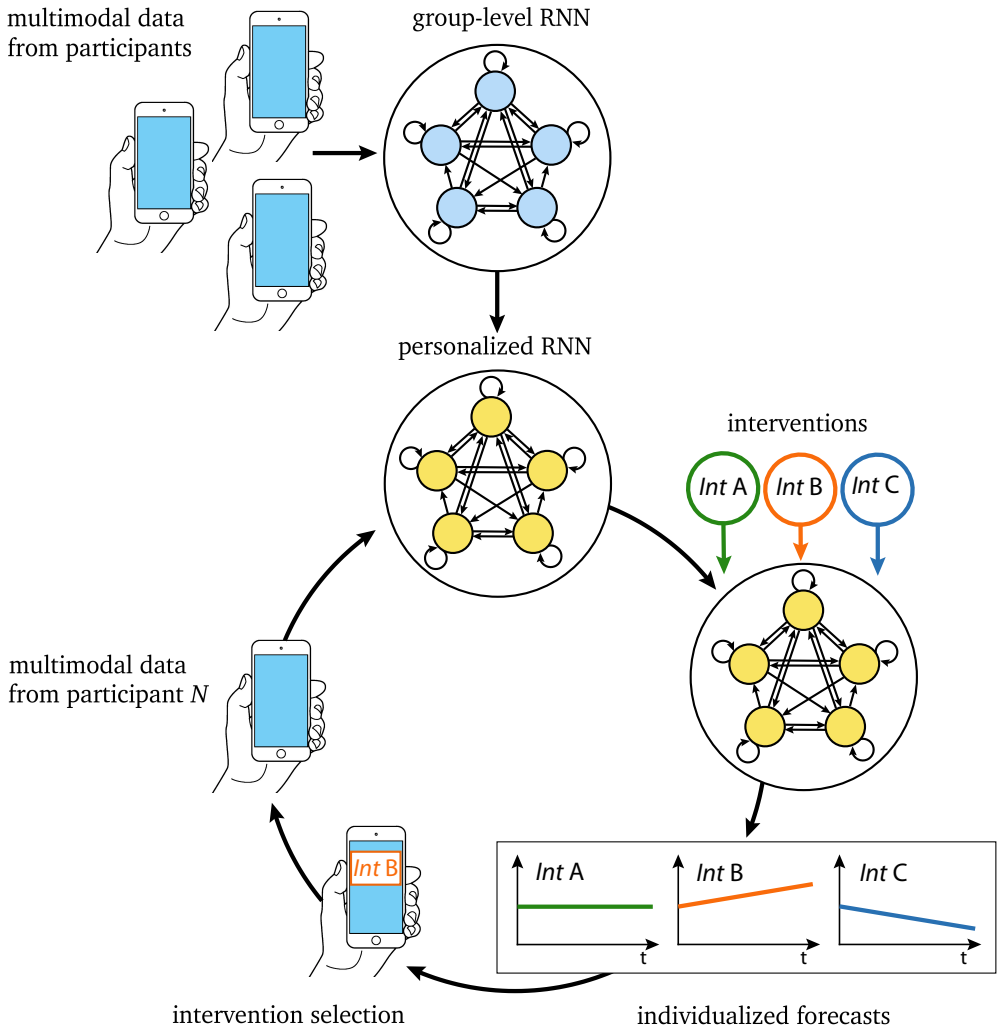


Figure 1: Multimodal data, collected through questionnaires and embedded smartphone sensors, can be used to train personalized RNN models. These subject-level models can be potentially improved by information collected from an entire group of participants, e.g. by pre-training the model or by employing a hierarchical parameter estimation. The trained model can then simulate the impact of several different mental health interventions on the future emotional trajectories of the participant. The model then selects the mobile intervention associated with the best future outcome, and sends it to the participant. The participant continues to log her emotional states, which allows the model to gain feedback on the recommended choice.

2 Theoretical Background

2.1 Fundamentals of Time Series Models

A time series is a sequence of observations made in time [Chatfield 2013]. Modeling and analyzing time series data presents a unique challenge from a statistical perspective [Durstewitz 2017b; Shumway and Stoffer 2017]. Many classical methods in statistics and machine learning depend on the common assumption that data points are identically and independently distributed. This is obviously not given for time series data, as observations made in close succession are usually highly dependent. This temporal dependency structure opens up the possibility of building models to forecast future values of a time series based on time points observed in the past [Chatfield 2013]. When attempting to find a plausible model for an observed time series, one usually needs to take into account some form of randomness [Chatfield 2013; Shumway and Stoffer 2017]. For instance, this might stem from not having observed all the key information required to describe and model the underlying process or simple measurement noise.
A time series can therefore be seen as the realization of a stochastic process, which in turn can be defined as a sequence of random variables $\{X_t\}$ indexed by time points t. An observed time series $\{x_t\}_{t=t_1}^{t_N}$ is one sample trajectory drawn from the corresponding random variables at each time point. Usually, a time series is assumed to be composed of discrete equidistant measurements $t_n = n\Delta t$, which is a consequence of limitations in the collection of data and computational analysis [Shumway and Stoffer 2017].

2.1.1 Moments of a Time Series Process

A stochastic process can be characterized by the cumulative joint distribution of all the random variables at all relevant time points [Chatfield 2013].

$$P(X_1 \leq x_1, \ldots, X_T \leq x_T) \tag{2.1}$$

In most cases, attaining this distribution is impossible or too impractical, which leads to using the moments of a stochastic process as a more straightforward informative description of the process [Shumway and Stoffer 2017].

- The **mean function** μ_t can be evaluated at all time points t.

$$\mu_t = \mathbb{E}[X_t] = \int_{-\infty}^{\infty} x p_t(x)\, dx \tag{2.2}$$

- The **autocovariance function** provides a measure of the linear dependencies between different time points [Durstewitz 2017b].

$$\text{acov}(X_t, X_{t+\Delta}) = \mathbb{E}[(X_t - \mu_t)(X_{t+\Delta} - \mu_{t+\Delta})] \tag{2.3}$$

  For a time difference of $\Delta = 0$ the autocovariance simply reduces to the variance. The corresponding autocorrelation is calculated by dividing by the respective standard deviations.

$$\text{acorr}(X_t, X_{t+\Delta}) = \frac{\text{acov}(X_t, X_{t+\Delta})}{\sqrt{\text{acov}(X_t, X_t)\text{acov}(X_{t+\Delta}, X_{t+\Delta})}} \tag{2.4}$$

  Usually, one expects the autocovariance to drop as the time difference between data points grows. As for the classical covariance, an autocovariance of zero only implies that time-points are not linearly related, but they still might very well exhibit some other form of nonlinear dependency.

2.1.2 Stationarity

All the mentioned properties refer to the infinite ensemble of potential time series realizations of the stochastic process. As discussed above, in almost all empirical settings, exceptions being very controlled lab settings, only a single realization can be accessed. This gives rise to the notion of **stationarity** [Durstewitz 2017b]. A stochastic process is called stationary (in the strict sense), when the overall cumulative distribution function is invariant in time.

$$P(X_1 \leq x_1, \ldots, X_T \leq x_T) = P(X_{1+\Delta} \leq x_{1+\Delta}, \ldots, X_{T+\Delta} \leq x_{T+\Delta}) \tag{2.5}$$

It follows that the marginal distributions of all the random variables X_t are the same, which for instance implies that mean μ_t and variance $\text{acov}(X_t, X_t)$ are also constant

in time [Shumway and Stoffer 2017]. Additionally, the dependency structure between random variables at different time points is invariant under time shift, and therefore only depends on the time lag between them. For example, under stationarity assumption the autocorrelation is independent of the specific time points and only depends on the time difference.

$$acorr(X_t, X_{t+\Delta}) = acorr(X_{t+h}, X_{t+\Delta+h}) \tag{2.6}$$

Again, it is important to realize that the probability distributions and expectations refer to the entire ensemble of the stochastic process [Durstewitz 2017b]. In one sample trajectory there still might very well be variation across time, induced through an underlying periodic process, without breaking any of the conditions of stationarity.
Oftentimes, a weaker form of stationarity is defined that only requires a constant mean and autocovariance. This condition is easier to determine, and forms the basis of much of linear time series analysis, where dependencies between time points are fully captured by the autocorrelation [Shumway and Stoffer 2017], but is clearly insufficient when considering more complex non-linear dynamics [Kantz and Schreiber 2004].
From a dynamical systems perspective, one might also call a time series stationary if it was produced by a dynamical system whose parameters θ are constant over the entire observed time period. [Kantz and Schreiber 2004; Durstewitz 2017b].

$$x_t = f_\theta(x_{t-1}) + \epsilon_t \tag{2.7}$$

In any case, one rarely has knowledge about the true underlying dynamics that generated a time series, which makes it quite difficult to determine if a system is stationary from a single time series. The recorded time period might just not be long enough to understand the general behavior; e.g. an oscillation might look like a linear trend on a short time scale. Practically speaking, one needs to ensure that the observed time series contains enough information, and that the dynamics show some kind of regularity to effectively build a model of the underlying process. [Kantz and Schreiber 2004; Durstewitz 2017b].

2.1.3 White Noise Process

The most basic building block for a time series model is the so called white noise process. It consists of a sequence of uncorrelated random variables $\epsilon_t \sim W(0, \sigma^2)$ with a finite and fixed variance σ^2 and a mean of zero [Chatfield 2013; Durstewitz 2017b; Shumway and Stoffer 2017].

$$acov(\epsilon_t, \epsilon_{t+\Delta}) = \begin{cases} \sigma^2, & \text{if } \Delta = 0. \\ 0, & \text{otherwise.} \end{cases} \tag{2.8}$$

$$\mathbb{E}[\epsilon_t] = 0, \text{ for all } t \tag{2.9}$$

Oftentimes, the additional assumption is made that the variables are Gaussian distributed $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$.

2.1.4 Linear Time Series Models

The arguably most popular type of stochastic time series model is the linear autoregressive process that assumes that current values of a time series result from a linear function of past observations combined with a white noise process $\epsilon_t \sim W(0, \sigma^2)$ and a constant parameter c [Chatfield 2013; Durstewitz 2017b].

$$X_t = c + \sum_{i=1}^{p} a_i X_{t-i} + \epsilon_t \tag{2.10}$$

The order parameter p of an **AR**(p) process corresponds to the length of the history of observations used to determine the next value. A second description is the so called moving average process **MA**(q), where the current output depends on a linear combination of past noise terms ϵ_{t-j}.

$$X_t = c + \sum_{j=1}^{q} b_j \epsilon_{t-j} + \epsilon_t \tag{2.11}$$

It is oftentimes useful to combine the **AR**(p) process and **MA**(q) process into a single model, the widely used Autoregressive Moving Average **ARMA**(p, q) process.

$$X_t = c + \sum_{i=1}^{p} a_i X_{t-i} + \sum_{j=1}^{q} b_j \epsilon_{t-j} + \epsilon_t \tag{2.12}$$

A **MA**(q) process can always be expressed as an infinite order **AR**(p) process and vice versa [Durstewitz 2017b] . A combined representation still has benefits, for instance expressing a process with less parameters than a pure **AR**(p) or **MA**(q) process would have needed [Chatfield 2013].

The parameters a_i of an **AR**(p) process can be estimated in a fairly straightforward way, for instance by ordinary least squares or using the Yule-Walker equations. Determining the moving average coefficients b_j is more involved and usually requires computational optimization methods [Chatfield 2013; Durstewitz 2017b]. The order q of a **MA**(q) process can be obtained by calculating the sample autocorrelation function. The autocorrelation is zero for time lags larger than q, so one can check at which time lags the empirical estimate becomes close to zero [Chatfield 2013; Durstewitz 2017b]. A first estimate for the order p of an **AR**(p) process can be determined through the so called partial autocorrelation function, which measures the correlation between time points separated by a certain lag k, while desregarding the correlation due to the dependency of the time points in between. Similar to the autocorrelation for the **MA**(q) process, the partial autocorrelation drops to zero for a time lag larger than p. In practice, it is oftentimes recommended to use other criteria to identify the correct order, such as the Akaike's Information Criterion (AIC) or the Bayesian Information Criterion (BIC) [Chatfield 2013].

After estimating all the necessary parameters, future values can be predicted by iterating the process forward [Durstewitz 2017b; Shumway and Stoffer 2017]. For instance, for an **AR**(p) process the best one step ahead prediction (minimizing the mean square error) is

$$\hat{x}_{T+1} = \mathbb{E}[x_{T+1}|x_1, \ldots, x_T] = c + \sum_{i=1}^{p} a_i x_{T+1-i} \tag{2.13}$$

There exist a variety of model extensions and ways to generalize the **ARMA**(p, q) model. For instance, the autoregressive integrated moving average process (**ARIMA**) can be used to fit non-stationary time series by calculating time differences between consecutive observations [Chatfield 2013]. The model can also be easily expanded

for multivariate time series, leading to the so called vector autoregressive process [Durstewitz 2017b].

2.2 State Space Models

When constructing a time series models for complex phenomena, one can very rarely assume that the observations made sufficiently capture and describe the dynamics of the system [Durstewitz 2017b]. This might be due to not having measured all the important system variables, or just not being able to directly observe them. These unobserved, or underlying variables describe the actual process of interest from which the observed time series $\mathbf{x}_t$ is generated. The underlying variables $\mathbf{z}_t$ are also oftentimes called latent variables. From a scientific perspective, it is of great interest to be able to access and infer the underlying dynamics to further our understanding of the system in question [Durstewitz 2017b]. For instance, in a psychiatric context, data produced through ecological momentary assessments might allow insights into the actual behavioral and cognitive mechanisms of participants [Koppe et al. 2019b; Durstewitz et al. 2021].

In state space models, the latent process is assumed to be a first-order Markov chain, meaning that the next latent state $\mathbf{z}_{t+1}$ is only dependent on the previous time step $\mathbf{z}_t$ [Durbin and Koopman 2012; Durstewitz 2017b; Shumway and Stoffer 2017]. In different terms, the future state conditioned on the present latent state is independent of past states. Thus, the transition probabilities $p(\mathbf{z}_{t+1}|\mathbf{z}_t)$ fully determine the evolution of the underlying stochastic process.

$$p(\mathbf{z}_{t+1}|\mathbf{z}_1, \ldots, \mathbf{z}_t) = p(\mathbf{z}_{t+1}|\mathbf{z}_t) \tag{2.14}$$

It is further assumed that the measured values $\mathbf{x}_t$ solely depend on the latent state $\mathbf{z}_t$ at the respective time step t. This mapping is described by the observation process $p(\mathbf{x}_t|\mathbf{z}_t)$ [Durstewitz 2017b].

$$p(\mathbf{x}_t|\mathbf{z}_1, \ldots, \mathbf{z}_t) = p(\mathbf{x}_t|\mathbf{z}_t) \tag{2.15}$$

Furthermore, observations given their corresponding latent states are independent of each other, which implies that the temporal dependency structure is fully encoded in the latent process [Durstewitz 2017b].

$$p(\mathbf{x}_t, \mathbf{x}_{t'}|\mathbf{z}_t, \mathbf{z}_{t'}) = p(\mathbf{x}_t|\mathbf{z}_t, \mathbf{z}_{t'})p(\mathbf{x}_{t'}|\mathbf{z}_t, \mathbf{z}_{t'}) = p(\mathbf{x}_t|\mathbf{z}_t)p(\mathbf{x}_{t'}|\mathbf{z}_{t'}) \tag{2.16}$$

The observations themselves can obviously still exhibit any kind of dependencies in time, only when conditioned on the markovian latent process the independence assumption holds [Durstewitz 2017b].
The latent process 2.14 and observation equation 2.15 form the so called generative model, which can be succinctly written as the joint probability distribution of observations $\{\mathbf{x}_t\}$ and latent states $\{\mathbf{z}_t\}$. Using the mentioned properties of state space models (2.14, 2.15, 2.16) the joint probability distribution factorizes into the transition probabilities of the latent states, the observation model and the prior distribution of $\mathbf{z}_1$.

$$p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}_1, \ldots, \mathbf{x}_T, \mathbf{z}_1, \ldots, \mathbf{z}_T) = p(\mathbf{z}_1)\prod_{t=2}^{T} p(\mathbf{z}_t|\mathbf{z}_{t-1})\prod_{t=1}^{T} p(\mathbf{x}_t|\mathbf{z}_t) \tag{2.17}$$

Oftentimes, it is convenient to use the logarithm of the joint distribution, because it allows for the product terms to be reformulated as sums.

$$\log p(\mathbf{x}, \mathbf{z}) = \log p(\mathbf{x}_1, \ldots, \mathbf{x}_T, \mathbf{z}_1, \ldots, \mathbf{z}_T) \tag{2.18}$$

$$= \log p(\mathbf{z}_1) + \sum_{t=2}^{T} \log p(\mathbf{z}_t|\mathbf{z}_{t-1}) + \sum_{t=1}^{T} \log p(\mathbf{x}_t|\mathbf{z}_t) \tag{2.19}$$

2.2.1 Linear State Space Models

The linear time series models presented in Section 2.1.4 can be used to formulate a linear state space model [Durbin and Koopman 2012; Durstewitz 2017b]. Under the assumption of Gaussian process $\boldsymbol{\epsilon}_t$ and observation noise $\boldsymbol{\eta}_t$, the multivariate latent process and observation equation can be expressed as

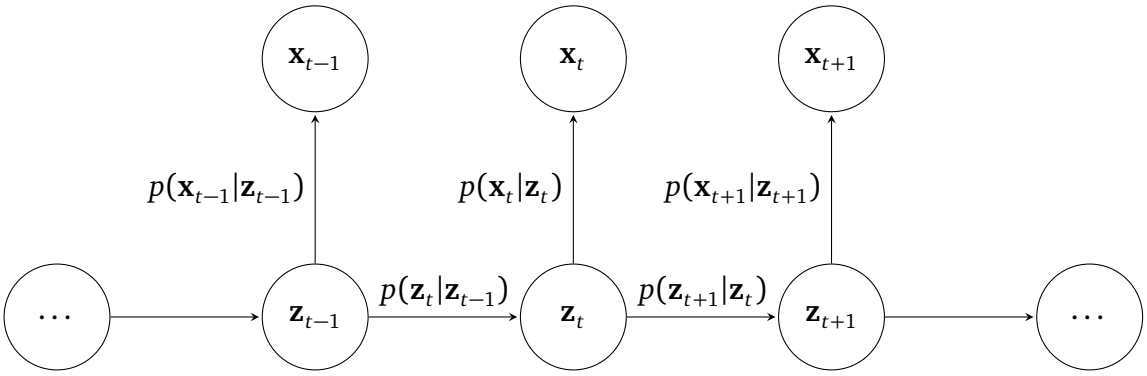


Figure 2: A state space model is compromised of a latent first-order Markov process, and an observation equation connecting the latent states at each time point t to the corresponding conditionally independent observation.

$$\mathbf{z}_t = \mathbf{A}\mathbf{z}_{t-1} + \boldsymbol{\epsilon}_t,\ \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}) \tag{2.20}$$

$$\mathbf{x}_t = \mathbf{B}\mathbf{z}_t + \boldsymbol{\eta}_t,\ \boldsymbol{\eta}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma}), \tag{2.21}$$

where the latent vectors $\mathbf{z}_t = (z_{1t} \dots z_{Mt})^T$ and observations $\mathbf{x}_t = (x_{1t} \dots x_{Nt})^T$ are of dimension M and N, and $\mathbf{B}$ denotes the $(N \times M)$ dimensional observation matrix. The initial state $\mathbf{z}_1$ is drawn from its own distribution $\mathbf{z}_1 \sim \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$. The state process equation corresponds to a vector autoregressive process of order one, while the observation equation is equivalent to a linear regression problem.
The Gaussian observation noise $\boldsymbol{\eta}_t$ represents the inherent randomness when taking measurements, e.g. precision errors of instruments [Durstewitz 2017b]. The process noise $\boldsymbol{\epsilon}_t$ is conceptually more difficult to grasp. Putting the question aside, if there are truly stochastic processes in the real-world, for instance close to the quantum level, in most cases we have to assume that the latent model can only capture an approximation of the real-world dynamics, which can be expressed using the process noise [Durstewitz 2017b].

2.2.2 Piecewise-Linear Recurrent Neural Networks

Linear time series models can be a very useful tool and have many benefits, such as fairly straightforward ways to estimate parameters, being more interpretable, or requiring little data and less computational resources to train, at least in comparison to deep learning methods [Durstewitz 2017b; Koppe et al. 2021]. On the other

hand, linear systems are restricted in the variety of dynamical system phenomena they can represent [Kantz and Schreiber 2004; Durstewitz 2017b]. In many empirical settings, one might expect the underlying dynamics to show complex and nonlinear dependencies that can not be fully explained through linear correlations. This creates the need for more expressive models that can deal with highly irregular, non-linear and potentially chaotic time series data.
Recurrent neural networks (RNNs) possess, at least in theory, the ability to approximate any kind of dynamical system [Funahashi and Nakamura 1993; Durstewitz 2017a], which should make them in principle powerful enough to even represent complex cognitive or behavioral dynamics. Piecewise-linear (PL) recurrent neural networks are a specific class of recurrent neural networks [Durstewitz 2017a; Koppe et al. 2019a] that can be used to formulate a nonlinear state space model. A big advantage of PLRNNs is that they allow for the analytical calculation of various dynamical system properties, such as fixed points or cycles [Durstewitz 2017a; Koppe et al. 2019a; Monfared and Durstewitz 2020a; Schmidt et al. 2021]. Using an analytically tractable system allows us to regain some of the interpretative power that is lost when moving form linear to nonlinear systems. This is of special interest in scientific settings, where one seeks to understand the underlying mechanisms. This is outside of the scope of this thesis, but I would still argue that it is preferable to use a model that is potentially interpretable in the future. In psychiatric or general health contexts it might be of large importance to make predictions and models explainable, e.g. give some insight why a specific intervention was recommended to a patient. Additionally, when working in low data limits, as in psychiatry or many scientific prediction settings, one needs to gain and leverage as much knowledge as possible about the data generating process to tailor the modeling approach to the empirical problem at hand.

The latent process equation of a PLRNN is given by [Durstewitz 2017a; Koppe et al. 2019a]:

$$\mathbf{z}_t = \mathbf{A}\mathbf{z}_{t-1} + \mathbf{W}\boldsymbol{\phi}(\mathbf{z}_{t-1}) + \mathbf{h} + \boldsymbol{\epsilon}_t, \quad \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}), \tag{2.22}$$

where $\boldsymbol{\phi}$ is a rectified linear unit activation function that is applied element wise to the latent states $\phi(\mathbf{z}_t)_i = \max(0, z_{it})$, see Figure 3. The auto-regressive connections between states correspond to the $(M \times M)$-dimensional diagonal matrix

A, while the connection weights between units are determined by the off-diagonal $(M \times M)$-dimensional **W** matrix. $\boldsymbol{\epsilon}_t$ is a Gaussian white noise process with a diagonal covariance matrix $\boldsymbol{\Sigma}$. **h** is a constant vector of bias parameters.

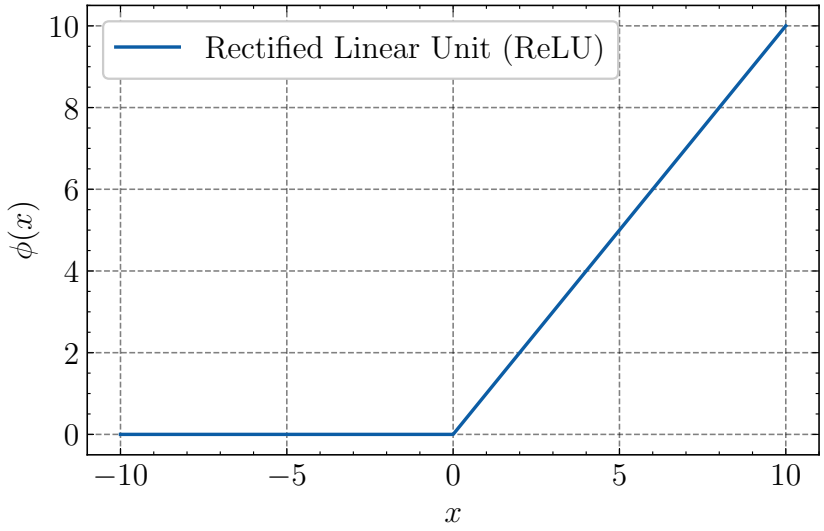


Figure 3: The Rectified Linear Unit (ReLU) activation function $\phi(x) = \max(0, x)$.

As in the linear case, the initial latent state $\mathbf{z}_1$ is sampled from a Gaussian distribution with mean $\boldsymbol{\mu}_0$ and covariance matrix $\boldsymbol{\Sigma}_0$.

$$\mathbf{z}_1 \sim \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0) \tag{2.23}$$

The process equation can also be represented using the Gaussian transition probabilities $p(\mathbf{z}_t|\mathbf{z}_{t-1})$.

$$p(\mathbf{z}_t|\mathbf{z}_{t-1}) = \mathcal{N}(\mathbf{A}\mathbf{z}_{t-1} + \mathbf{W}\boldsymbol{\phi}(\mathbf{z}_{t-1}) + \mathbf{h}, \boldsymbol{\Sigma}) = \mathcal{N}(\boldsymbol{\mu}_t(\mathbf{z}_{t-1}), \boldsymbol{\Sigma}) \tag{2.24}$$

2.2.2.1 Basis Expansion

In statistics, a basis or spline expansion is a popular tool to increase the capacity of a linear model [Hastie et al. 2009; Durstewitz 2017b]. The vector of regressors **x** is simply replaced by a set of nonlinear functions g_m, which conveniently retains the linearity in the parameters β_m once the functions are known.

$$f(\mathbf{x}) = \sum_{m=1}^{M} \beta_m g_m(\mathbf{x}) \tag{2.25}$$

A similar linear spline expansion can also be used to enhance the expressiveness of a PLRNN by replacing the ReLU-term $\boldsymbol{\phi}(\mathbf{z}_{t-1})$ with a linear combination of ReLU functions with different thresholds $\mathbf{h}_b \in \mathbb{R}^M$ and basis coefficients $\alpha_b \in \mathbb{R}$ [Brenner et al. 2021].

$$\phi(\mathbf{z}_{t-1}) = \sum_{b=1}^{B} \alpha_b \max(0, \mathbf{z}_{t-1} - \mathbf{h}_b) \tag{2.26}$$

B is called the number of bases. This leads to the expanded latent step equation:

$$\mathbf{z}_t = \mathbf{A}\mathbf{z}_{t-1} + \mathbf{W}\sum_{b=1}^{B} \alpha_b \max(0, \mathbf{z}_{t-1} - \mathbf{h}_b) + \mathbf{h} + \boldsymbol{\epsilon}_t, \quad \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}). \tag{2.27}$$

While the original formulation of the PLRNN should be universal enough to approximate any dynamics, the basis expansion is still useful as it leads to a more parsimonious representation of the model [Brenner et al. 2021]. This allows the model to represent the same dynamics with a smaller latent dimension M, and overall less dynamical parameters.

2.2.2.2 Clipped PLRNN

To ensure that the latent states do not diverge and stay bounded, a simple "clipped" version of the PLRNN can be formulated, as presented in Brenner et al. 2021.

$$\mathbf{z}_t = \mathbf{A}\mathbf{z}_{t-1} + \mathbf{W}\sum_{b=1}^{B} \alpha_b [\max(0, \mathbf{z}_{t-1} - \mathbf{h}_b) - \max(0, \mathbf{z}_{t-1})] + \mathbf{h} + \boldsymbol{\epsilon}_t \tag{2.28}$$

To simplify notation, the additional ReLU term will be omitted in the following Sections.

2.2.3 Maximum Likelihood Estimation

The next step is to infer all the parameters $\boldsymbol{\theta} = \{\mathbf{A}, \mathbf{W}, \{\alpha_b, \mathbf{h}_b\}, \mathbf{h}, \boldsymbol{\Sigma}, \mathbf{B}, \boldsymbol{\Gamma}\}$ of the generative model $p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})$ that consists of a PLRNN and a simple Gaussian observation model, from an observed time series $\{\mathbf{x}_t\}$.

$$\mathbf{z}_t = \mathbf{A}\mathbf{z}_{t-1} + \mathbf{W}\sum_{b=1}^{B}\alpha_b \max(0, \mathbf{z}_{t-1} - \mathbf{h}_b) + \mathbf{h} + \boldsymbol{\epsilon}_t, \quad \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}) \tag{2.29}$$

$$\mathbf{x}_t = \mathbf{B}\mathbf{z}_t + \boldsymbol{\eta}_t, \ \boldsymbol{\eta}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma}) \tag{2.30}$$

A common way to tackle this problem is to maximize the log-likelihood $\log p_{\boldsymbol{\theta}}(\mathbf{x})$ with respect to the unknown parameters $\boldsymbol{\theta}$ [Kingma and Welling 2014; Durstewitz 2017b; Kingma and Welling 2019].

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\mathbf{x}) \tag{2.31}$$

For a latent variable model, the marginal likelihood $p_{\boldsymbol{\theta}}(\mathbf{x})$ is calculated by integrating the generative joint likelihood $p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})$ across all possible latent trajectories [Durstewitz 2017b; Kingma and Welling 2019].

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) = \log \int p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})\, d\mathbf{z} = \log \int p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) p_{\boldsymbol{\theta}}(\mathbf{z})\, d\mathbf{z} \tag{2.32}$$

This integral is impossible or too impractical to analytically compute, which makes it difficult to directly optimize the log-likelihood, and in turn creates the need for specialized learning algorithms [Tzikas et al. 2008; Durstewitz 2017b]. This problem is strongly related to the task of inferring the posterior distribution of the latent states $p(\mathbf{z}|\mathbf{x})$ [Kingma and Welling 2019].

$$p(\mathbf{z}|\mathbf{x}) = \frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})}{p_{\boldsymbol{\theta}}(\mathbf{x})} = \frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})}{\int p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) d\mathbf{z}} \tag{2.33}$$

To summarize, the goal is to find point estimates of the parameters $\boldsymbol{\theta}$ from the observed time series $\mathbf{x}$, while still treating the underlying latent states $\mathbf{z}$ as random

variables. A step further would lead to a fully Bayesian framework, where also the parameters $\boldsymbol{\theta}$ are assumed to be random variables [Blei et al. 2017; Sayer 2020].

2.2.4 Evidence Lower Bound

Since the log-likelihood described in the previous section is not easily computable, a different optimization criterium needs to be found. To do so, a proposal density of the latent states $q(\mathbf{z}|\mathbf{x})$ can be introduced, which allows the derivation of a lower bound of the log-likelihood [Jordan et al. 1998; Bishop 2006]. The derivation holds for an arbitrary probability density $q(\mathbf{z})$, although its meaning will be made clear soon.

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) = \log \int p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) \frac{q(\mathbf{z}|\mathbf{x})}{q(\mathbf{z}|\mathbf{x})} d\mathbf{z} = \log \mathop{\mathbb{E}}_{q(\mathbf{z}|\mathbf{x})} \left[\frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})}{q(\mathbf{z}|\mathbf{x})} \right] \tag{2.34}$$

The lower bound can be found by using Jensen's inequality that holds if f is a concave function.

$$f(\mathbb{E}[x]) \geq \mathbb{E}[f(x)] \tag{2.35}$$

Applying Jensens's inequality to the logarithm in equation 2.34, we arrive at:

$$\begin{aligned} \log p_{\boldsymbol{\theta}}(\mathbf{x}) &= \log \mathop{\mathbb{E}}_{q(\mathbf{z}|\mathbf{x})} \left[\frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})}{q(\mathbf{z}|\mathbf{x})} \right] \\ &\geq \mathop{\mathbb{E}}_{q(\mathbf{z}|\mathbf{x})} \left[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})}{q(\mathbf{z}|\mathbf{x})} \right] \\ &= \mathop{\mathbb{E}}_{q(\mathbf{z}|\mathbf{x})} [\log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})] - \mathop{\mathbb{E}}_{q(\mathbf{z}|\mathbf{x})} [\log q(\mathbf{z}|\mathbf{x})] \\ &= \mathrm{ELBO}_{\mathbf{x}}(\boldsymbol{\theta}) \end{aligned} \tag{2.36}$$

This approximation of the log-likelihood is called the evidence lower bound (ELBO). The ELBO can be reformulated in terms of the Kullback-Leibler divergence, which makes the role of the proposal density q more obvious [Bishop 2006; Blei et al. 2017].

$$\begin{aligned}\log p_{\boldsymbol{\theta}}(\mathbf{x}) &\geq -\mathop{\mathbb{E}}_{q(\mathbf{z}|\mathbf{x})}[\log q(\mathbf{z}|\mathbf{x})] + \mathop{\mathbb{E}}_{q(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})] + \mathop{\mathbb{E}}_{q(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x})] \\ &= -KL(q(\mathbf{z}|\mathbf{x}) \parallel p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})) + \log p_{\boldsymbol{\theta}}(\mathbf{x}) \\ &= \mathrm{ELBO}_{\mathbf{x}}(\boldsymbol{\theta}) \end{aligned} \tag{2.37}$$

The Kullback-Leibler divergence becomes zero if the proposal density is equal to the posterior density of the latent states $q(\mathbf{z}|\mathbf{x}) = p_\theta(\mathbf{z}|\mathbf{x})$. Therefore, the better the agreement between q and the true posterior $p_\theta(\mathbf{z}|\mathbf{x})$, the tighter the bound; if they are equal the ELBO corresponds to the log-likelihood.

A very prominent way to then calculate the maximum likelihood estimate for a latent variable model is the Expectation-Maximization (EM) algorithm [Bishop 2006; Durstewitz 2017b]. It maximizes the log-likelihood iteratively, where every iteration consists of a so called E and M-step. During the E-step the EM-algorithm calculates (or approximates) the posterior probability distribution $p_\theta(\mathbf{z}|\mathbf{x})$, which is equivalent to maximizing the ELBO with regards to $q(\mathbf{z}|\mathbf{x})$ for a fixed global parameter set $\boldsymbol{\theta}$. In the M-step, the previously calculated density $q(\mathbf{z}|\mathbf{x}) \approx p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$ is fixed, which causes the KL-divergence to vanish. The ELBO is then maximized with regards to the generative model parameters $\boldsymbol{\theta}$, leading to the log-likelihood rising at least as much as the lower bound [Bishop 2006]. This is repeated until convergence. Therefore, the EM algorithm can be used to maximize the likelihood through two more easily to compute steps. For instance, for linear state space models that were discussed in Section 2.2.1, the E-step can be calculated analytically through the Kalman-filter recursions [Kalman 1960; Durstewitz 2017b].

It has been demonstrated that the EM-algorithm can be used to efficiently train a PLRNN with a Gaussian observation model by combining analytical calculations with a Newton-type iteration scheme [Durstewitz 2017a; Koppe et al. 2019a]. This approach can also be extended to non-Gaussian observation models, although more complicated observation models, such as non exponential family distributions, might become too difficult to handle [Bommer et al. 2021]. Additionally, it is oftentimes computationally prohibitive to train larger datasets with the EM-algorithm [Kingma and Welling 2019].

Alternatively, latent variable models can be trained by leveraging principles from variational inference, more specifically in the framework of variational autoencoders [Kingma and Welling 2014]. Variational autoencoders have the advantage that they

allow for greater flexibility for incorporating different generative models and data modalities. Additionally, they can be trained very efficiently for larger datasets and high-dimensional latent spaces [Kingma and Welling 2014; Kingma and Welling 2019].

2.3 Variational Autoencoders

2.3.1 Variational Inference

A general approach to approximate a posterior distribution $p(\mathbf{z}|\mathbf{x})$ for which no closed form solution exists, or only one that would be too computationally expensive to calculate, can be found in variational methods [Blei et al. 2017; Kingma and Welling 2019]. The central idea is to introduce an approximate posterior $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ that is part of a family of tractable distributions $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \in \mathcal{Q}$, which is determined by the variational parameters $\boldsymbol{\phi}$. The goal is then to choose the optimal parameters $\boldsymbol{\phi}$ so that the distribution $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ is as close as possible to the true posterior $p(\mathbf{z}|\mathbf{x})$. The family of distributions $\mathcal{Q}$ needs to be as flexible and expressive as possible without making the optimization problem too difficult [Bishop 2006]. The Kullback-Leibler divergence can be used to capture the similarity between the variational distribution $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ and the true conditional. Thus, we can reformulate the previous inference problem as an optimization task.

$$\hat{\boldsymbol{\phi}} = \arg\min_{\boldsymbol{\phi}} KL(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z}|\mathbf{x})) \tag{2.38}$$

Of course, the KL-divergence is not directly computable as it depends on $p(\mathbf{x})$, which basically leads us back to the start of our problem [Blei et al. 2017].

$$KL(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z}|\mathbf{x})) = \mathop{\mathbb{E}}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\Big[\log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\Big] - \mathop{\mathbb{E}}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x},\mathbf{z})] + \log p_{\theta}(\mathbf{x}) \tag{2.39}$$

The term $\log p_{\theta}(\mathbf{x})$ does not depend on the variational parameters $\boldsymbol{\phi}$, so it can be omitted from the optimization criterium for optimizing with regards to $\boldsymbol{\phi}$.

$$\hat{\boldsymbol{\phi}} = \arg\min_{\boldsymbol{\phi}} KL(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z}|\mathbf{x})) \tag{2.40}$$

$$= \arg\min_{\boldsymbol{\phi}} \mathop{\mathbb{E}}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\Big[\log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\Big] - \mathop{\mathbb{E}}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x},\mathbf{z})] \tag{2.41}$$

$$= \arg\max_{\boldsymbol{\phi}} \text{ELBO}_{\mathbf{x}}(\boldsymbol{\phi}) \tag{2.42}$$

This reveals that the newly found objective for minimizing the KL-divergence is the ELBO that was already previously derived using Jensen's inequality (see 2.36). In contrast to before, the ELBO now needs to be maximized at the same time with regards to the generative model parameters $\boldsymbol{\theta}$ and the newly defined variational parameters $\boldsymbol{\phi}$ [Kingma and Welling 2019]. By optimizing them jointly we ensure that the quality of the generative model improves by maximizing an approximation of the log-likelihood $p_{\boldsymbol{\theta}}(\mathbf{x})$, while the KL-divergence is minimized resulting in a better approximate posterior $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$.

$$\begin{aligned}\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\phi}} &= \arg\max_{\boldsymbol{\theta},\boldsymbol{\phi}} \log p_{\boldsymbol{\theta}}(\mathbf{x}) \\ &\geq \arg\max_{\boldsymbol{\theta},\boldsymbol{\phi}} \text{ELBO}_{\mathbf{x}}(\boldsymbol{\theta},\boldsymbol{\phi}) \\ &= \arg\max_{\boldsymbol{\theta},\boldsymbol{\phi}} \mathop{\mathbb{E}}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\log p_{\theta}(\mathbf{x},\mathbf{z})] - \mathop{\mathbb{E}}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\Big[\log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\Big]\end{aligned} \tag{2.43}$$

With that, the central optimization criterium of the variational autoencoder is found. The next sections will deal with how to calculate or approximate the individual ELBO terms, and how they can be maximized using gradient descent [Kingma and Welling 2014; Kingma and Welling 2019]. In the framework of variational autoencoders, the variational density $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ is also called encoder or recognition model, and the conditional distribution of the observations $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ is referred to as decoder or observation model.

2.3.2 Gaussian Posterior Approximation

A common choice for the family of the variational density is a multivariate Gaussian distribution with the mean $\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}) \in \mathbb{R}^{TM\times 1}$ and the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\phi}}(\mathbf{x}) \in \mathbb{R}^{TM\times TM}$, where T refers to the total number of time steps and M to the number of latent dimensions [Kingma and Welling 2019].

$$q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}), \boldsymbol{\Sigma}_{\boldsymbol{\phi}}(\mathbf{x})) \tag{2.44}$$

The variational parameters $\boldsymbol{\phi}$ define a functional mapping between the data points $\mathbf{x}$ and the parameters of the approximate posterior. The key idea is that the variational parameters $\boldsymbol{\phi}$ are shared between all observations, which makes training efficient and allows variational autoencoders to scale better for larger datasets, as the number of variational parameters is now independent from the dataset size [Gershman and Goodman 2014; Kingma and Welling 2014; Kingma and Welling 2019]. This approach is called amortized inference, and avoids the costly direct optimization of the distributional parameters for each data point as employed by more traditional variational methods, e.g. stochastic variational inference uses a mean-field approximation and optimizes the variational parameters locally for each data point [Hoffman et al. 2013; Cremer et al. 2018; Kingma and Welling 2019]. The usage of a shared encoder network also makes it straightforward to determine the posterior distribution for previously unknown observations, as we can simply input them into the encoder network without repeating the optimization. Amortized inference can also come at a cost, as it is technically more restrictive than directly determining the distributional parameters. This difference is also called the amortization gap [Cremer et al. 2018].
Typically, for variational autoencoders the mean and covariance are determined through deep neural networks. The specific parameterization chosen in this thesis will be discussed in later sections.

$$(\boldsymbol{\mu}_{\boldsymbol{\phi}}, \boldsymbol{\Sigma}_{\boldsymbol{\phi}}) = \text{NeuralNet}_{\boldsymbol{\phi}}(\mathbf{x}) \tag{2.45}$$

The reasonably simple choice of a multivariate normal distribution has the benefit that it allows to analytically write down the entropy term in the ELBO (see Equation 2.43) [Ahmed and Gokhale 1989].

$$\mathop{\mathbb{E}}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\Big[\log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\Big] = -\frac{TM}{2}(\log(2\pi)+1) - \frac{1}{2}\log|\boldsymbol{\Sigma}_{\boldsymbol{\phi}}(\mathbf{x})| \tag{2.46}$$

Thus, we can restate the ELBO as:

$$\begin{aligned}\text{ELBO}_{\mathbf{x}}(\boldsymbol{\theta},\boldsymbol{\phi}) &= \mathop{\mathbb{E}}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x},\mathbf{z})] - \mathop{\mathbb{E}}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\right] \\ &= \mathop{\mathbb{E}}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x},\mathbf{z})] + \frac{TM}{2}(\log(2\pi)+1) + \frac{1}{2}\log|\boldsymbol{\Sigma}_{\boldsymbol{\phi}}(\mathbf{x})| \end{aligned} \tag{2.47}$$

2.3.3 Monte Carlo Estimates

Generally, the expectancy value of the generative model $\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x},\mathbf{z})]$ can not be analytically determined. However, we can use the principle of Monte Carlo integration to gain an estimate for the expected value. An approximation of an expectation of a measurable function $f(x)$ of a random variable $x \sim p(x)$

$$\mathbb{E}[f(x)] = \int f(x)p(x)\,dx \tag{2.48}$$

is given by the Monte Carlo estimate,

$$\mathbb{E}[f(x)] = \int f(x)p(x)\,dx \approx \frac{1}{N}\sum_{n=1}^{N} f(x_n) \tag{2.49}$$

where $x_1,\ldots,x_N$ are independent, identically distributed observations sampled from the probability distribution $p(x)$ [Weinzierl 2000]. The Monte Carlo estimator is unbiased, and converges towards the true value for $N \to \infty$. We can utilize this result to formulate an unbiased and consistent estimate of the ELBO.

$$\begin{aligned}\text{ELBO}_{\mathbf{x}}(\boldsymbol{\theta},\boldsymbol{\phi}) &= \mathop{\mathbb{E}}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x},\mathbf{z})] + \frac{TM}{2}(\log(2\pi)+1) + \frac{1}{2}\log|\boldsymbol{\Sigma}_{\boldsymbol{\phi}}(\mathbf{x})| \\ &\approx \frac{1}{L}\sum_{l=1}^{L}\log p_\theta(\mathbf{x},\mathbf{z}^{(l)}) + \frac{TM}{2}(\log(2\pi)+1) + \frac{1}{2}\log|\boldsymbol{\Sigma}_{\boldsymbol{\phi}}(\mathbf{x})| \end{aligned} \tag{2.50}$$

The samples $\mathbf{z}^{(l)}$ are obtained by repeatedly drawing from the approximative posterior distribution $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ given the observed data $\mathbf{x}$.

2.3.4 Stochastic Gradient Variational Bayes

As mentioned earlier, the maximization of the ELBO with regards to the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ will be performed numerically using gradient descent. To do so, we need to compute the gradients of the ELBO with respect to $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ [Kingma and Welling 2019].

The gradient with respect to the generative model parameters $\boldsymbol{\theta}$ can be pulled into the expectation values as the approximate posterior does not depend on $\boldsymbol{\theta}$. For the same reason, the entropy term vanishes, leaving us only with:

$$\begin{aligned}\nabla_{\boldsymbol{\theta}}\mathrm{ELBO}_{\mathbf{x}}(\boldsymbol{\theta},\boldsymbol{\phi}) &= \nabla_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\Big[\log p_\theta(\mathbf{x},\mathbf{z}) - \log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\Big] \\ &= \mathop{\mathbb{E}}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\nabla_{\boldsymbol{\theta}} \log p_\theta(\mathbf{x},\mathbf{z})] \end{aligned} \tag{2.51}$$

The expectation can be approximated fairly straightforward with the Monte Carlo estimate previously discussed (see Section 2.3.3).

$$\nabla_{\boldsymbol{\theta}}\mathrm{ELBO}_{\mathbf{x}}(\boldsymbol{\theta},\boldsymbol{\phi}) = \mathop{\mathbb{E}}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\nabla_{\boldsymbol{\theta}} \log p_\theta(\mathbf{x},\mathbf{z})] \approx \frac{1}{L}\sum_{l=1}^{L} \nabla_{\boldsymbol{\theta}} \log p_\theta(\mathbf{x},\mathbf{z}^{(l)}) \tag{2.52}$$

Calculating the gradient with respect to the variational parameters $\boldsymbol{\phi}$ is more difficult, since the approximate posterior $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ obviously depends on the variational parameters $\boldsymbol{\phi}$. In contrast to before, we can not immediately use the Monte Carlo estimate of the expectation value as the term would lose its the dependency on $\boldsymbol{\phi}$ and evaluate to zero $\nabla_{\boldsymbol{\phi}} \frac{1}{L}\sum_{l=1}^{L} \log p_\theta(\mathbf{x},\mathbf{z}^{(l)}) = 0$, which makes gradient descent based training impossible.

The gradient can first be moved into the expectation value by using the log-derivative trick $\nabla_{\boldsymbol{\phi}} q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) = q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\nabla_{\boldsymbol{\phi}} \log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$, which in turn allows us to compute the Monte Carlo estimate that is also called the score-function estimator [Kleijnen and Rubinstein 1996; Mnih and Gregor 2014].

$$\begin{aligned}\nabla_{\boldsymbol{\phi}} \mathop{\mathbb{E}}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x},\mathbf{z})] &= \mathop{\mathbb{E}}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\Big[\log p_\theta(\mathbf{x},\mathbf{z})\nabla_{\boldsymbol{\phi}} \log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\Big] \\ &\approx \frac{1}{L}\sum_{l=1}^{L} \log p_\theta(\mathbf{x},\mathbf{z}^{(l)})\nabla_{\boldsymbol{\phi}} \log q_{\boldsymbol{\phi}}(\mathbf{z}^{(l)}|\mathbf{x}),\ \mathbf{z}^{(l)} \overset{\text{iid}}{\sim} q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \end{aligned} \tag{2.53}$$

The estimator is unbiased, but has the severe disadvantage that it oftentimes exhibits very high variance [Paisley et al. 2012; Kingma and Welling 2014], which makes it impractical for learning many models. A central idea of the variational autoencoder framework is the introduction of a different type of gradient estimator using the so called reparameterization trick that performs better in practice [Kingma and Welling 2014].

2.3.4.1 Reparameterization Trick

As stated above, the expectancy value can not be approximated using Monte Carlo sampling before calculating the gradient, because the expression would lose its dependency on the variational parameters $\boldsymbol{\phi}$. To tackle this problem, the approximate posterior distribution $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ can be reformulated by introducing a differentiable and invertible deterministic function parameterized by $\boldsymbol{\phi}$ [Kingma and Welling 2014; Kingma and Welling 2019].

$$\mathbf{z} = g_{\boldsymbol{\phi}}(\mathbf{x}, \boldsymbol{\epsilon}),\ \boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon}) \tag{2.54}$$

$\boldsymbol{\epsilon}$ is a random variable distributed according to a probability distribution $p(\boldsymbol{\epsilon})$ that is independent of the variational parameters $\boldsymbol{\phi}$ and the observations $\mathbf{x}$. Sampling from the approximate posterior $\mathbf{z}^{(l)} \sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ is now a two step process. First, a sample is drawn from the auxiliary noise variable $\boldsymbol{\epsilon}^{(l)} \sim p(\boldsymbol{\epsilon})$, which we input together with the observations $\mathbf{x}$ into the deterministic transformation to generate $\mathbf{z}^{(l)}$. The mapping $g_{\boldsymbol{\phi}}(\mathbf{x}, \boldsymbol{\epsilon})$ and the distribution $p(\boldsymbol{\epsilon})$ need to be chosen in such a way that the resulting distribution stays the same. It is possible to find such a transformation for many distributions, for instance for any location-scale family distribution, such as Gaussian or Logistic distributions, or for distributions with a tractable inverse cumulative density function, e.g. Exponential or Cauchy distributions [Kingma and Welling 2014].
For the multivariate Gaussian approximate posterior $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}), \boldsymbol{\Sigma}_{\boldsymbol{\phi}}(\mathbf{X}))$, introduced in Section 2.3.2, such a mapping can easily be formulated. The auxiliary random variable $\boldsymbol{\epsilon}$ is assumed to be distributed according to a standard normal distribution,

$$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbb{1}) \tag{2.55}$$

which yields:

$$\mathbf{z} = g_{\boldsymbol{\phi}}(\mathbf{x}, \boldsymbol{\epsilon}) = \boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}) + \boldsymbol{\Sigma}_{\boldsymbol{\phi}}(\mathbf{x})^{1/2}\boldsymbol{\epsilon} \tag{2.56}$$

for the deterministic function, where $\boldsymbol{\Sigma}^{1/2}$ is the Cholesky decomposition of $\boldsymbol{\Sigma}$. A similar formulation can be used for all location-scale family distributions (location + scale $\times\epsilon$).
Once such a function has been found, we seek to express the expectation values in the ELBO by a change of variables $\mathbf{z} = g(\boldsymbol{\epsilon})$.

$$\mathop{\mathbb{E}}_{\mathbf{z}}[h(\mathbf{z})] = \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}}[h(g(\boldsymbol{\epsilon}))] \tag{2.57}$$

This follows from how probability densities can be reformulated into each other via a strictly monotonic function g [Devore and Berk 2012].

$$f_{\mathbf{z}}(\mathbf{z}) = f_{\boldsymbol{\epsilon}}(g^{-1}(\mathbf{z})) \left|\det \frac{dg^{-1}(\mathbf{z})}{d\mathbf{z}}\right| \tag{2.58}$$

The transformation from above can then be inserted into the expectation value to arrive at equation 2.57.

$$\begin{aligned}
\mathop{\mathbb{E}}_{\mathbf{z}}[h(\mathbf{z})] &= \int_{\mathbf{z}} f_{\mathbf{z}}(\mathbf{z}) h(\mathbf{z})\, d\mathbf{z} \\
&= \int_{\boldsymbol{\epsilon}} f_{\boldsymbol{\epsilon}}(g^{-1}(\mathbf{z})) \left|\det \frac{\mathrm{d}g^{-1}(\mathbf{z})}{d\mathbf{z}}\right| h(\mathbf{z}) \left|\det \frac{d\mathbf{z}}{d\boldsymbol{\epsilon}}\right| d\boldsymbol{\epsilon} \\
&= \int_{\boldsymbol{\epsilon}} f_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon}) \left|\det \frac{d\boldsymbol{\epsilon}}{d\mathbf{z}}\right| h(g(\boldsymbol{\epsilon})) \left|\det \frac{d\mathbf{z}}{d\boldsymbol{\epsilon}}\right| d\boldsymbol{\epsilon} \\
&= \int_{\boldsymbol{\epsilon}} f_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon}) h(g(\boldsymbol{\epsilon}))\, d\boldsymbol{\epsilon} = \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}}[h(g(\boldsymbol{\epsilon})]
\end{aligned} \tag{2.59}$$

In the second line the probability distribution is transformed, and the variables of integration are changed. Finally, by exploiting the inversion function theorem for the Jacobian matrices, we can see that both determinants cancel each other out, and that the equality holds true.
Equipped with this result, the expectation values in the ELBO can be rewritten. Here we still assume a multivariate Gaussian approximate posterior, while in other cases

where the entropy term is not analytically tractable we could reformulate the expression in similar fashion.

$$\begin{aligned}\text{ELBO}_{\mathbf{x}}(\boldsymbol{\theta}, \boldsymbol{\phi}) &= \mathop{\mathbb{E}}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}, \mathbf{z})] + \frac{TM}{2}(\log(2\pi) + 1) + \frac{1}{2}\log|\boldsymbol{\Sigma}_{\boldsymbol{\phi}}(\mathbf{x})| \\ &= \mathop{\mathbb{E}}_{p(\boldsymbol{\epsilon})}\big[\log p_\theta(\mathbf{x}, g_{\boldsymbol{\phi}}(\mathbf{x}, \boldsymbol{\epsilon}))\big] + \frac{TM}{2}(\log(2\pi) + 1) + \frac{1}{2}\log|\boldsymbol{\Sigma}_{\boldsymbol{\phi}}(\mathbf{x})| \quad (2.60)\end{aligned}$$

We can now use Monte Carlo sampling to arrive at the famous Stochastic Gradient Variational Bayes (SGVB) estimator of the ELBO [Kingma and Welling 2014].

$$\text{ELBO}_{\mathbf{x}}(\boldsymbol{\theta}, \boldsymbol{\phi}) \approx \frac{1}{L}\sum_{l=1}^{L}\log p_\theta(\mathbf{x}, g_{\boldsymbol{\phi}}(\mathbf{x}, \boldsymbol{\epsilon}^{(l)})) + \frac{TM}{2}(\log(2\pi) + 1) + \frac{1}{2}\log|\boldsymbol{\Sigma}_{\boldsymbol{\phi}}(\mathbf{x})| \quad (2.61)$$

As the expectation is now with respect to $p(\boldsymbol{\epsilon})$, and the source of randomness has been separated from the variational parameters $\boldsymbol{\phi}$, the gradient can now be simply pulled inside the expectation. This estimator exhibits much lower variances than the score function estimator mentioned earlier [Rezende et al. 2014].

$$\begin{aligned}\nabla_{\boldsymbol{\phi}} \mathop{\mathbb{E}}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}, \mathbf{z})] &= \nabla_{\boldsymbol{\phi}} \mathop{\mathbb{E}}_{p(\boldsymbol{\epsilon})}\big[\log p_\theta(\mathbf{x}, g_{\boldsymbol{\phi}}(\mathbf{x}, \boldsymbol{\epsilon}))\big] \\ &= \mathop{\mathbb{E}}_{p(\boldsymbol{\epsilon})}\big[\nabla_{\boldsymbol{\phi}} \log p_\theta(\mathbf{x}, g_{\boldsymbol{\phi}}(\mathbf{x}, \boldsymbol{\epsilon}))\big] \\ &\approx \frac{1}{L}\sum_{l=1}^{L}\nabla_{\boldsymbol{\phi}} \log p_\theta(\mathbf{x}, g_{\boldsymbol{\phi}}(\mathbf{x}, \boldsymbol{\epsilon}^{(l)})) \quad (2.62)\end{aligned}$$

2.3.5 ELBO for Sequential Variational Autoencoders

So far, the entire formulation of the variational autoencoder framework was independent of the fact that we are considering time series data. We will now take a closer look at the recognition and generative model terms of the ELBO in the setting of state space models.

2.3.5.1 Recognition Model

As discussed in Section 2.3.2, an approximate Gaussian posterior $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}), \boldsymbol{\Sigma}_{\boldsymbol{\phi}}(\mathbf{x}))$ is a common choice for the recognition model, as it allows the analytical calculation of the entropy term and makes it fairly straightforward to perform the reparameterization trick. Generally, the parameters are calculated using

neural networks, although the exact parameterization and network architecture can vary.

$$(\boldsymbol{\mu}_{\boldsymbol{\phi}}, \boldsymbol{\Sigma}_{\boldsymbol{\phi}}) = \text{NeuralNet}_{\boldsymbol{\phi}}(\mathbf{x}) \tag{2.63}$$

A general problem with a Gaussian posterior is that the number of parameters scales quadratically in time ($\boldsymbol{\Sigma}_{\boldsymbol{\phi}} \in \mathbb{R}^{TM \times TM}$), making training larger datasets computationally unfeasible. To counter this, we can make different simplifying assumptions about the structure of the covariance matrix, e.g. a block-tridiagonal covariance matrix [Archer et al. 2015]. We will make use of the mean field assumption [Bishop 2006], which allows us to factorize the approximate posterior across time points and latent space.

$$q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) = \prod_{t=1}^{T} \prod_{m=1}^{M} q_{\boldsymbol{\phi}}^{(tm)}(z_{tm}|\mathbf{x}) \tag{2.64}$$

We therefore assume that the covariance matrix is completely diagonal, which greatly decreases training time. Of course by neglecting correlations between time points and states, it also significantly reduces the expressivity of the approximate posterior [Blei et al. 2017]. The mean and variance at each time point is then parameterized by multiple convolutional neural network layers [Warkentin 2021]. A CNN is used in the hopes that it might allow for the extraction of important temporal features from the observations that can then be encoded into the latent space [Cui et al. 2016; Zhao et al. 2017]. More specifically, a 4-layer CNN is used for computing the mean $\boldsymbol{\mu}_t \in \mathbb{R}^M$, and a 1-layer CNN for the log-variance $\log \boldsymbol{\sigma}_t^2 \in \mathbb{R}^M$, as recommended in Brenner et al. 2021; Warkentin 2021.

$$(\boldsymbol{\mu}_{\boldsymbol{\phi}}, \log \boldsymbol{\sigma}_{\boldsymbol{\phi}}^2) = (\text{CNN}_{\boldsymbol{\phi}_\mu}(\mathbf{x}), \text{CNN}_{\boldsymbol{\phi}_\Sigma}(\mathbf{x})) \in (\mathbb{R}^{T \times M}, \mathbb{R}^{T \times M}) \tag{2.65}$$

For the computation of the mean or log-variance vector at a specific time step t, the CNN considers the $2k$ observations before the time point $\{\mathbf{x}_{t-2k}, \ldots, \mathbf{x}_t\}$. In contrast to previous implementations [Warkentin 2021], I did not position the kernel window symmetrically around the time points to better respect the causal structure of the time series, which also makes it more straightforward to sample an initial latent

state for ahead prediction. For the first $2k$ time points a reflection padding is used. Finally, due to the diagonal covariance structure, the entropy term in the ELBO can be written as a simple sum across variances.

$$\mathbb{H}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})) = \frac{TM}{2}(\log(2\pi)+1) + \frac{1}{2}\sum_{t=1}^{T}\sum_{m=1}^{M}\log \boldsymbol{\sigma}^2_{\boldsymbol{\phi}}(\mathbf{x})_{tm} \tag{2.66}$$

2.3.5.2 ELBO of the PLRNN

After discussing the recognition model and the entropy term of the ELBO the last puzzle piece missing is the likelihood of the generative model. As latent model a PLRNN is employed, and as observation model we for now consider a multivariate Gaussian distribution, see Section 2.2.3.

$$\begin{aligned} p_{\boldsymbol{\theta}}(\mathbf{z}_t|\mathbf{z}_{t-1}) &= \mathcal{N}(\mathbf{A}\mathbf{z}_{t-1} + \mathbf{W}\sum_{b=1}^{B}\alpha_b \max(0, \mathbf{z}_{t-1} - \mathbf{h}_b) + \mathbf{h}, \boldsymbol{\Sigma}) \\ &= \mathcal{N}(\boldsymbol{\mu}_{z_t}(\mathbf{z}_{t-1}), \boldsymbol{\Sigma}) \end{aligned} \tag{2.67}$$

$$p_{\boldsymbol{\theta}}(\mathbf{x}_t|\mathbf{z}_t) = \mathcal{N}(\mathbf{B}\mathbf{z}_t, \boldsymbol{\Gamma}) = \mathcal{N}(\boldsymbol{\mu}_{x_t}(\mathbf{z}_t), \boldsymbol{\Gamma}) \tag{2.68}$$

Using the factorization of the joint log-likelihood, see equation 2.19, we can write:

$$\begin{aligned} \log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) &= \log p_{\boldsymbol{\theta}}(\mathbf{x}_1, \ldots, \mathbf{x}_T, \mathbf{z}_1, \ldots, \mathbf{z}_T) \\ &= \log p_{\boldsymbol{\theta}}(\mathbf{z}_1) + \sum_{t=2}^{T}\log p_{\boldsymbol{\theta}}(\mathbf{z}_t|\mathbf{z}_{t-1}) + \sum_{t=1}^{T}\log p_{\boldsymbol{\theta}}(\mathbf{x}_t|\mathbf{z}_t) \\ &= -\frac{M}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Sigma}_0| - \frac{1}{2}(\mathbf{z}_1 - \boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}_0^{-1}(\mathbf{z}_1 - \boldsymbol{\mu}_0) \\ &\quad + \sum_{t=2}^{T}\left(-\frac{M}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2}(\mathbf{z}_t - \boldsymbol{\mu}_{z_t})^T\boldsymbol{\Sigma}^{-1}(\mathbf{z}_t - \boldsymbol{\mu}_{z_t})\right) \\ &\quad + \sum_{t=1}^{T}\left(-\frac{N}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Gamma}| - \frac{1}{2}(\mathbf{x}_t - \boldsymbol{\mu}_{x_t})^T\boldsymbol{\Gamma}^{-1}(\mathbf{x}_t - \boldsymbol{\mu}_{x_t})\right) \end{aligned} \tag{2.69}$$

2.3.6 Training Overview

We arrive at the final optimization criterium by combining the entropy term from equation 2.66 and the likelihood of the generative model from the section before (see equation 2.69).

$$
\begin{aligned}
\mathrm{ELBO}_{\mathbf{x}}(\boldsymbol{\theta},\boldsymbol{\phi}) \approx & \frac{1}{L}\sum_{l=1}^{L}\log p_\theta(\mathbf{x}, g_{\boldsymbol{\phi}}(\mathbf{x},\boldsymbol{\epsilon}^{(l)})) + \mathbb{H}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})) \\
= & \frac{1}{L}\sum_{l=1}^{L}\Bigg(-\frac{M}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Sigma}_{\mathbf{0}}| - \frac{1}{2}(\mathbf{z}_1^{(l)} - \boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}_0^{-1}(\mathbf{z}_1^{(l)} - \boldsymbol{\mu}_0) \\
& + \sum_{t=2}^{T}\Bigg(-\frac{M}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2}(\mathbf{z}_t^{(l)} - \boldsymbol{\mu}_{z_t^{(l)}})^T\boldsymbol{\Sigma}^{-1}(\mathbf{z}_t^{(l)} - \boldsymbol{\mu}_{z_t^{(l)}})\Bigg) \\
& + \sum_{t=1}^{T}\Bigg(-\frac{N}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Gamma}| - \frac{1}{2}(\mathbf{x}_t - \boldsymbol{\mu}_{x_t})^T\boldsymbol{\Gamma}^{-1}(\mathbf{x}_t - \boldsymbol{\mu}_{x_t})\Bigg)\Bigg) \\
& + \frac{TM}{2}(\log(2\pi)+1) + \frac{1}{2}\sum_{t=1}^{T}\sum_{m=1}^{M}\log \boldsymbol{\sigma}^2_{\boldsymbol{\phi}}(\mathbf{x})_{tm}
\end{aligned} \tag{2.70}
$$

The ELBO is maximized by gradient descent with regards to the generative and recognition model parameters $\boldsymbol{\theta}, \boldsymbol{\phi}$. To do so, it is common to employ mini-batch gradient descent [Kingma and Welling 2019], which corresponds to drawing random subsets of the data and performing the gradient updates mini-batch-wise. The stochasticity introduced by sampling mini-batches reduces the danger of the optimization procedure getting stuck in saddle points or local minima, while at the same time usually being more computationally efficient.

For most of the empirical investigation in this thesis, gradient updates were performed over the entire time series at once. The empirical dataset that will later be used for evaluation only contains very short time series, so it is unfeasible to split them up without destroying much of the temporal structure. Notice that the training procedure is still stochastic, as samples $\mathbf{z}^{(l)}$ need to drawn from the encoder. The entire training procedure also has been dubbed Auto-Encoding Variational Bayes [Kingma and Welling 2014]. As optimization method `Adam` from the `PyTorch` package is used [Kingma and Ba 2015].

Figure 4 illustrates the computation of the ELBO estimator during one training epoch. The time series data $\mathbf{x}$ is used as input for the recognition model, which calculates the mean $\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x})$ and the covariance $\boldsymbol{\Sigma}_{\boldsymbol{\phi}}(\mathbf{x})$ of the Gaussian approximate posterior $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$. The covariance matrix can then directly be used to determine the entropy term $\mathbb{H}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}))$. The reparameterization trick is then exploited, and L samples $\mathbf{z}^{(l)} = \mu_{\boldsymbol{\phi}}(\mathbf{x}) + \Sigma_{\boldsymbol{\phi}}(\mathbf{x})^{\frac{1}{2}}\boldsymbol{\epsilon}^{(l)}$ are drawn with the aid of an external random variable $p(\boldsymbol{\epsilon}) \sim \mathcal{N}(\mathbf{0}, \mathbb{1})$. Commonly, it is sufficient to draw a single sample during each epoch [Kingma and Welling 2014]. Next, the samples $\mathbf{z}^{(l)}$ are fed into the genera-

tive model, where they are used to compute the likelihood terms of the observation model and the PLRNN respectively. Together they determine the joint log-likelihood of the generative model $p_\theta(\mathbf{x}, \mathbf{z})$.

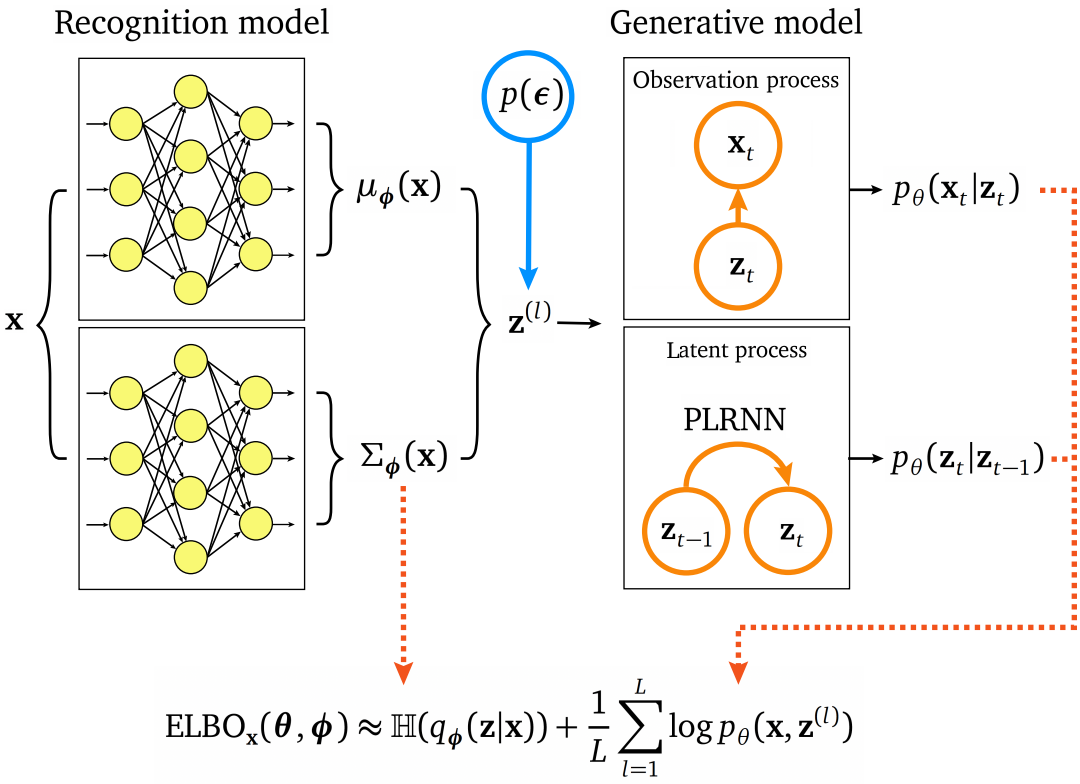


Figure 4: Diagram illustrating the process of calculating the ELBO estimator during maximum likelihood training of the sequential variational autoencoder.

2.3.7 Multimodal VAE

The usage of smartphones and other wearable devices allows for the collection of a large number of data modalities. In addition to providing an easy way to administer questionnaires, a variety of passive sensor data can be gathered that does not require active participant input [Koppe et al. 2019b]. For instance, it is possible to monitor location data, step counts, app usage, music listened to or phone calls and SMS activity.[3] Barring privacy concerns, this type of data can of course be collected

[3]For more examples for mobile sensing see: `https://docs.movisens.com/movisensXS/mobile_sensing/#features-library-version-version`

much more liberally, as one does not need to worry about tiring the participant with a high sampling frequency. In addition, it can also provide important context information on the behavior and emotional state of the subject that is not biased by self-reporting, e.g. sleep quality has a great effect on next-day mood [Triantafillou et al. 2019].

Bommer et al. 2021 show how the PLRNN framework can be extended to include multiple data modalities. For each modality, a new observation model needs to be introduced. Assuming conditional independence between the different measurements, the overall observation likelihood simply factorizes into the contributions of the different data types. For two modalities $\mathbf{x}$ and $\mathbf{q}$ we find:

$$p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{q}|\mathbf{z}) = p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) p_{\boldsymbol{\theta}}(\mathbf{q}|\mathbf{z}) \tag{2.71}$$

Thus, the ELBO can be modified in very straightforward way by simply adding the likelihood of each additional modality.

$$\begin{aligned} \text{ELBO}_{\mathbf{x}}(\boldsymbol{\theta}, \boldsymbol{\phi}) \approx & \frac{1}{L} \sum_{l=1}^{L} \left(\log p_{\theta}(\mathbf{x}|\mathbf{z}^{(l)}) + \log p_{\theta}(\mathbf{q}|\mathbf{z}^{(l)}) + \log p_{\theta}(\mathbf{z}^{(l)}) \right) && (2.72) \\ & + \mathbb{H}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}, \mathbf{q})) && (2.73) \end{aligned}$$

For the recognition model all data modalities can now be used as input for the neural networks parameterizing the approximate posterior $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{q}), \boldsymbol{\Sigma}_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{q}))$. Adapted from Tombolini 2021, a separate CNN is defined for each data modality. The output of all networks is then concatenated by a single feed-forward layer to arrive at the final mean (or covariance) matrix.

As an example for a potential sensor modality, I re-implemented and optimized the run time of the Zero-inflated Poisson (ZIP) model [Lambert 1992] inspired by code from Tombolini 2021. The ZIP model can be used for count data that is strongly zero-inflated. This situation might for instance occur when monitoring the step counts of participants, where for many hours of the day, e.g. during sleep phases or while sitting at the office, only zero entries are reported. The ZIP model splits up the process that generates the zero observations from the rest of the counts. A count variable q_{ti} at time step t and for feature i is zero with a probability of π_{ti} and Poisson-

distributed with probability $1-\pi_{ti}$. Notice that in the second case zero counts can still be drawn from the Poisson distribution.

$$p(q_{ti}|\mathbf{z}_t) = \begin{cases} \pi_{ti} + (1-\pi_{ti})e^{-\lambda_{ti}} & \text{if } q_{ti} = 0 \\ (1-\pi_{ti})\frac{\lambda_{ti}^{q_{ti}}}{q_{ti}!}e^{-\lambda_{ti}} & \text{if } q_{ti} > 0 \end{cases} \tag{2.74}$$

The parameters π_{ti} and λ_{ti} are determined by the latent states $\mathbf{z}_t$ through a generalized linear model with a logit and a log-link function, similar to what will be used for the ordinal observations in Section 3.3.3.
In the context of this thesis, I was not able to perform an empirical investigation of the multimodal setup, as the empirical dataset that I primarily worked with does not contain such modalities. In the spirit of building up a modeling framework for future use, I still deemed it important to integrate the multimodal aspect at least as a simple test case.

2.4 Manifold Attractor Regularization

Many time series models have severe problems dealing with slower time scales and sufficiently capturing long-range dependencies. To mitigate this problem for PLRNNs, the so called manifold attractor regularization has been proposed by Schmidt et al. 2021. The general idea is to regularize the parameter matrices $\mathbf{A}, \mathbf{W}, \mathbf{h}$ for a subset of the latent states $M_{\text{reg}} \leq M$. More specifically, we seek to push parts of the diagonal matrix $\mathbf{A}$ towards the identity matrix, and the off-diagonal matrix $\mathbf{W}$ and the vector $\mathbf{h}$ to $\mathbf{0}$. This is achieved by adding the following penalty term to the loss (or respectively subtracting it from the ELBO).

$$\mathcal{L}_{\text{reg}} = \lambda \sum_{m=1}^{M_{reg}} (A_{m,m} - 1)^2 + \lambda \sum_{m=1}^{M_{reg}} \sum_{n=1, n\neq m}^{M} W_{m,n}^2 + \lambda \sum_{m=1}^{M_{reg}} h_m^2 \tag{2.75}$$

To give a brief motivation for the expression, we can reformulate the PLRNN by introducing a diagonal matrix $\mathbf{D}^b_{\Omega(t)} = \alpha_b \text{diag}(\mathbf{d}^b_{\Omega(t)})$ for each basis, where $\mathbf{d}^b_{\Omega(t)} = (d_{\Omega(t)1}, \ldots, d_{\Omega(t)M})$ is a vector of indicator functions such that $d_{\Omega(t)m} = 1$ if $(z_{(t-1)m} - h_m) > 0$ and zero in the other case [Brenner et al. 2021; Monfared et al. 2021]. This gives us the following PLRNN equation with $\mathbf{D}^B_{\Omega(t)} = \sum_{b=1}^{B} \mathbf{D}^b_{\Omega(t)}$:

$$\mathbf{z}_t = \mathbf{A}\mathbf{z}_{t-1} + \mathbf{W}\sum_{b=1}^{B}\mathbf{D}^b_{\Omega(t)}\mathbf{z}_{t-1} + \mathbf{h} = (\mathbf{A} + \mathbf{W}\mathbf{D}^B_{\Omega(t)})\mathbf{z}_{t-1} + \mathbf{h} \tag{2.76}$$

We therefore find for the Jacobian of the PLRNN:

$$\mathbf{J}_t = \frac{\partial \mathbf{z}_t}{\partial \mathbf{z}_{t-1}} = (\mathbf{A} + \mathbf{W}\mathbf{D}^B_{\Omega(t)}) \tag{2.77}$$

We can now consider the Jacobian of two temporally distant latent states $\mathbf{z}_t$ and $\mathbf{z}_{t'},\ t \gg t'$ [Monfared et al. 2021].

$$\frac{\partial \mathbf{z}_t}{\partial \mathbf{z}_{t'}} = \prod_{i=0}^{t-t'-1}\frac{\partial \mathbf{z}_{t-i}}{\partial \mathbf{z}_{t-i-1}} = \prod_{i=0}^{t-t'-1}\mathbf{J}_{t-i} = \prod_{i=0}^{t-t'-1}(\mathbf{A} + \mathbf{W}\mathbf{D}^B_{\Omega(t-i)}) \tag{2.78}$$

If the largest absolute eigenvalue of the Jacobians is on average (according to the geometric mean $\left\|\prod_{i=0}^{t-t'-1}\mathbf{J}_{t-i}\right\|^{1/(t-t')}$) smaller or larger than one, the gradients will either tend to vanish or explode for $(t-t') \to \infty$ [Monfared et al. 2021]. This would imply that the model is unable to express long-range dependencies, which strongly links the question of preserving memory to the behavior of the gradients.
While we could simply enforce $\mathbf{A} = \mathbf{I}$ and $\mathbf{W} = \mathbf{0}$ to ensure that the Jacobians stay bounded, we in turn lose the expressivity of the system required to reconstruct a wide range of dynamical behavior. This would also be excessive, as we only require a system that can express long term dependencies, and do not need infinite memory. Fortunately, it can be shown that by only regularizing the matrices for a subset of the latent states, the Jacobians still stay bounded from above and below [Monfared et al. 2021; Schmidt et al. 2021]. This hinges on the assumption that the non-regularized latent states converge toward a stable fixed-point or k-cycle and do not exhibit chaotic behavior, which ensures that the gradients stay bounded from above. Conceptually speaking, the latent space is split in two, where part takes on the role of memory states, while the non-regularized latent states allow the model to retain its flexibility. As the regularization term does not scale with the number of time points T, we divide the rest of the ELBO by T to ensure an equal weighting.

2.5 Interventions in the PLRNN

As discussed in the introduction, one of the main goals for the psychiatric use case is to use the model forecasts to recommend different ecological momentary interventions to the participants. The key part is to first be able to produce sensible predictions, as it is then relatively straightforward to integrate the intervention feedback loop into the framework. This can be done by simply adding an external input term $\mathbf{C}\mathbf{s}_t$ to the PLRNN equation [Koppe et al. 2019a].

$$\mathbf{z}_t = \mathbf{A}\mathbf{z}_{t-1} + \mathbf{W}\sum_{b=1}^{B}\alpha_b \max(0, \mathbf{z}_{t-1} - \mathbf{h}_b) + \mathbf{h} + \mathbf{C}\mathbf{s}_t + \boldsymbol{\epsilon}_t, \quad \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}). \tag{2.79}$$

$\mathbf{C}$ is a coefficient matrix, while $\mathbf{s}_t$ encodes if and what intervention was selected at time step t, for instance by denoting the different intervention types as a simple one-hot encoded vector. After model training, it is then possible to generate a forecast of the emotional trajectories under the assumption that a specific intervention is recommended at the current time point. All the forecasts for the different potential interventions can then be compared, and the intervention that leads to the best future development is sent to the participant. Of course a key question is, how to compare the different forecasts. For instance, a summary statistics could be constructed for each predicted time series that simply averages the different Likert items while weighing desired emotions as positive, and negative affects correspondingly as negative. A large variety of comparison measures could be thought of, e.g. it might be beneficial to recommend interventions that lead to more stable emotional trajectories. In the end, this questions needs to be tackled in conjunction with mental health specialists. It might even be interesting to allow users to select their priorities themselves, e.g. someone might want to primarily focus on reducing how stressed they are, while others could be interested in a different use case.

3 Model Implementation

In the previous section, the general theoretical background of the variational autoencoder model was discussed, and how it can be trained using Stochastic Variational Bayes. Next, I will discuss several extensions to the model framework that I implemented that enable us to deal with a variety of common problems one faces when working with empirical time series data. Most importantly, I will discuss how to handle missing values, adapt the generative model for ordinal data and how to use hierarchical parameter estimation to potentially increase the predictive strength of individual predictions by exploiting group level information. I will showcase these issues with an empirical dataset from a psychiatric study.
The model code is based on the general version of the SVAE code from Leonard Bereska, a former PhD student at DurstewitzLab. Additionally, I used the CNN encoder implementation from Paul Warkentin [Warkentin 2021], and took inspiration for my own implementation of the categorical observation model from Bommer et al. 2021.

3.1 EMI Compass Data

The empirical investigation of this thesis is mainly based upon data collected during the "EMIcompass" study [Rauschenberg et al. 2021b; Schick et al. 2021]. The study sought to understand the potential therapeutic benefits and effects of compassion-focused interventions through the use of EMIs administered via an application on the smartphones of participants. The target group were young individuals that struggled with psychotic, depressive or anxiety-related symptoms. In the course of the study, participants also completed ecological momentary assessments in regular intervals. The study was structured into four phases: baseline, intervention phase, post-treatment and a four-week follow up.
I focused on the EMA data collected during the baseline, post-treatment and follow up phase, as during these periods the sampling frequency was the highest. These periods lasted six days each, during which participants had to respond to eight EMA notifications per day. Participants could set the time interval themselves in which EMA prompts could appear each day. In the set interval, e.g. 8:00 to 23:00, the EMAs were randomly scheduled with a minimum time difference of 30 minutes. Completing the EMA questionnaire takes around two minutes on average , and participants had 15 minutes to respond to the notification. A participant ignoring, dis-

missing or failing to fully complete an EMA was separately logged. Around 44 to 47 questions were asked at each time point, with most items being categorical or ordinal in nature. I chose to focus on a subset of items that were likely the most relevant for future studies in the "Living lab AI4U" project motivated by its stronger focus on general public health. The items used are listed in Table 1, and measure momentary positive and negative affects, self-esteem and self-compassion. They are all on a 7-point Likert scale (from 1 to 7), with a rating of 1 meaning "not at all" and a rating of 7 "very much".

Table 1: Likert-scale features from the EMIcompass study used for model training.

Feature name	Scale of measure	Emotion
EMA_relaxed	ordinal (range 1-7)	I feel relaxed.
EMA_scared	ordinal (range 1-7)	I feel scared.
EMA_feeldown	ordinal (range 1-7)	I am feeling down.
EMA_angry	ordinal (range 1-7)	I feel angry.
EMA_satisfied	ordinal (range 1-7)	I feel satisfied.
EMA_insecure	ordinal (range 1-7)	I feel insecure.
EMA_cheerful	ordinal (range 1-7)	I feel cheerful
EMA_annoyed	ordinal (range 1-7)	I feel annoyed.
EMA_enthusiastisch	ordinal (range 1-7)	I feel enthusiastic.
EMA_lonely	ordinal (range 1-7)	I feel lonely.
EMA_guilty	ordinal (range 1-7)	I feel guilty.
EMA_selfdoubt	ordinal (range 1-7)	I doubt myself.
EMA_disappointed	ordinal (range 1-7)	I feel disappointed about myself.
EMA_likemyself	ordinal (range 1-7)	I like myself.
EMA_safe	ordinal (range 1-7)	I feel safe.
EMA_benevolent	ordinal (range 1-7)	I feel benevolent.

3.1.1 Discrete Time Steps

As mentioned, the EMA notifications were sent out at random (continuous) time points during the allotted daily interval. The generative model expects discrete equidistant time steps, which creates the challenge of finding a sensible discretization in time.

As can be seen in Figure 5, the trigger times for EMA prompts are roughly equidistant during the day, while the larger time differences reflect the necessity of sleep. Based on this, I inserted empty values (filled with 'NaN's) in between EMAs that were more

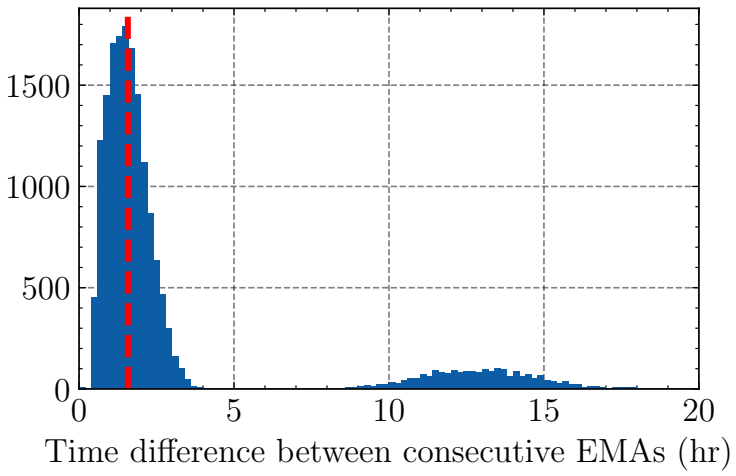


Figure 5: The histogram shows that EMA notifications are roughly equidistant in the data ($\sim$ 1.6 hours). The smaller peak between 10 and 15 hours corresponds to the night phases.

than three hours apart. By doing so, one day is roughly partitioned into 16 time steps that are 1.5 hours apart. I did not aggregate time steps that were closer than 1.5 hours to avoid losing any information from the already sparse data. Figure 6 shows an example time series after preprocessing.

This straightforward approach is a reasonable choice here, but it is important to mention that integrating features with varying time scales is a complicated challenge. When working with multimodal data, such as sensor data and EMA questionnaires, the sampling rates might wildly differ and a simple discretization in time might not allow the model to capture all the temporal dependencies in the data. For instance, step counts measured via the smart phone are obviously far more numerous than data points that require user input. Simply aggregating more frequently sampled data might be problematic, as it carries the danger of losing important dependencies. There exist multiple different avenues to extend state space models for multi-rate data [Li and Marlin 2020]. For instance, leveraging a hierarchical latent structure has been proposed as a way to learn on differently sampled data [Che et al. 2018a]. It might also prove worthwhile to use a continuous model formulation that does not require discrete time steps and is therefore closer to the real-world dynamics [Chen et al. 2018; Rubanova et al. 2019; Monfared and Durstewitz 2020b].

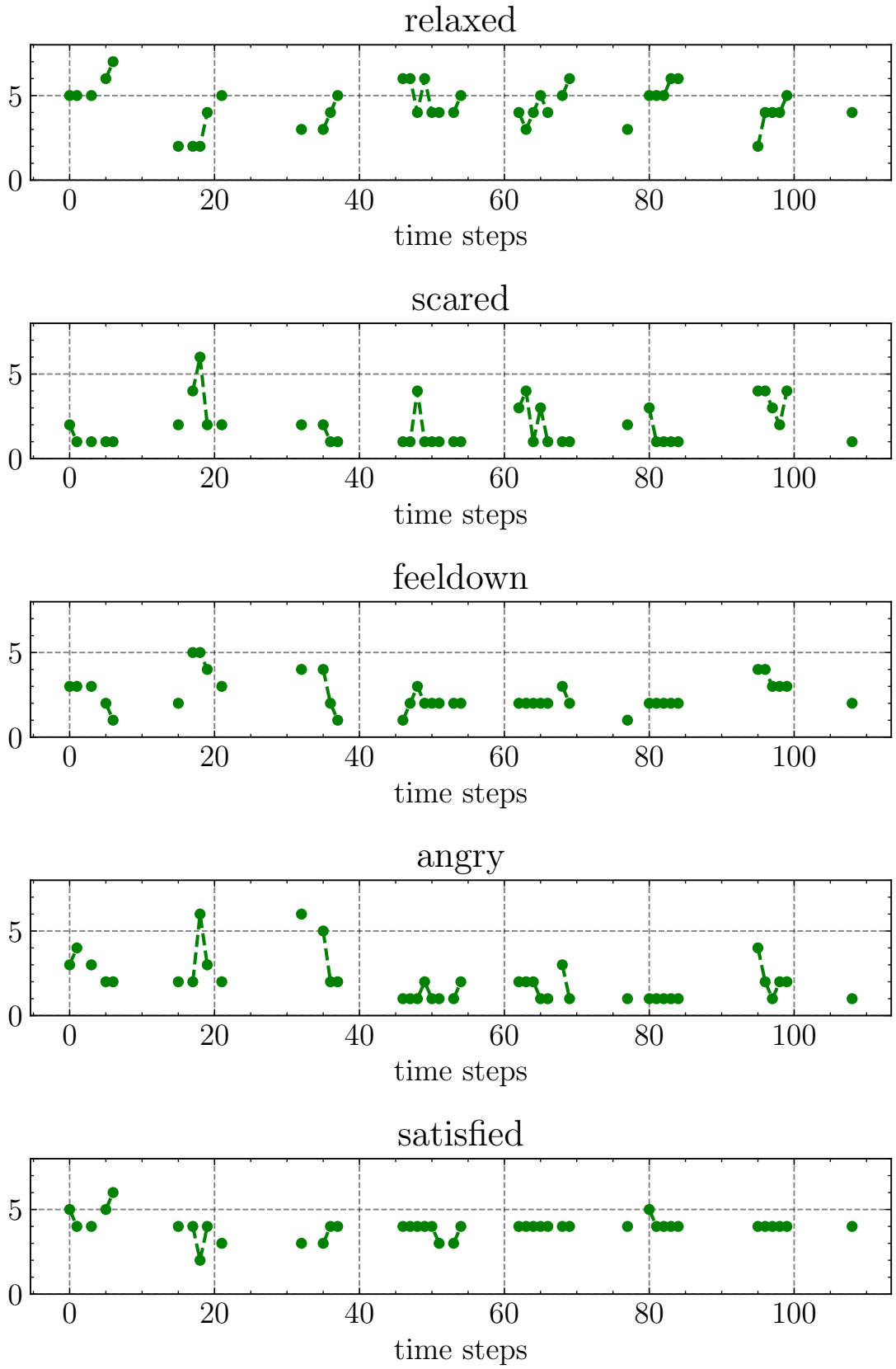
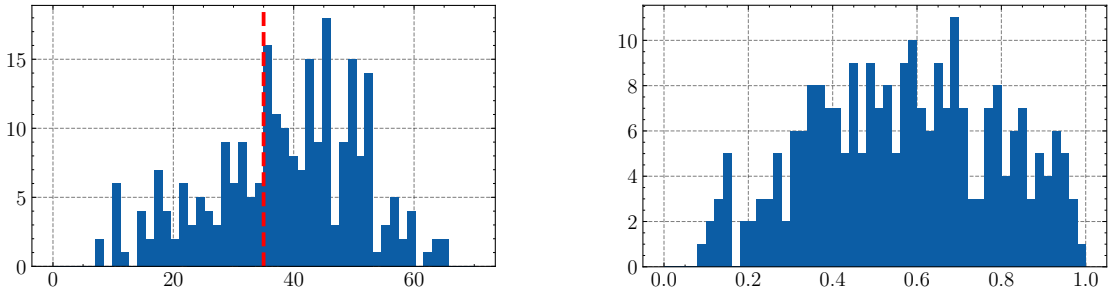


Figure 6: A one week EMA time series from the EMIcompass dataset depicting a subset of Likert-scale features. One time step roughly corresponds to 1.5 hours. Missing values were inserted to account for the night phases.

3.2 Missing Values

The dataset is comprised out of three time series (baseline, post-treatment, follow-up) for each of the 83 participants. Unsurprisingly, we find many missing values in the data, either because the participants did not react to an input prompt, or due to the time discretization described above. On average users fully completed around 55% of EMAs with quite a bit of variation across time series, see Figure 7. I chose to exclude time series with less than 35 non-missing time steps from further investigation, which leaves us with 156 time series that on average contain 47 non-missing time steps. It also important to highlight that there are no partially missing time steps, meaning either no feature is missing or all are at a specific time point. This is due to how the questionnaire is setup; it can only be sent off once the participant responded to all items.



(a) Number of non-missing time steps

(b) Compliance rate

Figure 7: Each time series roughly covers a one-week period with eight EMA prompts per day. The left histogram shows the number of non-missing EMAs for different time series, while the right histogram shows the percentage of questionnaires that participants completed.

All time series contain at least around 50% missing time steps that need to be sensibly integrated into the model framework. If we recall the SGVB estimate of the ELBO from the previous Chapter (see 2.70),

$$\begin{aligned} \text{ELBO}_{\mathbf{x}}(\boldsymbol{\theta}, \boldsymbol{\phi}) \approx & \frac{1}{L} \sum_{l=1}^{L} \log p_{\theta}(\mathbf{x}, g_{\boldsymbol{\phi}}(\mathbf{x}, \boldsymbol{\epsilon}^{(l)}) \\ & + \frac{TM}{2}(\log(2\pi) + 1) + \frac{1}{2} \log |\boldsymbol{\Sigma}_{\boldsymbol{\phi}}(\mathbf{x})| \end{aligned} \tag{3.1}$$

we can see that the observations are needed for calculating the likelihood of the generative model and for determining the parameters of the approximate posterior.

The likelihood of the observation model simply factorizes in time, which makes it possible to evaluate it only on the measured values by simply restricting the sum to the time points $t^{\text{obs}} \in \mathcal{O}_t = \{t \mid \mathbf{x}_t \text{ is observed}\}$ associated with non-missing data points [Fortuin et al. 2020; Nazábal et al. 2020].

$$\log p_{\theta}(\mathbf{x}^{\text{obs}}|\mathbf{z}) = \sum_{t^{\text{obs}} \in \mathcal{O}_t} \log p_{\theta}(\mathbf{x}_t|\mathbf{z}_t) \tag{3.2}$$

3.2.1 Imputation

Without making significant changes to the model architecture, the missing values need to be replaced before they get inputted into the encoder model. Thus, we need to choose a suitable imputation strategy. In principle, missing values from multivariate time series can be estimated either by making use of the temporal correlations in each feature, or exploiting the correlations across different variables [Fortuin et al. 2020]. Here, the feature correlations are not of much help, as time steps are always completely missing. This is also a general challenge in clinical research, as participants that do not self report on one item, tend to do the same for other variables [Pedersen et al. 2017]. A variety of imputation methods that are geared towards univariate imputation exist [Moritz et al. 2015]. For instance, very simple single value approaches can be used, such as replacing missing values by the overall temporal mean, or forward imputing by replacing missing values by the last known measurement.

I primarily relied on a weighted moving average to separately estimate the missing values in each category, see Figure 8. The missing data points are replaced by the weighted mean of the $2k$ neighboring values (k values before and after the time point). Thus, the estimated value of the feature i at time point t can be written as:

$$\tilde{x}_t^{(i)} = \frac{\sum_{j=-k}^{k} w_j x_{t+j}^{(i)}}{\sum_{j=-k}^{k} w_j} \tag{3.3}$$

The kernel weights w_j are chosen from a Gaussian distribution ($\sigma = 4$) centered around t, which leads to temporally adjacent measurements contributing more to the average. If many values are missing close to the time point, observations further away become more important. The window size k is set to 40; its size does not

matter much, as long as it is big enough, since far away values are automatically suppressed by the Gaussian weighting.

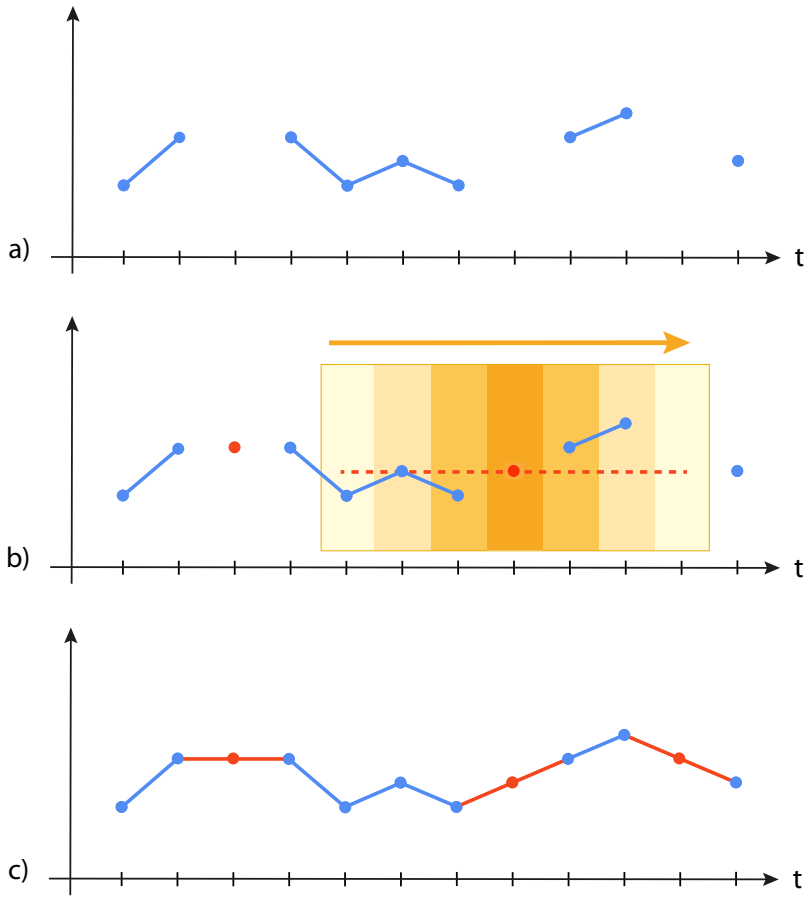


Figure 8: The missing observations are imputed through a weighted moving average of neighboring measurements with the weighting factors decreasing according to a Gaussian distribution.

As an alternative, I also explored an imputation strategy that makes direct use of the generative model. The general idea is quite intuitive, as we are already training a probabilistic model that should at least in theory be capable of generating new observations. More specifically, I draw an initial latent state at all non-missing time steps that are followed by a missing one. From these initial states, the PLRNN can be propagated forward until the next observed value. The resulting latent states

can then be used as input for the observation model, which allows us to generate replacement values at the missing time steps. The newly estimated observations together with the original measured values can then be used as input for the encoder to find the parameters of the approximate posterior, which in turn makes it possible to sample latent states and calculate the entropy term.
In theory, this method should be suited for finding reasonable estimates for the missing data. In fact, latent variable methods have oftentimes been specifically used for data imputation tasks [Che et al. 2018a; Fortuin et al. 2020]. A difficulty arising with the CNN encoder design is that the sampling of the initial latent states is not possible if too many values are missing. Since the CNN encoder always takes into account neighboring observations $\{\mathbf{x}_{t-2k}, \ldots, \mathbf{x}_t\}$ to generate a sample $\mathbf{z}_t$, we run into a problem if not enough observations are present before the next missing value. This issue can only be somewhat alleviated by reducing the kernel size, which generally might not be desirable. A potential solution might be too generate a full latent trajectory from $\mathbf{z}_0$, although this would likely be too inaccurate for large datasets.
Initial results showed that both imputation methods lead to very similar model performance with the CNN encoder. Since the model algorithm with PLRNN imputation takes much longer to train, I opted for using the moving average for the empirical investigation in this thesis. Still I think it is prudent to further explore this issue in future work, especially designing and testing an encoder that is capable of directly handling missing values. For instance, an encoder could be used that does not take into account temporally adjacent values, and therefore only requires $\mathbf{x}_t$ to draw $\mathbf{z}_t^{(l)}$. Furthermore, it could be very important to distinguish between types of missing values, as variables missing due to lack of user input and missing values in the night phases are fundamentally different. It might even be useful to consider a two stage imputation, where the generative model is used for imputation in the day phases, and a mean for the night periods, where we have less information about the dynamics.

3.2.2 Informative Missingness

Oftentimes, the presence of missing values does not necessarily only correspond to a lack of information, but can also be informative and provide meaningful insight on the question at hand [Rubin 1976; Che et al. 2018b; Little and Rubin 2020]. If that is the case, fundamentally depends on the underlying mechanism that leads to missing measurements [Rubin 1976]. Here, it is sensible to assume that missing

data points do not completely occur at random, but are dependent on the values of other features, the true value of the unobserved variable and the general point in time where the observation occurred. For instance, it is easy to imagine that participants might have a tendency to report more or less at certain hours or weekdays, or that a patient going through a depressive phase might be less inclined to fill out a questionnaire on his emotional state. This in turn allows us to potentially exploit the information inherent in missing values, as their occurrence is likely indicative of the underlying emotional dynamics.
In the course of this thesis, informative missingness was not yet implemented into the model framework. Still, I will give a brief outlook, how this could potentially be approached in future work. We can introduce an indicator variable $m_t \in \{0,1\}$ that describes if a time point is missing or not [Che et al. 2018b; Little and Rubin 2020].

$$m_t = \begin{cases} 1, & \text{if } \mathbf{x}_t \text{ is observed} \\ 0, & \text{otherwise} \end{cases} \tag{3.4}$$

Notice, that we could simply extend this to an indicator vector $\mathbf{m}_t \in \{0,1\}^N$, if partially observed time points would become more widespread in the future, e.g. due to the introduction of sensor modalities. The masking time series $\{m_t\}$ can then be used as an additional feature to train the model on. This requires the introduction of a new observation model that describes the Bernoulli probability of a time step missing.

$$p(m_t = 1|\mathbf{z}_t) = p = 1 - p(m_t = 0|\mathbf{z}_t) \tag{3.5}$$

The probability p can then be parameterized via the canonical logit link function, which leads to something akin a logistic regression model [Nazábal et al. 2020; Bommer et al. 2021].

$$p(m_t = 1|\mathbf{z}_t) = p = \frac{1}{1 + e^{-(\beta_0 + \beta^T \mathbf{z}_t)}} \tag{3.6}$$

It might be beneficial to include additional information on the missing values into the model framework. For instance, by expanding the observation model to a multicategorical distribution, we could indicate different types of missingness, e.g. a third

categorical label could mean a value is missing due to the participant sleeping. Furthermore, we could construct other features, such as how much time passed since the last missing value (as seen in Che et al. 2018b), although we might hope that the model can also learn to construct expressive representations on its own.

3.3 Modeling Ordinal Data

All the features selected for model training were recorded on a Likert scale. As can be seen in Figure 9, many of these clearly violate the commonly used distributional assumption of normality. Especially the questions regarding negative affects lead to strongly skewed and zero-inflated responses. This is not necessarily surprising as depending on how and to whom the question is posed, we would expect the responses to shift, although it is interesting to see that positive emotions do not exhibit the same skewness. Additionally, Likert scales are ordinal, which can lead to systematic errors, if modeled as metric [Liddell and Kruschke 2018].
This means that we need to adapt the model correspondingly to provide a better description of the data at hand. In variational autoencoders, this is generally quite straightforward, as we can simply switch out the observation model $p_{\theta}(\mathbf{z}|\mathbf{x})$ for the most sensible distribution. The optimization criterium does not change much, since one only needs to replace the likelihood of the observation model in the ELBO. This makes the variational autoencoder a very flexible and powerful modeling framework for dealing with the specific distributional assumptions of varying data types.
In this section, I will first introduce the multi-categorical observation model (following Bommer et al. 2021), as a first attempt to model ordinal data. An implementation of a categorical observation model is also useful for future studies, as questionnaires commonly include categorical response variables. We will then focus on how to specifically model ordinal data by using the ordered-probit (and logit) model [McCullagh 1980].

3.3.1 Categorical Observation Model

A categorical or nominal variable can take on a discrete and fixed number of values. The different categories are defined by some kind of qualitative property, meaning that no ordering exists between them. Random categorical variables are described by the categorical probability distribution that is simply given by the probability values $\{p_1, \dots, p_K\}$ of each of the K categories with the constraint $\sum_{k=1}^{K} p_k = 1$. For

instance, each individual Likert item can take on one of seven categorical responses.

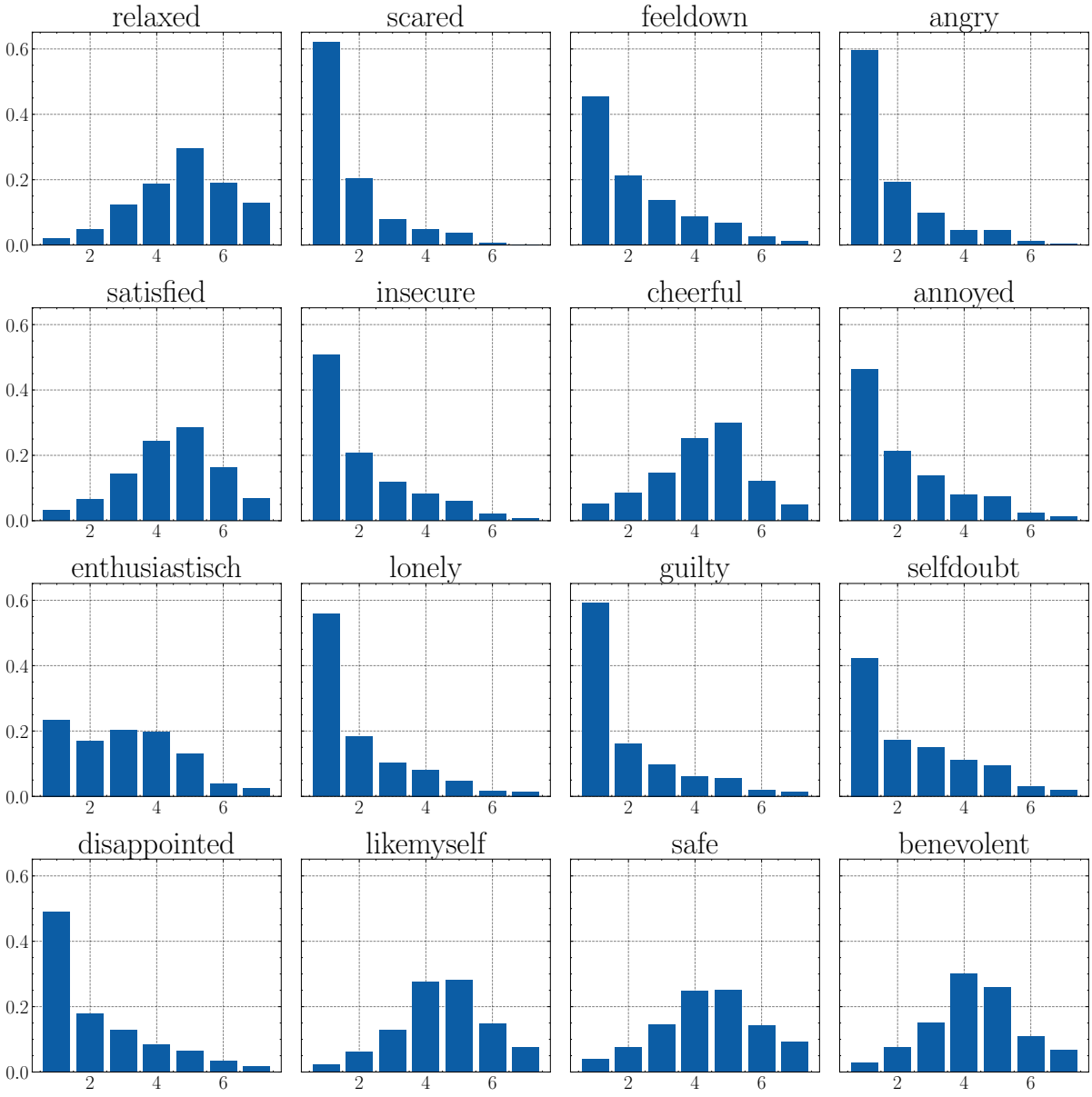


Figure 9: The histograms show the overall distribution of the different Likert items in the EMIcompass dataset. Features associated with negative affects are more zero-inflated, while positive emotions are more Gaussian-like distributed.

In our case, we need to consider N features indexed by i that are measured at different time steps t. Each potential integer value of the observations $x_{ti} \in \{1, \ldots, K\}$ corresponds to one of K categories. Which exact integer label $\{1, \ldots, K\}$ is assigned to each category is of course arbitrary, but using them allows for a more convenient

notation. An alternative representation can be achieved by using indicator vectors $\mathbf{c}_{ti} \in \{0,1\}^{K\times 1}$ for which exactly one element corresponding to the observed category takes on the value 1 and the rest are 0. The measured values are distributed according to the following multi-categorical distribution.

$$p(\mathbf{x}|\mathbf{z}) = \prod_i^N \prod_t^T \prod_k^K p(x_{ti} = k|\mathbf{z}_t)^{[x_{ti}=k]} \tag{3.7}$$

The dependency on the latent states is then realized as a generalized linear model with the canonical logit link function, which ensures that the probabilities stay bounded in $[0,1]$. The relative log-odds of the different categories are therefore expressed as a linear combination of the regression parameters $\boldsymbol{\beta_{ik}}$ and the latent states $\mathbf{z}_t$ [Durstewitz 2017b; Bommer et al. 2021].

$$\log \frac{p(x_{ti} = k|\mathbf{z}_t)}{p(x_{ti} = K|\mathbf{z}_t)} = \boldsymbol{\beta}_{ik}^T \mathbf{z}_t \quad \forall k = 1,\dots,K-1 \tag{3.8}$$

The latent vectors are expanded $\mathbf{z}_t \in \mathbb{R}^{M+1}$ by a leading column of ones to account for an offset term. In total, we have one parameter vector $\boldsymbol{\beta}_{ik} \in \mathbb{R}^{(M+1)}$ for each scale feature i and category $k = 1\dots K-1$. Finally, by respecting the constraint $\sum_{k=1}^K p(x_{ti} = k|\mathbf{z}_t) = 1$, we can invert the link function and arrive at the categorical probabilities.

$$\begin{aligned} p(x_{ti} = k|\mathbf{z}_t) &= \frac{e^{\boldsymbol{\beta}_{ik}^T \mathbf{z}_t}}{1 + \sum_l^{K-1} e^{\boldsymbol{\beta}_{il}^T \mathbf{z}_t}} \text{ for } k = 1\dots K-1 \\ p(x_{ti} = K|\mathbf{z}_t) &= \frac{1}{1 + \sum_l^{K-1} e^{\boldsymbol{\beta}_{il}^T \mathbf{z}_t}} \end{aligned} \tag{3.9}$$

To calculate the log-likelihood of the observation model, we can insert the probabilities in the categorical distribution above.

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) = \sum_{i}^{N}\sum_{t}^{T}\sum_{k}^{K}[x_{ti}=k]\log p(x_{ti}=k|\mathbf{z}_t) \tag{3.10}$$

$$= \sum_{i}^{N}\sum_{t}^{T}\sum_{k}^{K}[x_{ti}=k]\left([x_{ti}<K]\boldsymbol{\beta}_{ik}^{T}\mathbf{z}_t - \log\left(1+\sum_{l}^{K-1}e^{\boldsymbol{\beta}_{il}^{T}\mathbf{z}_t}\right)\right) \tag{3.11}$$

The categorical observation model is quite costly when it comes to its number of parameters. Since every feature and category is effectively modeled separately, we need to optimize $N \times (K-1) \times (M+1)$ parameters. This comes at no surprise, since categorical data is only defined by its qualitative groupings. In the case of ordinal data, such as the Likert items, we do have information about the ordering of the different responses that should be taken into account when formulating the model. Using nominal classification on ordinal data is effectively throwing away information, which might not be permissible when working in small data environments [Gutiérrez et al. 2016].
My code is based on the implementation of the categorical observation model from Philine Bommer [Bommer et al. 2021], former master's student at DurstewitzLab. I improved upon it by vectorizing the operations for faster training and increasing its numerical stability.

3.3.2 Ordinal Variables

As previously mentioned, ordinal data is not associated with a metric space. While we do have a natural ordering between items, such as "strongly disagree" and "disagree", there exists no distance measure between categories, and it can not be guaranteed that the different response items are equidistant [Winship and Mare 1984; Liddell and Kruschke 2018]. The positive integer labels that are commonly assigned for the different ordinal responses only indicate the ordering of the values, but do not provide information about the spacing between points. One of the most widespread examples of ordinal data are Likert items, which measure the response of an individual to a question on a ordered-categorical scale [Likert 1932]. An aggregation of multiple Likert items is also referred to as a Likert scale.
Ordinal data is commonly assumed to being generated from a underlying continuous variable. This latent variable is segmented into contiguous intervals that represent the different categories of the ordinal scale in question [McCullagh 1980; O'Brien 1985]. For instance, an individuals emotional state, e.g. feeling of happiness, is

likely best described as a continuous variable. When answering a questionnaire, the participant is then forced to report on a discrete scale. This quantization process connecting internal representation to the measurement scale might take on many different forms, e.g. it has been suggested that humans tend to perceive many different types of stimuli on a logarithmic scale [Sun et al. 2012; Varshney and Sun 2013]. In general, a variety of errors can occur if ordinal observations are analyzed as if they were metric interval-level data [Liddell and Kruschke 2018], although arguments have been made that this is commonly less of an issue [Norman 2010]. To a certain extent this question is also strongly related to the domain of interest, and if the study questions can be designed in such a way that it can be guaranteed that the distances between responses are at least roughly equivalent.
Let us denote the unobserved continuous variable as x^*_{ti} that is divided into intervals by the threshold parameters $\beta^0_{i1}, \ldots, \beta^0_{i(K-1)}$ that correspond to the different discrete responses that the observed ordinal variable x_{ti} can take on [Winship and Mare 1984].

$$x_{ti} = k \quad \text{if} \quad \beta^0_{i(k-1)} < x^*_{ti} \leq \beta^0_{ik} \tag{3.12}$$

The interval points are assumed to be ordered with $\beta^0_{i0} = -\infty$ and $\beta^0_{iK} = \infty$:

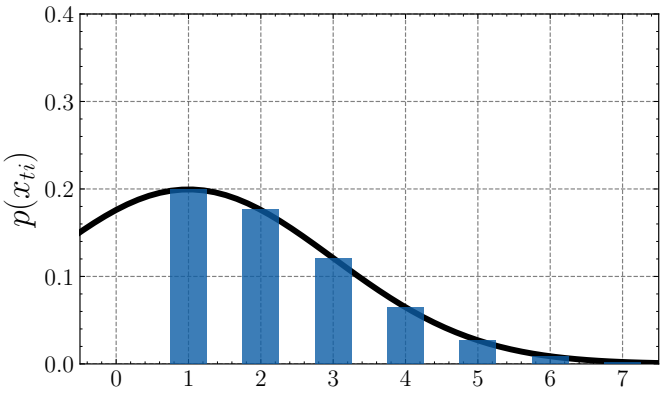
$$-\infty < \beta^0_{i1} < \beta^0_{i2} < \cdots < \beta^0_{i(K-1)} < \infty \tag{3.13}$$

If we assume that the latent variable is distributed according to some kind of probability density, we can then describe the probabilities of the different Likert responses as the cumulative probabilities between the respective cut-points of the latent distribution [Liddell and Kruschke 2018], as illustrated in Figure 10 .

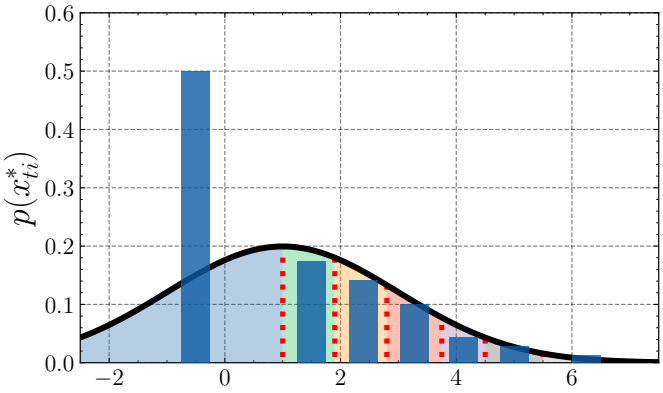
$$p(x_{ti} = k) = p(\beta^0_{i(k-1)} < x^*_{ti} \leq \beta^0_{ik}) = F_{x^*_{ti}}(\beta^0_{ik}) - F_{x^*_{ti}}(\beta^0_{i(k-1)}) \tag{3.14}$$

3.3.3 Ordinal Regression Models

The dependency on the latent states $\mathbf{z}_t$ of the generative model can again be expressed as a generalized linear model. The underlying continuous variable x^*_{ti} is



(a) Gaussian observation model



(b) Ordered-probit model

Figure 10: In the upper plot, a Gaussian observation model for ordinal data is depicted. There, we assume that the numerical integer values associated with the different Likert response items can be mapped onto an interval scale. The lower diagram, shows the distribution of the underlying latent variable x^*_{ti} segmented into intervals by the threshold parameters. The cumulative probabilities associated with the intervals under the curve correspond to the probabilities $p(x_{ti} = k)$ presented in the bar plot.

assumed to be a linear function of the latent states $\mathbf{z}_t$ and the model parameters $\boldsymbol{\beta}_i^T \in \mathbb{R}^M$. ϵ_{ti} is an independently distributed noise term with zero expectation $\mathbb{E}[\epsilon_{ti}] = 0$.

$$x^*_{ti} = \boldsymbol{\beta}_i^T \mathbf{z}_t + \epsilon_{ti} \tag{3.15}$$

The cumulative probabilities of the ordinal response variable $\mathbf{x}_{ti}$ satisfy the following equation with $F_{\epsilon_{ti}}$ being the cumulative density function of the error term [McCullagh 1980; Winship and Mare 1984].

$$
\begin{aligned}
p(x_{ti} \leq k|\mathbf{z}_t) &= p(x^*_{ti} \leq \beta^0_{ik}) \\
&= p(\boldsymbol{\beta}_i^T \mathbf{z}_t + \epsilon_{ti} \leq \beta^0_{ik}) \\
&= p(\epsilon_{ti} \leq \beta^0_{ik} - \boldsymbol{\beta}_i^T \mathbf{z}_t) \\
&= F_{\epsilon_{ti}}(\beta^0_{ik} - \boldsymbol{\beta}_i^T \mathbf{z}_t)
\end{aligned} \tag{3.16}
$$

It now becomes evident that the distribution $F_{\epsilon_{ti}}$ takes the role of an inverse link function that we denote as g^{-1}. Therefore, the distributional assumption we make for the noise ϵ_{ti} determines the exact form of the generalized linear model. The two most common choices are either the logistic or the normal distribution that respectively lead to the ordered logit and ordered probit model [Winship and Mare 1984], although a variety of other distributions can be considered as well, e.g. the proportional hazards model rooted in survival analysis [McCullagh 1980; Gutiérrez et al. 2016].

The ordered logit model is very similar to the multiple logistic regression model with the difference being that the cumulative probabilities $p(x_{ti} \leq k|\mathbf{z}_t)$ are parameterized instead of the probabilities $p(x_{ti} = k)$.

$$
g(p(x_{ti} \leq k|\mathbf{z}_t)) = \log \frac{p(x_{ti} \leq k|\mathbf{z}_t)}{1 - p(x_{ti} \leq k|\mathbf{z}_t)} = \beta^0_{ik} - \boldsymbol{\beta}_i^T \mathbf{z}_t \tag{3.17}
$$

The model is also called the proportional odds model [McCullagh 1980], since the odds $\kappa(x_{ti} \leq k|\mathbf{z}_t)$ of $x_{ti} \leq k$ can be expressed as:

$$
\kappa(x_{ti} \leq k|\mathbf{z}_t) = \frac{p(x_{ti} \leq k|\mathbf{z}_t)}{1 - p(x_{ti} \leq k|\mathbf{z}_t)} = \exp(\beta^0_{ik} - \boldsymbol{\beta}_i^T \mathbf{z}_t) \tag{3.18}
$$

If we now consider the ratio of odds for different covariate values $\mathbf{z}_t$ and $\mathbf{z}'_t$

$$
\frac{\kappa(x_{ti} \leq k|\mathbf{z}_t)}{\kappa(x_{ti} \leq k|\mathbf{z}'_t)} = \exp(-\boldsymbol{\beta}_i^T(\mathbf{z}_t - \mathbf{z}'_t)) \tag{3.19}
$$

we see that the ratio only depends on the difference between covariate values, and is independent of the category k in question. In other words, the odds of all ordinal response categories k change in the same proportionate way for different $\mathbf{z}_t$.

Finally, we can invert the link function to arrive at an expression for the cumulative probabilities.

$$p(x_{ti} \leq k|\mathbf{z}_t) = \frac{\exp(\beta_{ik}^0 - \boldsymbol{\beta}_i^T \mathbf{z}_t)}{1 + \exp(\beta_{ik}^0 - \boldsymbol{\beta}_i^T \mathbf{z}_t)} \tag{3.20}$$

As discussed, the alternative ordered probit model simply results from using the inverse standard normal distribution Φ^{-1} as a link function [Aitchison and Silvey 1957].

$$p(x_{ti} \leq k|\mathbf{z}_t) = \Phi(\beta_{ik}^0 - \beta_i^T z_t) \tag{3.21}$$

I implemented both the ordered probit and logit model, and finally chose to primarily work with the ordered logit model, as its computation is slightly more straightforward. In practice, both models usually lead to very similar empirical results [Rodrígues 2007], as the logistic and normal distribution are not that different. My experience when testing both models corroborated this assessment.
During optimization, it is not always necessary to separately enforce the ordering of the threshold parameters $\beta_{i(k-1)}^0 < \beta_{ik}^0$ [Greene and Hensher 2010; Christensen 2018], e.g. sometimes the threshold parameters stay ordered after careful initialization. This was not the case here, where the ordering of the parameters usually broke down only after a couple of epochs. Thus, I settled on using the following reparameterization, as suggested in Greene and Hensher 2010, to guarantee the non-decreasing nature of the threshold parameters.

$$\beta_{ik}^0 = \beta_{i(k-1)}^0 + e^{\alpha_{ik}^0} = \alpha_{i1}^0 + \sum_{l=2}^{k} \exp \alpha_{il}^0 \text{ with } \beta_{i1}^0 = \alpha_{i1}^0 \tag{3.22}$$

Finally, regardless of the chosen model, we can simply calculate the probabilities $p(x_{ti} = k|\mathbf{z}_t)$ from the cumulative probabilities,

$$p(x_{ti} = k|\mathbf{z}_t) = p(x_{ti} \leq k|\mathbf{z}_t) - p(x_{ti} \leq k-1|\mathbf{z}_t) \tag{3.23}$$

which then can be inserted into the likelihood term.

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) = \sum_{i}^{N}\sum_{t}^{T}\sum_{k}^{K}[x_{ti} = k]\log p(x_{ti} = k|\mathbf{z}_t) \tag{3.24}$$

The model could be easily expanded by choosing a nonlinear function, e.g. a neural network, instead of the linear model in Equation 3.15 [Mathieson 1996; Nazábal et al. 2020]. Although this might greatly increase the models flexibility [Gutiérrez et al. 2016], I opted for using a linear model as I did not want to excessively inflate the number of observation model parameters. A too expressive observation model might increase the danger of overfitting, especially when working with small data, and might in turn make it harder for the latent model to properly learn the temporal dynamics.

3.4 Hierarchical Parameter Estimation

A central challenge when working with many deep learning architectures is the sheer amount of data required to train them. For instance, state of the art image classification [Dai et al. 2021] or language models [Brown et al. 2020] require millions of parameters and data points to be successfully trained. In contrast to that, data in many scientific disciplines, e.g. materials science [Zhang and Ling 2018], biomedical engineering [Shaikhina et al. 2015] or psychiatry [Cearns et al. 2019; Durstewitz et al. 2019; Koppe et al. 2021] is generally much less abundant and more costly to gather. Especially when working with self-reported data, there are always practical limits to how often participants are willing to answer a survey [Wen et al. 2017]. Additionally, it is prudent to consider that high sampling frequencies might also put a mental burden on the participant [Stone et al. 2007] that could in turn decrease the potential benefits of the intervention framework. Therefore, to reach the goal of individualized predictions and interventions, it is necessary to develop methods that are capable of dealing with smaller sample sizes [Koppe et al. 2021].
In Section 2.3.7, we already discussed the possibility of integrating additional data modalities into the model, e.g. sensor data directly collected from a participants smart phone. These types of measurements can obviously be collected at a much higher frequency, and might help to somewhat alleviate the small data problem, although the combined modeling of differently sampled data is a non-trivial challenge in itself.
Here, I will discuss an alternative approach inspired by transfer learning [Pan and

Yang 2010; Weiss et al. 2016]. The general idea behind transfer learning is to leverage knowledge from another closely related problem domain to the question at hand, which allows models to be trained with even a fairly limited amount of data. More specifically, this is oftentimes realized by using model parameters that were pre-trained on a larger source dataset to initialize a new model and fine-tune its parameters on the smaller dataset in question. The key assumption is that both datasets are sufficiently statistically related [Weiss et al. 2016], such that the model can already extract some common general features and statistical properties from the source dataset. In a psychiatric context this could mean attempting to exploit information from a group of participants to strengthen the quality of the individual subject-level predictions [Durstewitz et al. 2019; Koppe et al. 2021]. In such a way, predictions could hopefully be made more robust [Durstewitz et al. 2019], while at the same time still allowing for personalized models that can capture the uniqueness of each patient or participant [Chekroud et al. 2017]. For larger study cohorts subjects will tend to be more diverse and exhibit less homogeneous characteristics, e.g. because they were recruited from a larger geographic range. In these cases, it might be useful to group participants into clusters based of some kind of measure of similarity before training the models [Cearns et al. 2019].

As a first step towards this goal, I implemented a form of hierarchical parameter estimation that allows the model to be jointly trained on time series of different participants. The general idea is to infer a subset of the model parameters $\boldsymbol{\theta}_{\text{group}}$ at the group-level, while fine-tuning the rest of the parameters $\boldsymbol{\theta}_{\text{subj}}^{(j)}$ individually for each subject $j = 1, \ldots, N_{\text{subj}}$. I chose to train the parameters of the basis expansion and the observation model at a subject level, leaving the rest of the generative model parameters to be inferred over all participants. Additionally, the parameters $\boldsymbol{\phi}$ of the recognition model are also shared.

$$\begin{aligned} \boldsymbol{\theta}_{\text{group}} &= \{\mathbf{A}, \mathbf{W}, \mathbf{h}, \boldsymbol{\Sigma}\} \\ \boldsymbol{\theta}_{\text{subj}}^{(j)} &= \left\{\boldsymbol{\theta}_{\text{obs}}^{(j)}, \{\alpha_b^{(j)}, \mathbf{h}_b^{(j)}\}\right\} \text{ for } j = 1, \ldots, N_{\text{subj}} \end{aligned} \tag{3.25}$$

In future studies, one could test many alternative combinations of group and subject-level parameters. Additionally, it might be fruitful to introduce new expressive parameters that are especially suitable for multi-level parameter inference. Finding these parameters is not a very straightforward task, but could potentially be inspired by domain knowledge, e.g. parameters that are directly interpretable. In the same

Algorithm 1: Hierarchical parameter estimation

input : subject time series $\mathbf{x}^{(j)}$ $j = 1, \ldots, N_{\text{subj}}$
output: model parameters $\boldsymbol{\theta} = \{\boldsymbol{\theta}_{\text{group}}, \boldsymbol{\theta}_{\text{subj}}\}$, $\boldsymbol{\phi}$

$\boldsymbol{\theta}, \boldsymbol{\phi} \leftarrow$ Initialized randomly;
for $i \leftarrow 0$ **to** N_{epochs} **do**
  Partition $\left\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N_{\text{subj}})}\right\}$ into L mini-batches $b_l = \left\{\mathbf{x}^{(j_1)}, \ldots, \mathbf{x}^{(j_{M_b})}\right\}$;
  for $b_l \in \{b_1, \ldots, b_L\}$ **do**
    for $\mathbf{x}^{(j)} \in b_l = \left\{\mathbf{x}^{(j_1)}, \ldots, \mathbf{x}^{(j_{M_b})}\right\}$ **do**
      $\boldsymbol{\epsilon}^{(l)} \sim p(\boldsymbol{\epsilon})$;
      $\tilde{\mathcal{L}}_{\mathbf{x}^{(j)}}(\boldsymbol{\theta}_{\text{group}}, \boldsymbol{\theta}^{(j)}_{\text{subj}}, \boldsymbol{\phi}) \leftarrow \mathbf{x}^{(j)}, \boldsymbol{\epsilon}^{(l)}, \boldsymbol{\theta}_{\text{group}}, \boldsymbol{\theta}^{(j)}_{\text{subj}}, \boldsymbol{\phi}$;
    end
    $\boldsymbol{g}_{b_l} \leftarrow \nabla_{\boldsymbol{\theta},\boldsymbol{\phi}} \frac{1}{M_b} \sum_{\mathbf{x}^{(j)} \in b_l} \tilde{\mathcal{L}}_{\mathbf{x}^{(j)}}(\boldsymbol{\theta}_{\text{group}}, \boldsymbol{\theta}^{(j)}_{\text{subj}}, \boldsymbol{\phi})$;
    Update the parameters $\boldsymbol{\theta}, \boldsymbol{\phi}$ with regard to the average gradients $\boldsymbol{g}_{b_l}$;
  end
end

spirit, one could attempt to find ways to initialize parameters through other prior information that might potentially be available on a subject, such as epigenetic risk factors [Keverne and Binder 2020].

As mentioned, the model is simultaneously trained on the time series of all participants. During each epoch the individual subject time series $\mathbf{x}^{(j)} = \{\mathbf{x}_t\}^{(j)}$ are randomly grouped into L different mini-batches $b_l = \left\{\mathbf{x}^{(j_1)}, \ldots, \mathbf{x}^{(j_{M_b})}\right\}$ of size M_b. Thus, the mini-batches $b_1, \ldots, b_L$ define a partitioning over the set of subject time series $\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N_{\text{subj}})}\}$ so that each time series $\mathbf{x}^{(j)}$ belongs to exactly one mini batch b_l. Notice that this implies that the time series in the same mini-batch can be of different length, although all time series in the EMIcompass study cover a very similar duration. If the number of time series is not divisible by the batch size M_b the last mini batch contains less elements. My implementation also allows for the subject time series to be split up into smaller sequences, which alternatively can be used to construct the batches. I used this feature very rarely as smaller sequences carry the danger of losing too much temporal structure, and the individual time series are already very small to begin with. We then iterate through all the mini-batches, and perform the gradient updates jointly for each mini-batch of observations. The gradients with respect to the group-level parameter need to be averaged over the different time series in the mini-batch, while the subject-level gradients only depend on the respective subject time series.

$$\nabla_{\boldsymbol{\theta}_{\text{group}}} \mathcal{L}_{b_l}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \nabla_{\boldsymbol{\theta}_{\text{group}}} \frac{1}{M_b} \sum_{\mathbf{x}^{(j)} \in b_l} \mathcal{L}_{\mathbf{x}^{(j)}}(\boldsymbol{\theta}_{\text{group}}, \boldsymbol{\theta}_{\text{subj}}^{(j)}, \boldsymbol{\phi}) \tag{3.26}$$

$$\nabla_{\boldsymbol{\theta}_{\text{subj}}^{(j)}} \mathcal{L}_{b_l}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \nabla_{\boldsymbol{\theta}_{\text{subj}}^{(j)}} \mathcal{L}_{\mathbf{x}^{(j)}}(\boldsymbol{\theta}_{\text{group}}, \boldsymbol{\theta}_{\text{subj}}^{(j)}, \boldsymbol{\phi}) \tag{3.27}$$

The epoch is concluded after iterating through all of the mini-batches and performing L parameter updates. Finally, we note that stochasticity is injected into the algorithm in two places; first through the random partitioning of the time series into mini-batches, and second by Monte Carlo sampling $\boldsymbol{\epsilon}^{(l)} \sim p(\boldsymbol{\epsilon})$ [Kingma and Welling 2019]. The hierarchical optimization procedure is summarized in Algorithm 1.

4 Empirical Investigation

In this section, I will present a preliminary empirical investigation of the model framework using the EMIcompass data set presented in Section 3.1, and then attempt to generate realistic benchmark data to further test the model.

4.1 Prediction Evaluation

4.1.1 Cross-Validation for Time Series

To accurately evaluate the quality of predictions we can expect in future application settings, we need to calculate an out-of-sample error. Using the same data for model training and evaluation creates the risk of severely overestimating the model capabilities by overfitting the training data [Hastie et al. 2009; Durstewitz 2017b]. This can be especially problematic for complex model architectures consisting of many parameters that are able to very accurately represent the training data, but in doing so overly adapt to the noise present in the data, leading to poor performance on new samples. On the other hand, many modern deep learning models apparently exhibit better generalization in the overparameterized regime, defying the conventional wisdom that overly expressive models typically exhibit low accuracy on test data [Belkin et al. 2019]. This phenomenon has been attributed to the effective usage of regularization techniques, but is still debated and much of the communities understanding of how model complexity and sample size impact generalization still seems to be developing [Nakkiran et al. 2021; Zhang et al. 2021].
In any case, it is difficult or impossible to theoretically evaluate the optimal model complexity or find an estimate for the minimum amount of data required to successfully train a model [Koppe et al. 2021]. Therefore, it is prudent to perform a careful empirical investigation to find how accurate the models predictions will be in practice. Cross-validation is one of the most widespread techniques for estimating the expected prediction error [Hastie et al. 2009]. The basic idea is to remove a subset of the training data before the model is fitted. After the model parameters are estimated it can then be used to test how the models predictions generalize for unseen data. To find a robust estimate of the generalization error and to make more efficient use of the data, the procedure is usually performed various times by partitioning the data set into k segments, and using each segment once as a test set, while fitting the model on the remaining data. The average of the k test set errors can then be used as a prediction error estimate.

However, when working with time series data simple cross-validation is problematic, as the i.i.d. assumption does not hold anymore due to temporally adjacent time points usually being highly dependent on each other [Bergmeir and Benítez 2012; Koppe et al. 2021]. In that case, it can not be assumed that training and test set are independent of each other, and one needs to be careful to respect the temporal dependency structure present in the data [Bergmeir and Benítez 2012]. Typically one selects a section at the end of the time series for an out-of-sample evaluation to ensure that only prior observations are used for the forecast, which also mimics the usual application setting [Tashman 2000]. We can then use the model to calculate ahead predictions $\hat{\mathbf{x}}_{k+1}, \hat{\mathbf{x}}_{k+2}, \dots$ from the last point of the training set $\mathbf{x}_k$, also called the forecast origin, and compare the results to the test data $\mathbf{x}_{k+1}, \dots, \mathbf{x}_T$.
A problem with this approach is that we can only calculate one forecast per time series, which might lead to the error estimate being highly dependent on the chosen forecast origin. This could be especially be problematic, if the data contains non-stationary behavior. Alternatively, different train-test splits can be used by sequentially moving the forecast origin and retraining the entire model on each training data set. This also sometimes involves removing values from the beginning of the time series to keep the length of the training set constant [Tashman 2000].
There exist a variety of other evaluation methods for time series data [Bergmeir and Benítez 2012], but I opted for using a small test set at the end of each time series. This choice stems from the fact that the empirical data available only contains a very small number of time steps (around 100), which made it unfeasible to significantly shorten the training data. For future studies covering longer time periods, I would argue for using a rolling forecast origin for evaluation. I also average the prediction error over the models of multiple participants, which should help in getting a more accurate estimate for the out-of sample predictions.
The error on the training set can be calculated in similar fashion, with the difference being that we can choose any time point that is not too close to the end of the training time series as forecast origin. This allows us to calculate an average error over multiple forecasts for each training set.

4.1.2 Ordinal Predictions

Predictions on the test set can be produced by propagating the latent model $p_{\boldsymbol{\theta}}(\mathbf{z}_t|\mathbf{z}_{t-1})$ forward. We do so by using the recognition model to estimate a latent state at the forecast origin $\hat{\mathbf{z}}_k = \mathbb{E}_{q_{\boldsymbol{\phi}}}[\mathbf{z}_k|\mathbf{x}_{1:k}]$ and then itertively calling the latent

step until we reach the time step we want to predict $\hat{\mathbf{z}}_{k+n}$. The final latent state is then inserted into the observation model $p(\mathbf{x}_{k+n}|\mathbf{z}_{k+n})$, from which we can generate a prediction $\hat{\mathbf{x}}_{k+n}$ by using the expected value $\mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})}[\mathbf{x}_{k+n}|\hat{\mathbf{z}}_{k+n}]$ of the distribution. In this thesis, I made the compromise to use the MSE for evaluating the predictions of the ordinal data, while also calculating the categorical precision to check if the MSE somehow distorts the results. This choice was primarily made to be able to quickly test the first model iterations. Of course, it would be ideal to build up an evaluation pipeline that also utilizes the ordinal character of the data, but this was not possible due to time constraints.

4.1.2.1 RMSE

For the EMIcompass data the n-step ahead prediction error is calculated for $n = \{1, 2, 3\}$ steps, while on the benchmark data it is possible to evaluate longer ahead predictions. The n-step RMSE for one time series j consisting of N Likert items i is then given by:

$$\text{RMSE}_n^{(j)} = \frac{1}{\sqrt{N}} \|\mathbf{x}_{k+n}^{(j)} - \hat{\mathbf{x}}_{k+n}^{(j)}\|_2 = \frac{1}{\sqrt{N}} \|\mathbf{x}_{k+n}^{(j)} - \mathop{\mathbb{E}}_{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})}[\mathbf{x}_{k+n}^{(j)}|\hat{\mathbf{z}}_{k+n}^{(j)}] \|_2 \tag{4.1}$$

If an observation is missing, the corresponding RMSE term is dropped. The RMSE values from the different time series models are then averaged over the entire group of participants, which gives us the following summary measure

$$\text{RMSE}_n^{\text{model}} = \frac{1}{N_{\text{subj}}} \sum_j^{N_{\text{subj}}} \text{RMSE}_n^{\text{model},(j)} = \frac{1}{N_{\text{subj}}} \sum_j^{N_{\text{subj}}} \frac{1}{\sqrt{N}} \|\mathbf{x}_{k+n}^{(j)} - \hat{\mathbf{x}}_{k+n}^{(j)}\|_2 \tag{4.2}$$

As a simple baseline measure, we use the mean of the training set as a constant forecast for the test set, and also determine its RMSE.

$$\text{RMSE}_n^{\text{mean}} = \frac{1}{N_{\text{subj}}} \sum_j^{N_{\text{subj}}} \text{RMSE}_n^{\text{mean},(j)} = \frac{1}{N_{\text{subj}}} \sum_j^{N_{\text{subj}}} \frac{1}{\sqrt{N}} \|\mathbf{x}_{k+n}^{(j)} - \frac{1}{k_j} \sum_{t=1}^{k_j} \mathbf{x}_t^{(j)}\|_2 \tag{4.3}$$

We then calculate the difference of both error measures and average over the participants $\text{RMSE}_n^{\text{diff}} = \frac{1}{N_{\text{subj}}} \sum_j^{N_{\text{subj}}} (\text{RMSE}_n^{\text{model},(j)} - \text{RMSE}_n^{\text{mean},(j)}) = (\text{RMSE}_n^{\text{model}} - \text{RMSE}_n^{\text{mean}})$

to gain a sense if the model is performing better than a simple predictor.

4.1.2.2 Confusion matrix and Precision

As it is common for categorical classification tasks, we can construct a confusion matrix C, where the entries C_{ij} count the number of observations i that were predicted to be in class j. Accordingly, we need to use the mode $\hat{x}_{ti} = \arg\max_k p(x_{ti} = k|\mathbf{z}_t)$ instead of the expected value for prediction. We can treat each of the $K = 7$ ordinal values of the N Likert items as a different class label, which gives us a $K \times K$ dimensional confusion matrix $C^{(i)}$ for each feature. Here, we will determine the confusion matrix for the single time point of the relevant n-step ahead prediction, but we could of course summarize the predictions over multiple time points.
We can then calculate an overall classification metric for the n-step ahead-prediction of a time series j by averaging over the entries of the different categories. This ensures that all observed values contribute the same, rather than weighting each class equally. The precision is then defined as the ratio of the number of correct predictions divided by the total number of predictions across all categories.

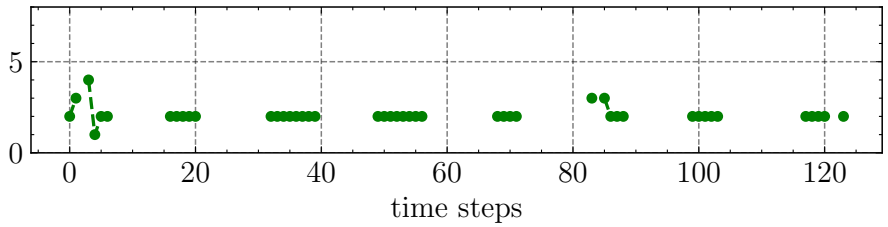
$$\text{Precision}^{(j)} = \frac{\sum_{i=1}^{N}\sum_{l=1}^{K} C_{ll}^{(i)}}{\sum_{i=1}^{N}\sum_{l=1}^{K}\sum_{m=1}^{K} C_{lm}^{(i)}} \tag{4.4}$$

Again, we ignore missing observations, and average the precision score over the entire group of participants.

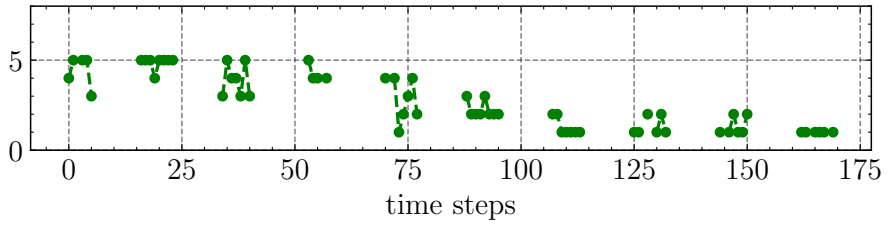
4.2 EMIcompass Data

As discussed in Section 3.1, I am using the data collected during the EMIcompass study to conduct the first empirical evaluations of the model. In total, the model is tested on 90 subject time series that at least contain 35 non-missing time points, and enough consecutive time points at the end of the time series so that a small test set can be constructed. For each time series a separate model is trained. By averaging the prediction error across the entire group, we then calculate a summary measure for the model's performance, as described in Section 4.1.2.
These time series are very challenging to train on, as they are very short, contain many missing values and can exhibit very erratic and noisy behavior. Additionally,

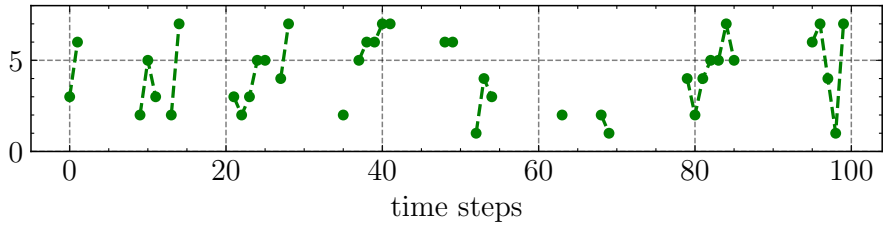
we observe seemingly non-stationary behavior in some of the time series that is difficult to distinguish from slow oscillations. The time series vary strongly for different participants and features, with some almost being constant in time, while others seemingly showing no regular pattern, see Figure 11.



(a) Nearly Constant



(b) Slow Time Scale



(c) Erratic Behavior

Figure 11: The EMIcompass dataset contains time series of varying time scales. Many of the time series are quite constant in time, while others exhibit more erratic or non-stationary behavior.

The different Likert items can be approximately grouped into two categories, depending on whether they are associated with a negative or a positive affect. The positive Likert items are more Gaussian-like distributed, while the negative affects are zero-inflated, as we saw in Figure 9. The features also exhibit a moderate amount of correlation, which can be determined by calculating the Spearman rank order correlation coefficient ρ between all the features i and j, as displayed in Figure 12.

$$\rho^{(ij)} = \frac{\mathrm{cov}(R(\mathbf{x}_i), R(\mathbf{x}_j))}{\sigma_{R(\mathbf{x}_i)} \sigma_{R(\mathbf{x}_j)}} \in [-1, 1] \tag{4.5}$$

$R(\mathbf{x}_i)$ denotes the ranks of all the ordinal observations for feature i. The correlation between the features is important to consider, as high correlation might indicate that some of the features are superfluous and do not add further information in training. Of course the question of feature selection is also tied to the application setting, e.g. what emotional attributes are important to predict and improve from a psychiatric standpoint.

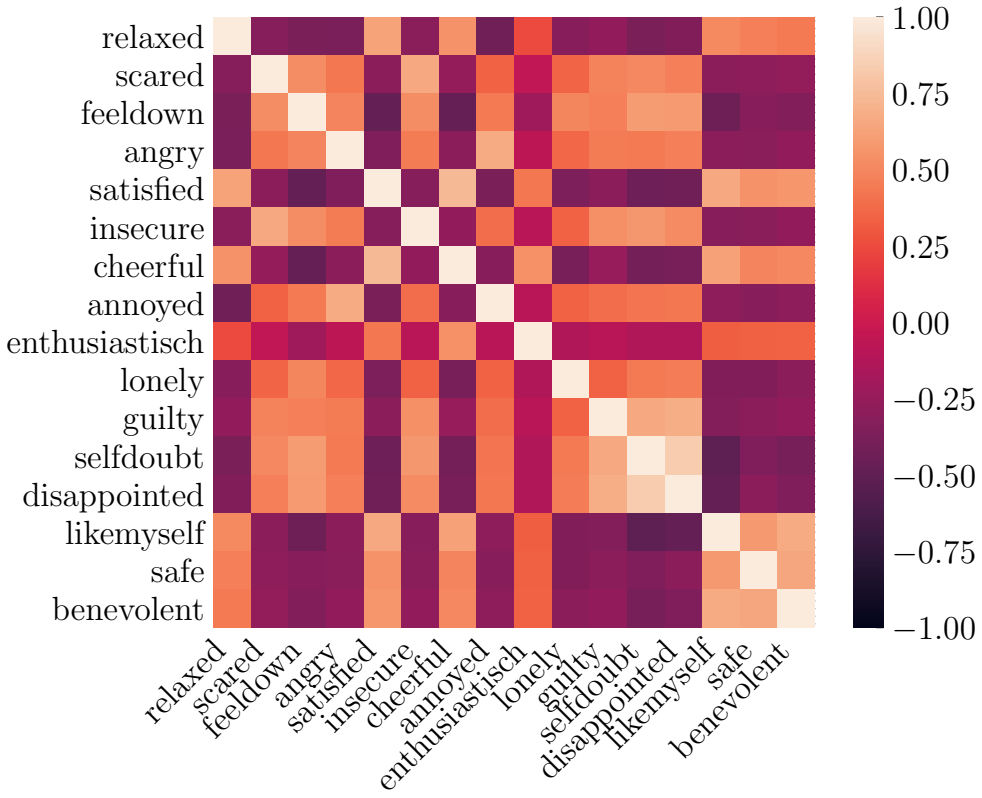
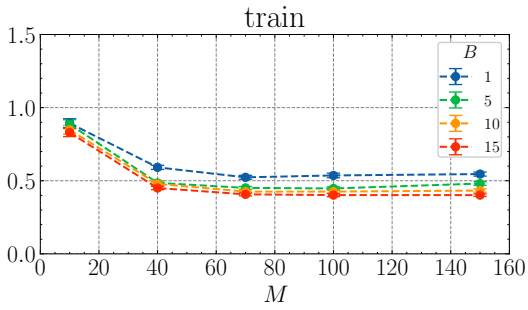
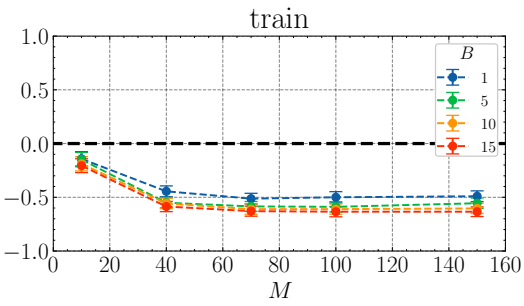


Figure 12: Likert items measuring positive attributes are positively correlated with each other, with the same holding true for negative emotions. Unsurprisingly, the Spearman correlation for negative and positive features is negative.

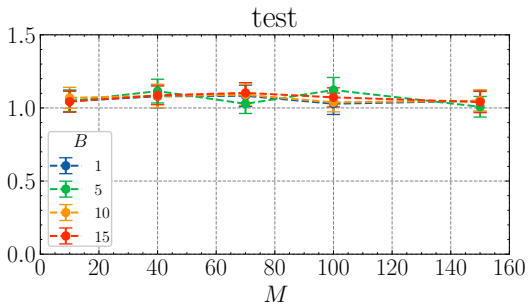
On the other hand, this should also be investigated empirically, as a specific subset of features might prove to be especially predictive to select adequate interventions. From a study design there also exists a trade off between the sampling frequency and the number of Likert items on the questionnaire. The more extensive and time consuming a single EMA becomes, the less often we can expect participants to be willing to answer it.



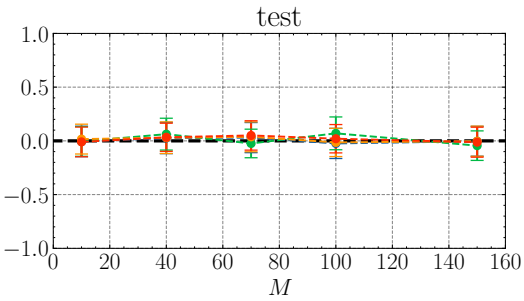
(a) $\text{RMSE}_{n=1}^{\text{model}}$



(b) $\text{RMSE}_{n=1}^{\text{model}} - \text{RMSE}_{n=1}^{\text{mean}}$



(c) $\text{RMSE}_{n=1}^{\text{model}}$



(d) $\text{RMSE}_{n=1}^{\text{model}} - \text{RMSE}_{n=1}^{\text{mean}}$

Figure 13: The upper row shows the 1-step ahead prediction error for the training set, while the lower row corresponds to the test set. The plots positioned on the left display the RMSE. The right figures show if the model is performing better than simply using the mean of the training set as forecast; if a value falls under the black line the model is more accurate. The error bars correspond to the standard error of the mean over the different participants.

4.2.1 Hyper-Parameter Search

All models are trained with the ordinal observation model described in Section 3.3.3. The CNN-encoder for the mean of the approximate posterior consists of four layers with the respective kernel sizes $\{11, 7, 5, 3\}$, while the covariance is parameterized with a single layer, as suggested in Warkentin 2021, with a kernel window of size 11. The models are trained for 10.000 epochs with the Adam optimization method and a learning rate of 0.001 [Kingma and Ba 2015]. The manifold attractor regularization is used on $\frac{M_{\text{MAR}}}{M} = 0.3$ of the states with $\lambda_{\text{MAR}} = 1$. The effect of different MAR parameters will be investigated later. As the time series are very short, we do not split them up into batches. The evaluation after training is performed on the model associated with the lowest epoch loss, which typically corresponds to the last epoch. To find the optimal number of latent parameters M and bases B, a hyper-parameter grid search is conducted for $M \in \{10, 40, 70, 100, 150\}$ and $B \in \{1, 5, 10, 15\}$.

As can be seen in Figure 13, a larger number of latent states and bases improves performance on the training set up to a RMSE of approximately 0.4 for $M = 70$ and $B = 10$ after which little change can be observed. In contrast to that, the number of dynamical parameters does not affect the performance on the test set, for which the RMSE remains slightly above one. The model is not able to perform a better out-of sample forecast than the mean. This result is also consistent with the categorical precision.

Although the predictions are subpar at this point, we can occasionally observe interesting oscillatory patterns in the freely generated trajectories, e.g. see Figure 14.

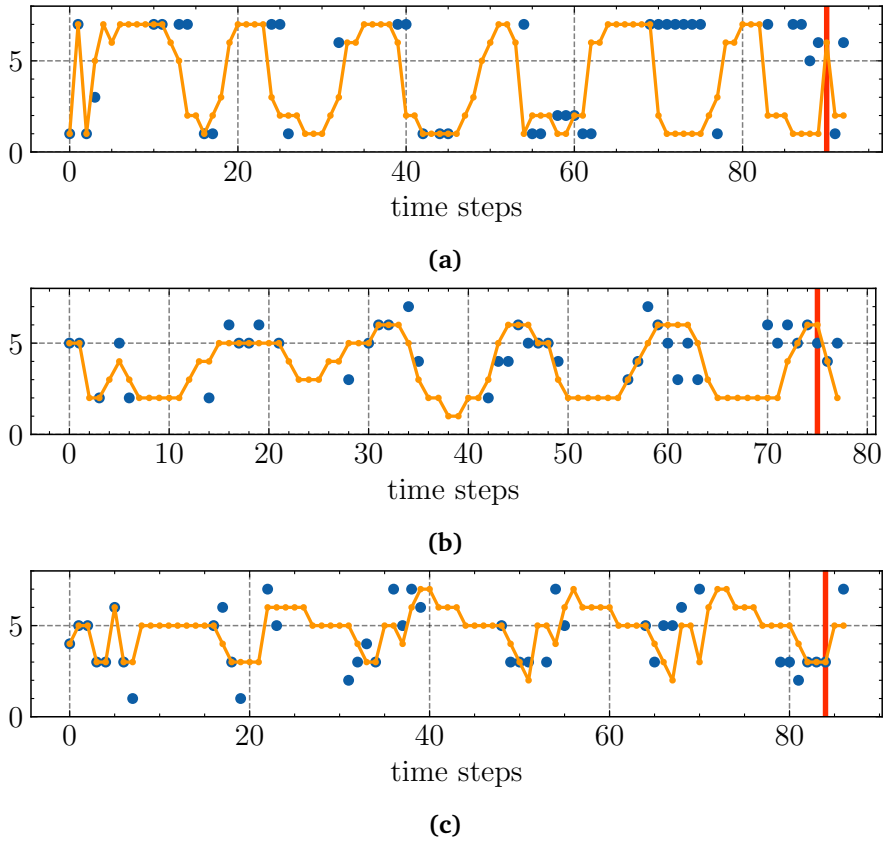
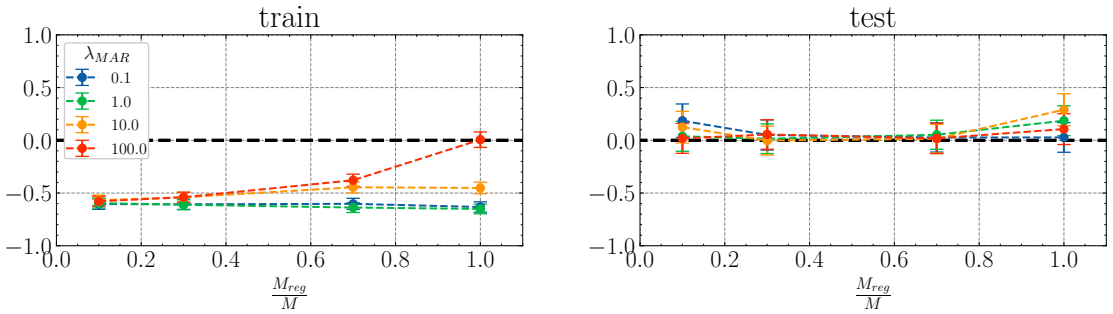


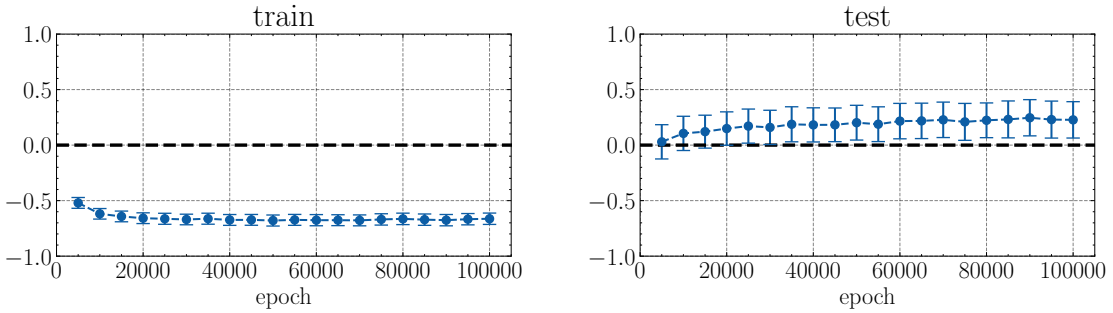
Figure 14: The predicted trajectories of different subjects (orange) are freely generated from $\mathbf{z}_0$. At the start of the test set (marked in red), the latent process is reinitialized using the recognition model.

Of course, it is speculative to assume that the generated patterns here are indicative of some kind of true underlying emotional dynamics, but it is still intriguing to see that the model does sometimes recover more behavior than a simple constant process. If longer time series should become available in the future, it could be especially interesting to test if participants grouped according so some kind of



(a) $\text{RMSE}_{n=1}^{\text{model}} - \text{RMSE}_{n=1}^{\text{mean}}$ **(b)** $\text{RMSE}_{n=1}^{\text{model}} - \text{RMSE}_{n=1}^{\text{mean}}$

Figure 15: The diagram illustrates the 1-step ahead prediction error for different parameter settings for the MAR ($M = 70$, $B = 15$).

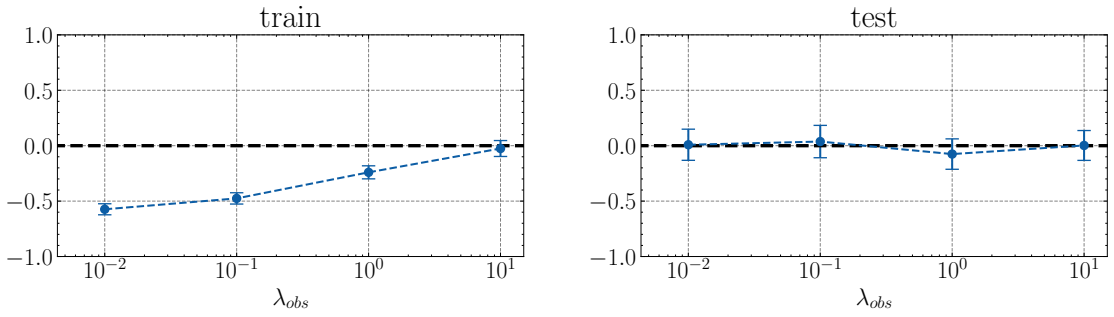


(a) $\text{RMSE}_{n=1}^{\text{model}} - \text{RMSE}_{n=1}^{\text{mean}}$ **(b)** $\text{RMSE}_{n=1}^{\text{model}} - \text{RMSE}_{n=1}^{\text{mean}}$

Figure 16: The 1-step ahead prediction does not change significantly for longer training time (M=70, $B = 15$).

criteria, potentially motivated by psychiatric insight, also show similarities in the underlying latent space.

To see if the prediction performance on the EMIcompass dataset can be improved, the effect of the manifold attractor regularization is evaluated for the parameter settings $\lambda_{\text{MAR}} = \{0.1, 1, 10, 100\}$ and $\frac{M_{\text{reg}}}{M} = \{0.1, 0.3, 0.7, 1.0\}$. As presented in Figure 15, the out-of sample performance is still only equivalent to the mean. Regularizing only a small subset of the states or all of them leads to slightly worse performance on the test set, with the optimal $\frac{M_{\text{reg}}}{M}$ and λ_{MAR} being around 0.3 and 1.0.
Drastically increasing the training time to up to 100.000 epochs only slightly reduces the training error (see Figure 16), while increasing it on the test set, most likely due to stronger overfitting.
As the model consists of a large amount of parameters in contrast to the small number of data points available, an additional regularization term for the observation model



(a) $\text{RMSE}_{n=1}^{\text{model}} - \text{RMSE}_{n=1}^{\text{mean}}$ **(b)** $\text{RMSE}_{n=1}^{\text{model}} - \text{RMSE}_{n=1}^{\text{mean}}$

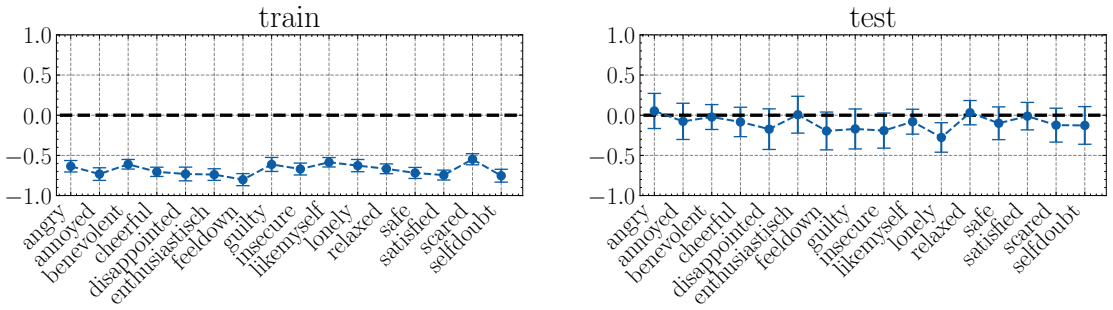
Figure 17: The 1-step ahead prediction error for different L_1 regularization parameters λ_{obs} (M=70, $B = 15$).

parameters is introduced. This is also done in the hopes to encourage the model to put more emphasis on learning the right dynamics via the latent process, instead of overfitting too much through the observation model. We do so by adding a L_1 regularization term to to the likelihood. The $\boldsymbol{\beta}_0$ parameters are not regularized as they define the thresholds between the different ordinal responses, and also make up a smaller fraction of the parameters as they do not scale with the number of latent dimensions M.

$$\mathcal{L}_{\text{reg}}^{\text{obs}} = \lambda_{\text{obs}} \sum_{i}^{N} \sum_{m}^{M} |\beta_{im}| \tag{4.6}$$

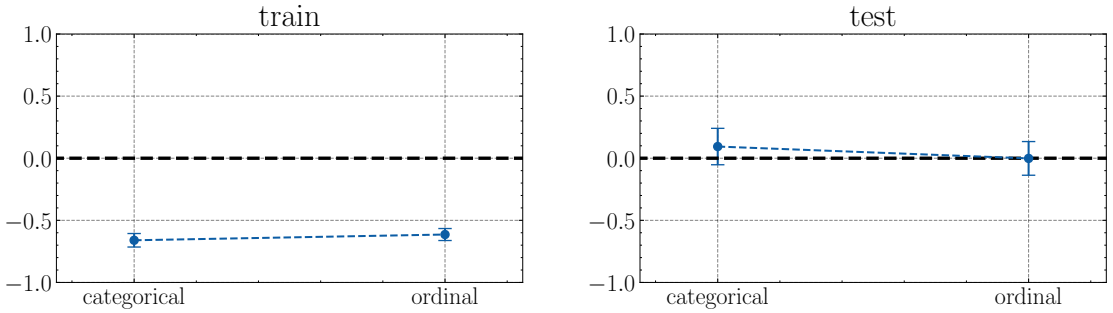
We can see in Figure 17 that the regularization term increases the error on the training set. In contrast to this, we observe no significant change on the test set, where the model performance remains very similar to the mean for all parameter settings. The added regularization seems to push the model more towards constant trajectories, which is not in itself a bad thing, as this might just indicate that the model overfits less of the noise. Again, it is difficult to differentiate between the capabilities of the model and the natural limitations of the data.

To investigate if certain features are more predictable than others, the RMSE is individually calculated for each feature in each subject time series, and averaged in the same way as before over the entire group of participants. Although the performance does not vary greatly between the different Likert items, see Figure 18, we still see for a subgroup of features a fairly significant deviation from the mean prediction error. Interestingly, all the features that are overall easier to predict have the shared characteristic that they are associated with a negative affect. This could potentially be a very interesting finding, which should be investigated more in the



(a) $\text{RMSE}_{n=1}^{\text{model}} - \text{RMSE}_{n=1}^{\text{mean}}$ **(b)** $\text{RMSE}_{n=1}^{\text{model}} - \text{RMSE}_{n=1}^{\text{mean}}$

Figure 18: The 1-step ahead prediction error of each feature averaged over all subject time series ($M = 70,\ B = 15$).

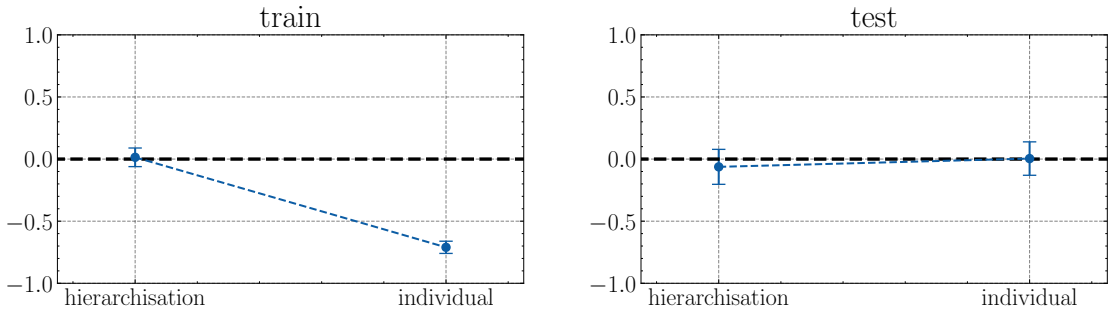


(a) $\text{RMSE}_{n=1}^{\text{model}} - \text{RMSE}_{n=1}^{\text{mean}}$ **(b)** $\text{RMSE}_{n=1}^{\text{model}} - \text{RMSE}_{n=1}^{\text{mean}}$

Figure 19: The 1-step ahead prediction error for the categorical and the ordinal observation model ($M = 70,\ B = 15$).

future to ensure that this is not just a quirk of this specific data set, e.g. through an appropriate statistical test. It is also important to remember that a feature being more predictable is not the same thing as it being more predictive of the underlying emotional dynamics. Although it might turn out that some Likert items are especially difficult to forecast, they still might be very informative for the intervention selection.

Finally, I tested the ordinal against the categorical observation model. We would assume that the categorical model performs better on the training set, as it is more expressive than the ordinal model due to the individual parameterization of each category. Correspondingly, we would hope to see the ordinal model generalize better on the test set. Here, as displayed in Figure 19, there is seemingly little variation between the two observation processes, but we can still see a slight indication of the expected difference. In any case, even if both models work similarly well, one would prefer using the ordinal observation model as it requires less parameters.



(a) $\text{RMSE}_{n=1}^{\text{model}} - \text{RMSE}_{n=1}^{\text{mean}}$ **(b)** $\text{RMSE}_{n=1}^{\text{model}} - \text{RMSE}_{n=1}^{\text{mean}}$

Figure 20: The 1-step ahead prediction error averaged over all subjects for the hierarchical parameter estimation compared to the average performance of the individually trained models ($M = 70,\ B = 15$).

4.2.2 Hierarchical Parameter Estimation

In the hopes of augmenting the predictive strength of the individual time series through group-level information, the hierarchical parameter estimation as described in Section 3.4 is used. All 90 time series are jointly trained, with individual observation model and basis expansion parameters for each time series, while the rest of the parameters are shared across the entire group. In each epoch, the time series are split up into nine batches that each contain ten subject time series. The model is again trained for 10.000 epochs, but now each epoch corresponds to nine gradient updates. As can be observed in Figure 20, the hierarchical parameter estimation does not manage to significantly improve upon the individual-level model forecasts on the test set, while at the same time performing much worse on the training set and therefore seemingly reducing the amount of overfitting. Varying the batch size has little effect on this result. Here, the hierarchical parameter estimation seems to encourage more constant latent space dynamics and thus performs more comparable to the mean. It is at this point very difficult to make any conclusions about the functioning of the procedure. If the data truly does not contain enough information to infer more complicated dynamics, constant latent states would be the expected result. On the other hand, the hierarchical parameter estimation might just generally push the dynamics towards group averages, which of course is not the intended effect.

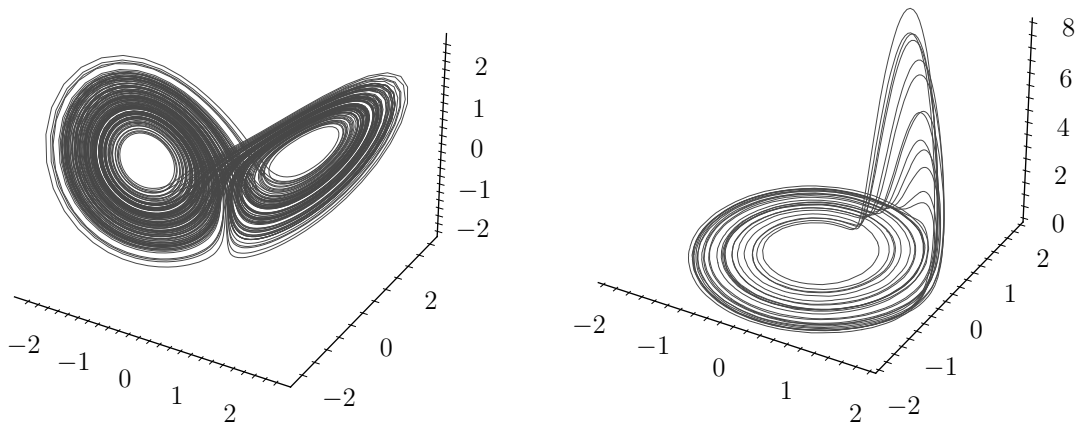
4.3 Benchmark Data

The empirical investigation so far has shown that the data from the EMIcompass study is likely not extensive enough to make clear statements about the functioning of the proposed model setup. It is unclear, where the strengths and weaknesses of the model lie, as we might be dealing with a ceiling effect, where we can not really discern between different model configurations and hyper-parameters. The only real remedy for this issue is to gather more data and especially increase the number of observed time points. At this point in my thesis, I did not have access to a more comprehensive empirical data set, so I focused on creating a method to produce a fairly realistic benchmark data set for the ordinal mobile data. This allows for the generation of unlimited simulated time series data, which can then in turn be used to thoroughly evaluate the model and its limitations. As mentioned before, working with smaller data sets is a common challenge in many scientific contexts, e.g. in medicine and psychiatry [Koppe et al. 2021], so it is especially important to find a principled way to test how much data is needed for sensible forecasts. Thus, a benchmark data set is also very important to inform future study design, e.g. how many participants or what kind of sample frequencies are required.

4.3.1 Underlying Dynamics

The first step for creating benchmark data involves generating artificial latent trajectories from a chaotic system. This is partially motivated from the fact that a number of psychiatric phenomena, such as schizophrenia [Bob et al. 2009] or bipolar disorders and recurrent depressions [Gottschalk et al. 1995; Ortiz et al. 2021] have been associated with chaotic system dynamics [Durstewitz et al. 2021]. Some studies also suggest that especially for healthy subjects mood fluctuations might be determined by chaotic processes [Ortiz et al. 2021]. As the available empirical time series are very short, it is not possible to confirm if the system exhibits chaotic tendencies, or if the irregularities are simply due to a high amount of noise or external inputs to the system. In any case, we will later attempt to confirm that the generated benchmark data is fairly similar to the empirical data.

The goal is now to select an appropriate chaotic dynamical system as a benchmark model of the latent process. The Lorenz attractor [Lorenz 1963] is arguably one of the most famous examples of a chaotic system. The geometry of the attractor is characterized by two "wings", which give rise to the iconic "butterfly" shape. Thus, the



(a) Lorenz attractor **(b)** Rössler attractor

Figure 21: The characteristic 3D-shape of the Lorenz ($\sigma = 10,\ \rho = 28,\ \beta = 8/3$) and the Rössler attractor ($a = 0.1,\ b = 0.1,\ c = 14$).

Lorenz attractor implies a very specific temporal structure that is defined by the time scales contained in each wing, and the time it takes the system to switch between them. As we will later see, the data does not seem to contain two characteristic time scales, which makes the Lorenz system impractical for our purpose. For this reason, I opted for using the so called Rössler attractor that behaves similar to the Lorenz system, but has a simplified topological structure that only contains one spiral, see Figure 21 [Rössler 1976]. The Rössler system is described by a set of three ordinary differential equations.

$$\dot{\mathbf{x}} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = \mathbf{f}(\mathbf{x}) = \begin{pmatrix} -y - z \\ x + ay \\ b + z(x - c) \end{pmatrix} \tag{4.7}$$

In accordance with common practice, the parameters are set to $a = 0.1,\ b = 0.1,\ c = 14$. We also inject the system with process noise, which converts the system of equations $\mathbf{f}(\mathbf{x})$ to a stochastic one [Bärwolff 2020]. For each of the three Rössler dimensions we add an independent driving Wiener process $d\boldsymbol{W} = (dW_x, dW_y, dW_z)^T$. As the Wiener process is not differentiable, we bring the equations into the following form:

$$d\boldsymbol{x} = \mathbf{f}(\mathbf{x})\, dt + \boldsymbol{C} d\boldsymbol{W} = \mathbf{f}(\mathbf{x})\, dt + d\boldsymbol{\epsilon} \tag{4.8}$$

Correspondingly, the matrix $\boldsymbol{C} = \text{diag}(c_x, c_y, c_z)$ is diagonal with all the entries assumed to be constant in time. Each matrix entry c_i is set to $\sqrt{10^{-5}}$ times the standard deviation σ_i of the Rössler system in the respective dimension. In other words, the noise is generated by drawing from the Gaussian term $d\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, 10^{-5} dt \times \mathbf{I})$. To solve the equations and generate trajectories from the system, the famous Itô integral is used. More specifically, I apply the `itoint` routine from the package `sdeint` with a step size of $\Delta t = 0.0005$. The system is randomly initialized close to the attractor, and a transient of 100.000 time steps is cut off.

4.3.2 Ordinal Trajectories

After the latent trajectories $\mathbf{z}_t$ have been generated, we can sample the simulated ordinal trajectories from a categorical distribution parameterized by an ordered-logit model that uses the latent states as input, as discussed in Section 3.3.3.

$$x_{ti} \sim p(x_{ti}|\mathbf{z}_t) = \prod_{k=1}^{K} p(x_{ti} = k|\mathbf{z}_t)^{[x_{ti}=k]} \tag{4.9}$$

The probabilities $p(x_{ti} = k|\mathbf{z}_t)$ are determined by the cumulative probabilities $p(x_{ti} \leq k|\mathbf{z}_t)$ that in turn can be expressed through a generalized linear model with a logit link function.

$$p(x_{ti} \leq k|\mathbf{z}_t) = \frac{\exp(\beta_{ik}^0 - \boldsymbol{\beta}_i^T \mathbf{z}_t)}{1 + \exp(\beta_{ik}^0 - \boldsymbol{\beta}_i^T \mathbf{z}_t)} \tag{4.10}$$

4.3.2.1 Observation Model Parameters

The question becomes how to find sensible parameters $\boldsymbol{\beta}_0$ and $\boldsymbol{\beta}$ for the creation of the benchmark data. First, I attempted to simply reuse the parameters from the observation models that were trained on the EMIcompass dataset. This did not prove to be fruitful, as the parameters need to be in good correspondence with the respective latent process to arrive at reasonable trajectories. I attempted to fine-tune the parameters by hand, e.g. by changing the amplitude of the generated

latent process or by shifting the threshold parameters, but this did not lead to much success.
Finally, I settled on fitting the observation model parameters in such a way that the overall distributions of the generated features match with the histograms of the Likert items calculated from the empirical data (see Figure 9). This is done by using multiple simulated latent trajectories as input and then optimizing the parameters by least squares so that the deviation between the distributions of the empirical and simulated data is minimized. After solving some numerical issues with the optimizer, the procedure worked well, and the simulated data now perfectly reproduces the overall distributions from the EMIcompass data set, see Appx. Figure 26. Thus, the benchmark data also contains the characteristic Gaussian-like and zero-inflated features.
To impose a similar feature correlation structure as observed in the EMIcompass data, the signs of a subset of the $\boldsymbol{\beta}_i$ parameters were manually changed. In doing so, we arrive at a similar Spearman rank order correlation matrix, see Appx. Figure 27, compared to the one estimated from the empirical data.

4.3.2.2 Time Scales

So far, we only focused on reproducing overall properties of the empirical data, but neglected to enforce a specific temporal structure. A general challenge lies in the fact that the given empirical time series are very short, so it is difficult to be certain about what kind of dynamics should be introduced into the benchmark data.
To gain a sense of the time scales present in the empirical data, the time series are fast Fourier transformed feature-wise (by using the `np.fft.rfft` routine). The individual subject time series are not long enough to perform a Fourier transformation in a sensible way. For this reason, one long time series is constructed by attaching multiple subject time series to each other. Time series that exhibit very small variance and only nearly constant behavior are discarded, as they do not provide us with any interesting information about the temporal dynamics. In future more sophisticated criteria might be thought of to sensibly group the participants according to the observed dynamics. Missing value gaps that are longer than eight time steps, which mostly correspond to the night phases, are cut out of the time series. The rest of the missing observations are imputed by a moving average with a Gaussian kernel. Additionally, the data is standardized before apply the Fourier transformation. Finally, the resulting power spectrum is smoothed with a Gaussian kernel. Before plotting,

the frequencies are re-scaled to account for the filtered out night phases.

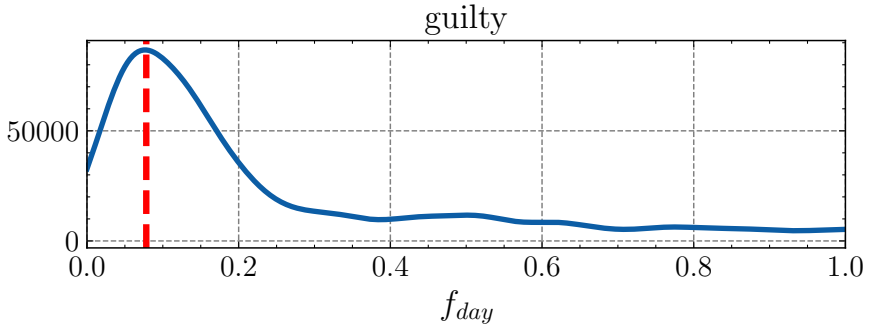


Figure 22: Smoothed power spectrum of the Likert item associated with guilt. The red line indicates the largest frequency component, and the frequencies are given in days under the assumption that two time steps are 1.5 hours apart.

An example power spectrum of a single Likert item is presented in Figure 22, for all features see Appx. Figure 28. We do not observe a very sharp peak in the power spectrum, but can still clearly see that most of the power is found in the low-frequency components. On average, the maximum frequency corresponds to a period length of ~ 13.5 days.

In general, these low-frequency oscillations can seemingly be found in the data, but it is difficult to truly distinguish them from potential non-stationary behavior. Here, the empirical time series on average only cover a period of around seven days, so we are not even able to observe a full oscillation, which makes it effectively impossible to accurately answer this question with the available data [Kantz and Schreiber 2004]. Therefore, it is definitely necessary to repeat the analysis on longer empirical time series to gain more certainty about the presented result. If it turns out that non-stationary behavior can indeed be commonly observed in the dynamics of such time series, appropriate modeling strategies would need to found.

Additionally, the question remains if there are truly no higher frequency patterns present as the power spectrum would suggest. In many of the time series, we can observe behavior at smaller time scales, but it is difficult to say if it is due to the high stochasticity of the system or if it can be attributed to some kind of underlying dynamics. We also need to consider that the irregular patterns might be due to some unknown external context that can not be easily modeled as noise. For instance, an unexpected event, such as a supervisor sending a fascinating paper or winning the lottery, will of course have a strong impact on the emotional trajectories that cannot be feasibly predicted by the model. Thus, it might be crucial to integrate additional

information into the model framework, for instance provided by the smartphone sensors, e.g. the movement patterns of a person. The EMIcompass data set also contains a question on the type of social activity a person might be engaged in, which could for instance be used as an external input for the PLRNN.

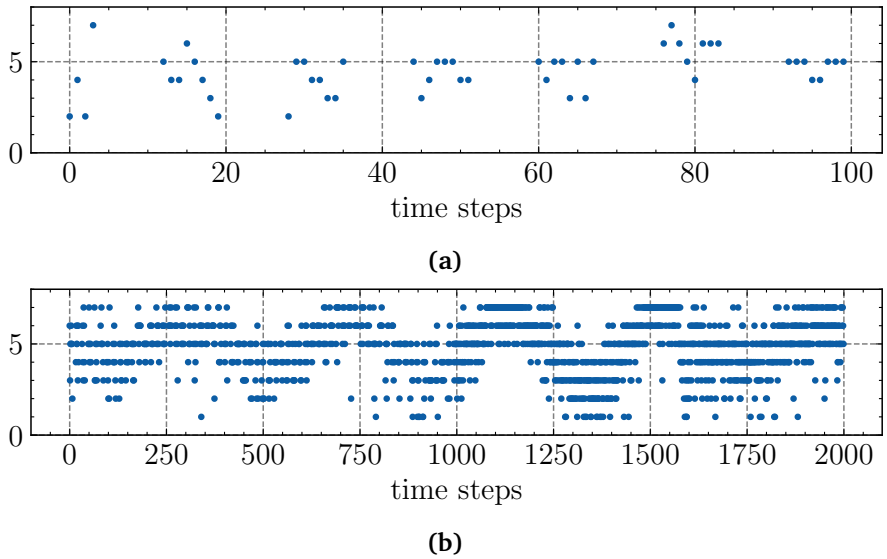


Figure 23: Benchmark trajectories sampled from an underlying Rössler system through an ordinal observation model.

For now, we proceed under the assumption that only low-frequency oscillations are present in the data, and therefore attempt to induce the same time-scale in the benchmark data. This can be simply done by sub-sampling the generated trajectories. We find that by taking every 30th time point, we can arrive at a similar power spectrum for the benchmark data, see Figure 24. Finally, we mimic the day and night structure from the empirical data by introducing missing values into the time series. We assume that each day corresponds to 16 time steps from which the last eight are set to be missing.

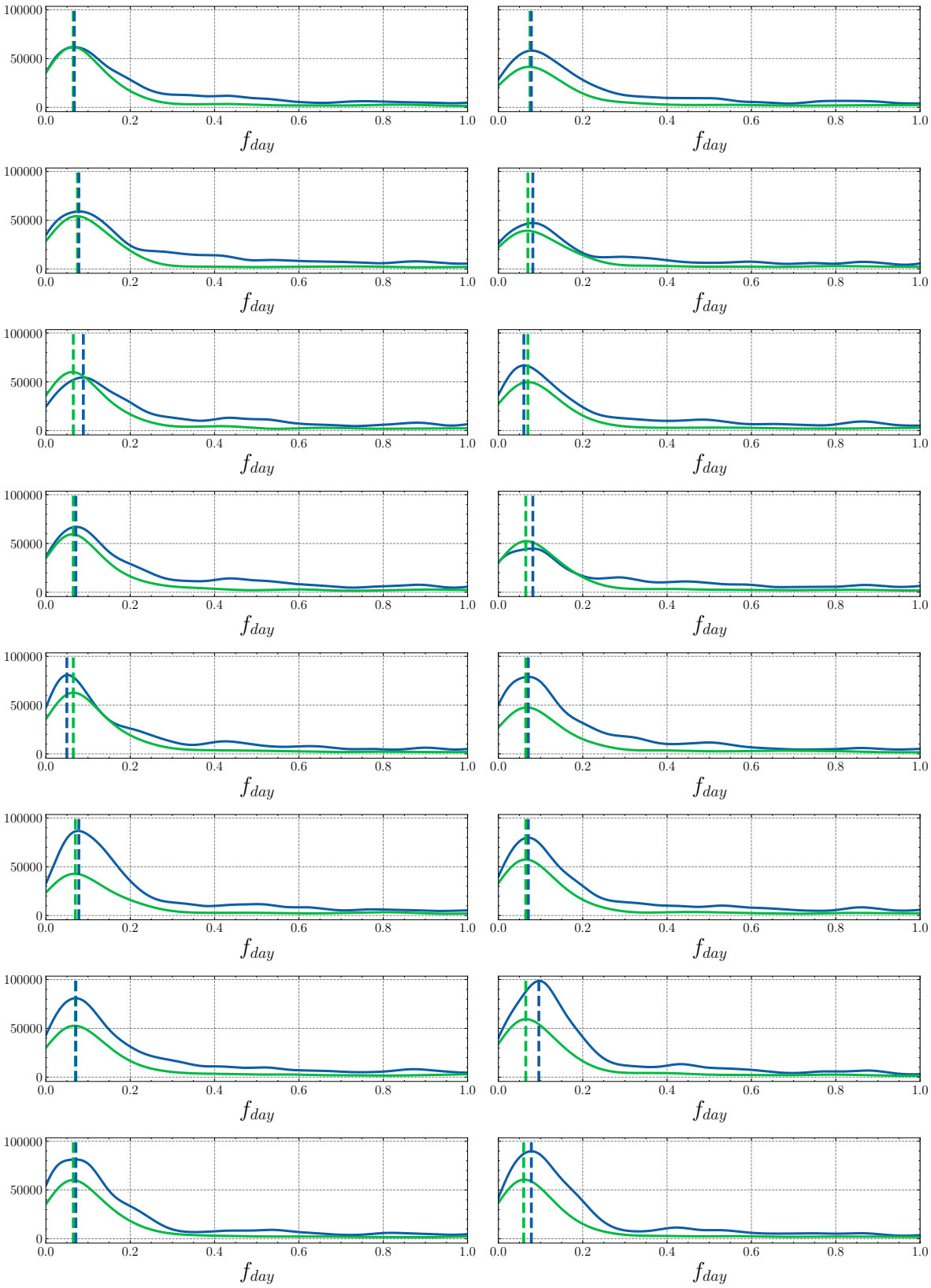
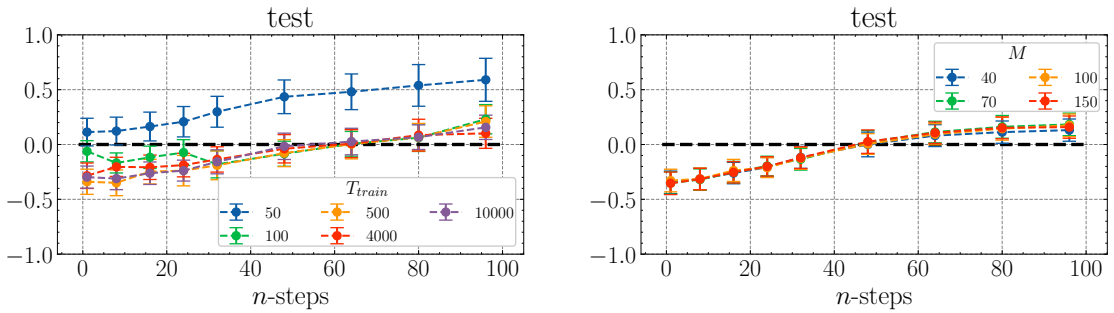


Figure 24: Power spectrum of the benchmark data (green) overlayed with the power spectrum from the EMIcompass data set (blue). The dotted lines indicate the maximum frequency components.



(a) $\text{RMSE}_n^{\text{model}} - \text{RMSE}_n^{\text{mean}}$ **(b)** $\text{RMSE}_n^{\text{model}} - \text{RMSE}_n^{\text{mean}}$

Figure 25: The left diagram shows model performance for different training set sizes T_{train} ($M = 40$, $B = 10$). On the right, we see the impact of the number of latent parameters M on the error for $T_{\text{train}} = 10.000$.

4.3.3 Model Evaluation

Figure 23 shows two simulated trajectories covering different time periods. Only long-term dynamics are present, and the time series are overall quite noisy. The number of time steps of the upper trajectory corresponds to the average time frame that the empirical time series cover, and it generally seems comparable to the statistical difficulty of the EMIcompass dataset. Apart form the question if the right time scales were introduced into the simulated trajectories, it is also difficult to control the amount of noise present. Overall, the noise level is determined by the imposed distribution of the individual Likert items, but this does not directly control the noise at each time step. All in all, the created trajectories are very challenging, as they are based on a chaotic system, exhibit much stochasticity, and only contain slow moving dynamics.

Models are trained on the benchmark data with the same parameter setup described in Section 4.2.1. Each hyper-parameter combination is evaluated by training 50 individual models and averaging the error metrics as before. Model performance is compared to using the constant mean of the feature distributions as forecast.

The primary motivation for generating the benchmark data was to assess model performance for larger amounts of training data T_{train}. As can be observed on the left in Figure 25, increasing the number of time steps for training does indeed improve the out-of sample error. For longer time series, the forecasts outperform the mean for up to a 50-step ahead prediction. For longer ahead predictions the model performance is again worse in relation to the mean forecasts. As we provide the model with up to 10.000 time steps, this definitely should not happen. Even for chaotic systems, we

would always want the model to at least recover the mean for long-term forecasts. The right figure shows that even with a large amount of training data, the number of latent dimensions has very little impact on the resulting performance. This is not what we would necessarily expect, as we would hope that for effectively unlimited training data and dynamical parameters the generalization error would fall further. Tuning the MAR also has little impact on this result, see Appx. Figure 29.

Currently, the model is not able to reconstruct the dynamical system, and only manages to provide a reasonable local forecast. In general, the models do not exhibit much dynamical behavior, and the latent states mostly run into fixed-points. This allows the model to perform short term ahead predictions when initialized correctly, but performance than quickly falls off, as it is not able to truly recover the underlying dynamics.

Finally, we notice that the performance of the models trained on the simulated trajectories containing 100 time points is roughly equivalent to the results we observed on the empirical data set. This might indicate that the generated benchmark data has some correspondence with the empirical data set. On the other hand, it is important to point out that the models trained on the EMIcompass data generally showed more dynamical behavior, which might be due to some faster underlying dynamics that were missed when generating the benchmark data.

5 Discussion and Conclusion

A central goal of this thesis was to build up a model that is capable of directly handling ordinal time series, without requiring the assumption that the data can be approximately treated as metric or categorical. This was realized by making use of an ordered logit model, that was integrated as an observation model into the sequential variational autoencoder framework.

Before discussing the empirical results, it is important to mention that while the central model components respect the ordinal character of the data, the same can not be said for the measures that were used during the empirical evaluation. As discussed in Section 3.3.2, the numerical encoding of the ordinal categories is somewhat arbitrary, as we do not have information about the distances between different items. This poses a challenge for model evaluation, as calculating the expected value of the observation model $\mathbb{E}_{p_\theta(\mathbf{x}|\mathbf{z})}[\mathbf{x}_{k+n}|\hat{\mathbf{z}}_{k+n}]$ of course implies that we presume that the different ordinal responses are equidistant, which might simply not be true and mislead us depending on the distribution of the relevant features. Additionally, this makes it difficult to choose a sensible error metric, as ordinal data is neither truly categorical nor metric. It is possible to use measures commonly used for nominal classification, e.g. precision or the confusion matrix, but this does not feel satisfying as we ignore the ordering in the data. In other words, it seems worse to misclassify a very unhappy person as happy than to confuse a happy participant with a moderately happy subject. Alternatively, we can calculate the mean squared error (MSE), or a similar measure, but in doing so we pretend that the ordinal classes live in an equidistant metric space, which we can not simply presuppose. For instance, for a zero-inflated feature it should be more important for the model to correctly discern between the first and the second category than between the second and third, as the first jump is likely more meaningful and might hint at a larger change in the underlying dynamics. Due to time constraints, I made the compromise of working with the MSE, but double checked the results by calculating the categorical precision. So far I did not encounter a situation, in which the error measures differed significantly. Still, I would recommend that for future testing the possibility of designing a purely ordinal evaluation method should definitely be explored.

In the literature I reviewed I could not find a clear consensus on how to best proceed, and a variety of approaches have been proposed for evaluating ordinal data, e.g. [Baccianella et al. 2009; Gaudette and Japkowicz 2009; Cardoso and Sousa

2011; Amigó et al. 2020; Sakai 2021]. They all vary greatly, and also differ on the question if equally spaced intervals can be safely assumed. This choice also likely depends on the problem at hand, the exact distribution of the data, and might need to be informed by domain knowledge. It might especially fruitful and interesting to review work on how to best extract metric information from ordinal data [Shepard 1966], e.g. by exploiting the overall feature distribution to find sensible weightings for the different intervals between the ordinal categories. It is also important to consider that the perception of ordinal scales might vary in a participant population, and various individuals might interpret the Likert item intervals very differently. This might make it necessary to find a different approximation of the ordinal to the metric scale for each participant.

In Section 4.2, the model performance was evaluated on the real-world EMIcompass data set for several different hyper-parameter settings. Overall, the model is not able to perform better out-of sample forecasts than the mean. In the hopes of reducing overfitting, I introduced an additional regularization term for the observation model, which proved to have little effect on the generalization error.
This is obviously not a very satisfying result, but it needs to be stressed that the time series are very challenging from a statistical standpoint. As was seen, many of the subject time series seemingly exhibit very slow time scales or irregular behavior while at the same time being very short and containing a sizable amount of missing values. Additionally, many of the time series show non-stationary behavior, e.g. see Figure 11, that is difficult to distinguish from low frequency oscillations. In total, it might just not be feasible to generate predictions that are significantly better than the mean on the available dataset. In other words, it oftentimes seems unclear if there is even a pattern present in the data for the model to recover.
Nevertheless, when honing in on specific features or participants, it is possible to observe interesting dynamics that also seem to have some correspondence to the empirical data, e.g. see Figure 14. This might indicate that for future analysis, it is especially important to zoom in on specific participant and feature subsets that prove to be more predictable than others.
Here, for instance we found that Likert items that are associated with negative affects, such as loneliness or guilt, can be predicted with higher accuracy that also surpasses the mean forecast. As a next step, it should be investigated what exactly makes the negative emotional states easier to predict, e.g. if it is possible to find clear similarities in their dynamical behavior. Of course such a finding would need

to be corroborated with robust statistical testing, and should be reproduced on a more extensive data set, but it still might give some indication in which direction future efforts should be focused. It might also be interesting to see, if such insight could be connected to available domain knowledge on psychological phenomena. For instance, negative events and impressions seem to generally have a more dominant impact on people's behavior and emotional states [Lewicka et al. 1992; Baumeister et al. 2001]. As mentioned before, an investigation of the features also needs to be tied to the specific application setting. If some features prove to be less predictable than others, they still might be very predictive for the selection of the right interventions. In any case, a careful analysis of the different EMA features could also provide valuable information for future studies, e.g. if a questionnaire should specifically focus on one sub-type. Dropping some of the Likert items, could also allow for more densely sampled trajectories, as it would take participants less time to fill out a single questionnaire. In my opinion, a larger number of time points is likely more important than a larger number of Likert items, as they exhibit a fair amount of dependency and oftentimes similar time scales.

In similar fashion, one could attempt to group participants that show similarities in their underlying dynamics. It would be extremely interesting to see, if individuals with a comparable latent space, can also be matched according to another similarity measure, ideally motivated by psychiatric insight. Additionally, one could check, how much model performance varies in between subgroups. For instance, certain dynamical patterns might be easier to predict than others, which could also inform future model development.

In the hopes of exploiting such group similarities, I also implemented and tested the first version of a hierarchical parameter estimation. First results did not show much improvement on the test set, and the hierarchisation generally led to more constant latent space dynamics. Of course, it is difficult to draw a final conclusion, as the empirical data set is fairly limited, but I suspect that the chosen parameter split for group and individual-level inference was still far from optimal and needs to be further explored. For instance, it might be more sensible to at least train parts of the parameter-rich connection matrix $\mathbf{W}$ on the individual level, while sharing the thresholds of the basis expansion over the entire group. As mentioned before, it might also be useful to group participants from larger study cohorts before training the models [Cearns et al. 2019].

In general, it seems likely that to reach a significant performance boost on empirical data sets, a fine-tuned hierarchisation procedure will be necessary. As seen before,

the self-reported information of a single individual might oftentimes not be comprehensive enough to fully train a complicated model on. The hierarchisation might then be a structured way to integrate group-level insight, while still allowing for enough individual variation to construct truly personalized models, which are likely very crucial for improving future mental health treatments [Chekroud et al. 2017]. Along the same line, it might be fruitful to find ways to integrate prior domain knowledge from psychiatry or computational neuroscience into the model framework. For instance, it might be possible to incorporate existing insight on what kind of underlying dynamics to expect by defining sensible priors for the latent model parameters, similar to a fully Bayesian framework [Sayer 2020]. This could also inspire the introduction of new expressive parameters that correspond with other subject-specific information, such as epigenetic risk factors [Keverne and Binder 2020].

In addition to the empirical investigation of the EMIcompass data set, I attempted to create realistic benchmark data to further test the model's capabilities on, see Section 4.3. The underlying latent process was modeled by using the chaotic Rössler attractor. The ordinal trajectories were then sampled through an ordered logit model, for which the model parameters were fitted in such a way that the overall feature distribution corresponded to the EMIcompass data set. Additionally, I manually introduced a similar feature correlation structure as observed in the real-world data, and attempted to mimic the dominant time scales by sub-sampling the time series.
As discussed, it is difficult to say if the dynamics of the benchmark data are a truly accurate representation of the real-world data. The large amount of power observed in the low-frequency components might be a strong indication that they need to be treated as non-stationary [Kantz and Schreiber 2004]. Additionally, I suspect that there might be predictive behavior at smaller time scales, at least for some individuals, that was missed when calculating the power spectrum over multiple participants. In total, the creation of the benchmark data needs to be repeated with better data, which should be fairly straightforward now that the process is set up. In addition, alternative techniques for generating benchmark data could be tried out, e.g. block bootstrapping, where a time series gets divided into multiple contiguous blocks from which new time series can be produced by sampling with replacement [Kreiss and Paparoditis 2011].
A preliminary evaluation of the benchmark data was performed by training multiple

models for different training set sizes. In general, for a large enough number of observations the model provides reasonable short-term forecasts, but falls off in its accuracy for longer ahead predictions. It is surprising that at some point the model performance becomes worse than the mean again, as we would expect the model to at least recover the overall mean as a long-term forecast. In general, the model is struggling to reconstruct the underlying Rössler dynamics, and the latent states oftentimes become fixed-points. The exact reason why this is happening still needs to be investigated, but it is also important to mention that while performance is comparable for a similar training set size, the models trained on the empirical data exhibit much more varied dynamical behavior. This might indicate that the benchmark data still might be missing faster dynamical patterns that potentially exist in the EMIcompass data.
It is important to note that the hierarchisation procedure was not tested on the benchmark data. To do so in a principled manner would require adjusting the benchmark generation process so that multiple individual time series can be created that still share some similarities in their dynamical behavior. This could for instance be realized by using the same underlying latent process, e.g. the Rössler attractor, but choosing different parameter regions for each time series. It could then be tested, if the parameter hierarchisation can recover the overall dynamical system, and can manage to encode the differing dynamical system parameters in the individually inferred model parameters.

Besides improving the hierarchical parameter estimation, there are also several other changes to the model architecture that might positively impact the empirical results. For the recognition model a completely diagonal covariance matrix was chosen. This is likely a too simplifying assumption, and it might be necessary to test alternative encoder models with a more complicated covariance structure. For instance, it might be possible to combine the usage of a CNN with the block-tridiagonal parameterization proposed in Archer et al. 2015. It might also be fruitful to consider a recognition model that can directly handle missing values, e.g. by making use of a PLRNN, similar to the second imputation technique that was proposed. As long as model interpretability is a secondary goal, one could also try out different generative models, such as LSTMs or a deep PLRNN consisting of multiple layers.

Overall, I think that the highest performance jump will arguably be tied to the integration of other data modalities. As discussed before, collecting self-reported information is difficult and costly, as the patience of participants to answer extensive questionnaires multiple times a day will always be limited [Wen et al. 2017]. High sampling frequencies might also become strenuous and put an unacceptable mental burden on the user [Stone et al. 2007]. Additionally, solely relying on self-reported data might also carry the danger of introducing biases; for example, the validity of electronic self-monitoring of mood for patients suffering from mania has been questioned [Faurholt-Jepsen et al. 2016].
Therefore, the usage of passively gathered sensor data might be especially appealing for the construction of personalized models, as they can be collected without user participation and with a much higher sampling frequency [Durstewitz et al. 2019; Seppälä et al. 2019; Koppe et al. 2021]. A variety of smartphone-based sensor data has been shown to be potentially predictive for mental health and general mood, ranging from mobility patterns inferred from geo-location traces [Canzian and Musolesi 2015; Mikelsons et al. 2017], application usage and communication history [Likamwa et al. 2013], physical activity levels measured through accelerometers [Rodriguez et al. 2017] to microphone data [Abdullah et al. 2016]. Taken together, the integration of such rich and densely sampled sensor data might allow the model to make better sense of the irregular patterns and non-stationary behavior observed in the EMA trajectories, e.g. a seemingly random spike in a Likert item might become predicable when taking into the account the unusual communication and mobility patterns observed before. In addition to sensor data, it might also be prudent to make better use of missing information, as mentioned in Section 3.2.2.
As discussed in Section 2.3.7, multimodal data can in principle be easily included into the sequential variational autoencoder framework. For instance, I implemented a version of the ZIP model, based on work from Bommer et al. 2021, that could be used to to describe step counts measured by smartphone sensors. The main difficulty for integrating various features will be how to deal with their varying time scales. As mentioned, the sampling rate of sensor data will of course widely differ from the self-reported questionnaires, making the discretization of the data into equidistant time steps rather difficult. We then might require a specifically structured latent model [Che et al. 2018a] or a continuous model formulation [Chen et al. 2018; Rubanova et al. 2019; Monfared and Durstewitz 2020b] to be able to train successfully on differently sampled data.

Overall, the methodological approach that I started to develop in the context of this thesis holds a lot of promise from an application perspective. It offers a potentially useful way to forecast ordinal time series using a probabilistic deep learning model. The insights gained during this investigation reveal fruitful directions future research can take to realize this model's potential, not only in the field of psychiatry, but also other fields that rely on similar kinds of data. Especially as more and more mobile data becomes available, the need for models that solve the problems this thesis has identified will only increase.

Bibliography

Likert, R. (1932). “A Technique for the Measurement of Attitudes”. *Archives of Psychology* 22 140, pp. 55–55.

Aitchison, J. and Silvey, S. D. (1957). “The Generalization of Probit Analysis to the Case of Multiple Responses”. *Biometrika* 44:1/2, pp. 131–140. DOI: 10.2307/2333245.

Kalman, R. E. (1960). “A New Approach to Linear Filtering and Prediction Problems”. *Journal of Basic Engineering* 82:1, pp. 35–45. DOI: 10.1115/1.3662552.

Lorenz, E. N. (1963). “Deterministic Nonperiodic Flow”. *Journal of the Atmospheric Sciences* 20:2, pp. 130–141. DOI: 10.1175/1520-0469(1963)020<0130:DNF>2.0.CO;2.

Shepard, R. N. (1966). “Metric Structures in Ordinal Data”. *Journal of Mathematical Psychology* 3:2, pp. 287–315. DOI: 10.1016/0022-2496(66)90017-4.

Rössler, O. E. (1976). “An Equation for Continuous Chaos”. *Physics Letters A* 57:5, pp. 397–398. DOI: 10.1016/0375-9601(76)90101-8.

Rubin, D. B. (1976). “Inference and Missing Data”. *Biometrika* 63:3, pp. 581–592. DOI: 10.2307/2335739.

McCullagh, P. (1980). “Regression Models for Ordinal Data”. *Journal of the Royal Statistical Society. Series B (Methodological)* 42:2, pp. 109–142.

Verbrugge, L. M. (1980). “Health Diaries”. *Medical Care* 18:1, pp. 73–95. DOI: 10.1097/00005650-198001000-00006.

Winship, C. and Mare, R. D. (1984). “Regression Models with Ordinal Variables”. *American Sociological Review* 49:4, pp. 512–525. DOI: 10.2307/2095465.

O’Brien, R. M. (1985). “The Relationship between Ordinal Measures and Their Underlying Values: Why All the Disagreement?” *Quality and Quantity* 19:3, pp. 265–277. DOI: 10.1007/BF00170998.

Csikszentmihalyi, M. and Larson, R. (1987). “Validity and Reliability of the Experience-Sampling Method”. *The Journal of Nervous and Mental Disease* 175:9, pp. 526–536.

Ahmed, N. and Gokhale, D. (1989). “Entropy Expressions and Their Estimators for Multivariate Distributions”. *IEEE Transactions on Information Theory* 35:3, pp. 688–692. DOI: 10.1109/18.30996.

Lambert, D. (1992). “Zero-Inflated Poisson Regression, with an Application to Defects in Manufacturing”. *Technometrics* 34:1, pp. 1–14. DOI: 10.2307/1269547.

Lewicka, M. et al. (1992). “Positive-Negative Asymmetry or ‘When the Heart Needs a Reason’”. *European Journal of Social Psychology* 22:5, pp. 425–434. DOI: 10.1002/ejsp.2420220502.

Funahashi, K.-i. and Nakamura, Y. (1993). “Approximation of Dynamical Systems by Continuous Time Recurrent Neural Networks”. *Neural Networks* 6:6, pp. 801–806. DOI: 10.1016/S0893-6080(05)80125-X.

Molenberghs, G. and Lesaffre, E. (1994). “Marginal Modeling of Correlated Ordinal Data Using a Multivariate Plackett Distribution”. *Journal of the American Statistical Association* 89:426, pp. 633–644. DOI: 10.2307/2290866.

Gottschalk, A. et al. (1995). “Evidence of Chaotic Mood Variation in Bipolar Disorder”. *Archives of General Psychiatry* 52:11, pp. 947–959. DOI: 10.1001/archpsyc.1995.03950230061009.

Kleijnen, J. P. and Rubinstein, R. Y. (1996). “Optimization and Sensitivity Analysis of Computer Simulation Models by the Score Function Method”. *European Journal of Operational Research* 88:3, pp. 413–427. DOI: 10.1016/0377-2217(95)00107-7.

Mathieson, M. (1996). “Ordered Classes and Incomplete Examples in Classification”. In: *Proceedings of the 9th International Conference on Neural Information Processing Systems*. MIT Press, Denver, Colorado, pp. 550–556.

Williamson, J. and Kim, K. (1996). “A Global Odds Ratio Regression Model for Bivariate Ordered Categorical Data from Ophthalmologic Studies”. *Statistics in Medicine* 15:14, pp. 1507–1518. DOI: 10.1002/(SICI)1097-0258(19960730)15:14<1507::AID-SIM316>3.0.CO;2-Z.

Jordan, M. I. et al. (1998). “An Introduction to Variational Methods for Graphical Models”. In: *Learning in Graphical Models*. Ed. by M. I. Jordan. Springer Netherlands, Dordrecht, pp. 105–161. DOI: 10.1007/978-94-011-5014-9_5.

Minois, G. (1998). *Geschichte der Zukunft: Orakel - Prophezeiungen - Utopien - Prognosen*. Trans. by E. Moldenhauer. Artemis & Winkler, Düsseldorf Zürich.

Böckenholt, U. (1999). “Measuring change: Mixed Markov models for ordinal panel data”. *British Journal of Mathematical and Statistical Psychology* 52:1, pp. 125–136. DOI: https://doi.org/10.1348/000711099159008.

Tashman, L. J. (2000). “Out-of-sample tests of forecasting accuracy: an analysis and review”. *International Journal of Forecasting* 16:4, pp. 437–450. DOI: https://doi.org/10.1016/S0169-2070(00)00065-0.

Von Korff, M. et al. (2000). “Assessing Global Pain Severity by Self-Report in Clinical and Health Services Research”. *Spine* 25:24, pp. 3140–3151.

Weinzierl, S. (2000). “Introduction to Monte Carlo Methods”. *ArXiv High Energy Physics - Phenomenology e-prints*. DOI: 10.1007/978-0-387-87837-9_1.

Baumeister, R. F. et al. (2001). “Bad Is Stronger than Good”. *Review of General Psychology* 5:4, pp. 323–370. DOI: 10.1037/1089-2680.5.4.323.

Pruscha, H. and Göttlein, A. (2003). “Forecasting of Categorical Time Series Using a Regression Model”. 18:2, pp. 223–240. DOI: 10.1515/EQC.2003.223.

Goldberg, R. M. et al. (2004). “A Randomized Controlled Trial of Fluorouracil plus Leucovorin, Irinotecan, and Oxaliplatin Combinations in Patients with Previously Untreated

Metastatic Colorectal Cancer". *Journal of Clinical Oncology: Official Journal of the American Society of Clinical Oncology* 22:1, pp. 23–30. DOI: 10.1200/JCO.2004.09.046.

Kantz, H. and Schreiber, T. (2004). *Nonlinear Time Series Analysis*. 2nd ed. Cambridge University Press, Cambridge, UK ; New York.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, New York.

Johnson, V. E. and Albert, J. H. (2006). *Ordinal Data Modeling*. Springer Science & Business Media.

Lee, K. and Daniels, M. J. (2007). "A Class of Markov Models for Longitudinal Ordinal Data". *Biometrics* 63:4, pp. 1060–1067. DOI: 10.1111/j.1541-0420.2007.00800.x.

Rodrígues, G. (2007). *Lecture Notes on Generalized Linear Models*. Available at https://data.princeton.edu/wws509/notes/.

Stone, A. et al. (2007). *The Science of Real-Time Data Capture: Self-Reports in Health Research*. Oxford University Press.

Todem, D. et al. (2007). "Latent-Variable Models for Longitudinal Data with Bivariate Ordinal Outcomes". *Statistics in Medicine* 26:5, pp. 1034–1054. DOI: 10.1002/sim.2599.

Cheng, J. et al. (2008). "A Neural Network Approach to Ordinal Regression". In: *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pp. 1279–1284. DOI: 10.1109/IJCNN.2008.4633963.

Lee, K. and Daniels, M. J. (2008). "Marginalized Models for Longitudinal Ordinal Data with Application to Quality of Life Studies". *Statistics in Medicine* 27:21, pp. 4359–4380. DOI: 10.1002/sim.3352.

Tzikas, D. G. et al. (2008). "The Variational Approximation for Bayesian Inference". *IEEE Signal Processing Magazine* 25:6, pp. 131–146. DOI: 10.1109/MSP.2008.929620.

Baccianella, S. et al. (2009). "Evaluation Measures for Ordinal Regression". In: *2009 Ninth International Conference on Intelligent Systems Design and Applications*, pp. 283–287. DOI: 10.1109/ISDA.2009.230.

Ben-Zeev, D. et al. (2009). "Retrospective Recall of Affect in Clinically Depressed Individuals and Controls". *Cognition and Emotion* 23:5, pp. 1021–1040. DOI: 10.1080/02699930802607937.

Bob, P. et al. (2009). "Chaos in schizophrenia associations, reality or metaphor?" *International Journal of Psychophysiology* 73:3, pp. 179–185. DOI: https://doi.org/10.1016/j.ijpsycho.2008.12.013.

Cagnone, S. et al. (2009). "Latent Variable Models for Multivariate Longitudinal Ordinal Responses". *British Journal of Mathematical and Statistical Psychology* 62:2, pp. 401–415. DOI: 10.1348/000711008X320134.

Ebner-Priemer, U. W. and Trull, T. J. (2009). “Ecological Momentary Assessment of Mood Disorders and Mood Dysregulation”. *Psychological Assessment* 21:4, pp. 463–475. DOI: 10.1037/a0017075.

Gaudette, L. and Japkowicz, N. (2009). “Evaluation methods for ordinal classification”. In: *Canadian conference on artificial intelligence*. Springer, pp. 207–210.

Hastie, T. et al. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed. Springer Series in Statistics. Springer, New York.

Ahmed, N. K. et al. (2010). “An Empirical Comparison of Machine Learning Models for Time Series Forecasting”. *Econometric Reviews* 29:5-6, pp. 594–621. DOI: 10.1080/07474938.2010.481556.

Greene, W. H. and Hensher, D. A. (2010). *Modeling Ordered Choices: A Primer*. Cambridge University Press, Cambridge. DOI: 10.1017/CBO9780511845062.

Norman, G. (2010). “Likert Scales, Levels of Measurement and the "Laws" of Statistics”. *Advances in Health Sciences Education* 15:5, pp. 625–632. DOI: 10.1007/s10459-010-9222-y.

Pan, S. J. and Yang, Q. (2010). “A Survey on Transfer Learning”. *IEEE Transactions on Knowledge and Data Engineering* 22:10, pp. 1345–1359. DOI: 10.1109/TKDE.2009.191.

Varin, C. and Czado, C. (2010). “A Mixed Autoregressive Probit Model for Ordinal Longitudinal Data”. *Biostatistics* 11:1, pp. 127–138. DOI: 10.1093/biostatistics/kxp042.

Cardoso, J. S. and Sousa, R. (2011). “Measuring the Performance of Ordinal Classification”. *International Journal of Pattern Recognition and Artificial Intelligence* 25:08, pp. 1173–1195. DOI: 10.1142/S0218001411009093.

Kreiss, J.-P. and Paparoditis, E. (2011). “Bootstrap Methods for Dependent Data: A Review”. *Journal of the Korean Statistical Society* 40:4, pp. 357–378. DOI: 10.1016/j.jkss.2011.08.009.

Yoon, J. W. et al. (2011). “Bayesian inference for an adaptive Ordered Probit model: An application to Brain Computer Interfacing”. *Neural Networks* 24:7, pp. 726–734. DOI: https://doi.org/10.1016/j.neunet.2011.03.019.

Bergmeir, C. and Benítez, J. M. (2012). “On the Use of Cross-Validation for Time Series Predictor Evaluation”. *Information Sciences*. Data Mining for Software Trustworthiness 191, pp. 192–213. DOI: 10.1016/j.ins.2011.12.028.

Bystritsky, A. et al. (2012). “Computational Non-Linear Dynamical Psychiatry: A New Methodological Paradigm for Diagnosis and Course of Illness”. *Journal of Psychiatric Research* 46:4, pp. 428–435. DOI: 10.1016/j.jpsychires.2011.10.013.

Devore, J. L. and Berk, K. N. (2012). *Modern Mathematical Statistics with Applications*. 2nd ed. Vol. 285. Springer, New York.

Durbin, J. and Koopman, S. J. (2012). *Time Series Analysis by State Space Methods*. 2nd ed. Vol. 38. Oxford Statistical Science Series. Oxford University Press, Oxford.

Paisley, J. et al. (2012). “Variational Bayesian Inference with Stochastic Search”. *Proceedings of the 29th International Conference on Machine Learning, ICML 2012* 2.

Sun, J. Z. et al. (2012). “A Framework for Bayesian Optimality of Psychophysical Laws”. *Journal of Mathematical Psychology* 56:6, pp. 495–501. DOI: 10.1016/j.jmp.2012.08.002.

Castro, M. et al. (2013). “A Spatial Generalized Ordered Response Model to Examine Highway Crash Injury Severity”. *Accident Analysis & Prevention* 52, pp. 188–203. DOI: 10.1016/j.aap.2012.12.009.

Chatfield, C. (2013). *The Analysis of Time Series: An Introduction, Sixth Edition*. Chapman and Hall/CRC, New York.

Donker, T. et al. (2013). “Smartphones for Smarter Delivery of Mental Health Programs: A Systematic Review”. *Journal of Medical Internet Research* 15:11, e247. DOI: 10.2196/jmir.2791.

Hoffman, M. D. et al. (2013). “Stochastic Variational Inference”. *Journal of Machine Learning Research* 14:4, pp. 1303–1347.

Likamwa, R. et al. (2013). “MoodScope: Building a Mood Sensor from Smartphone Usage Patterns”. In: *MobiSys 2013 - Proceedings of the 11th Annual International Conference on Mobile Systems, Applications, and Services*. DOI: 10.1145/2462456.2464449.

Varshney, L. R. and Sun, J. Z. (2013). “Why Do We Perceive Logarithmically?” *Significance* 10:1, pp. 28–31. DOI: 10.1111/j.1740-9713.2013.00636.x.

Gershman, S. and Goodman, N. (2014). “Amortized inference in probabilistic reasoning”. In: *Proceedings of the Annual Meeting of the Cognitive Science Society*. Vol. 36.

Kingma, D. P. and Welling, M. (2014). “Auto-Encoding Variational Bayes”. In: *2nd International Conference on Learning Representations, ICLR 2014*.

Mnih, A. and Gregor, K. (2014). “Neural Variational Inference and Learning in Belief Networks”. In: *Proceedings of the 31th International Conference on Machine Learning, ICML 2014*. Vol. 32. JMLR Workshop and Conference Proceedings. JMLR.org, pp. 1791–1799.

Rezende, D. J. et al. (2014). “Stochastic Backpropagation and Approximate Inference in Deep Generative Models”. In: *Proceedings of the 31st International Conference on Machine Learning*. Vol. 32. Proceedings of Machine Learning Research 2. PMLR, pp. 1278–1286.

Archer, E. et al. (2015). “Black Box Variational Inference for State Space Models”. *arXiv:1511.07367 [stat]*. arXiv: 1511.07367 [stat].

Canzian, L. and Musolesi, M. (2015). “Trajectories of depression: unobtrusive monitoring of depressive states by means of smartphone mobility traces analysis”. In: *Proceedings of the 2015 ACM international joint conference on pervasive and ubiquitous computing*, pp. 1293–1304.

Kingma, D. P. and Ba, J. (2015). “Adam: A Method for Stochastic Optimization”. In: *3rd International Conference on Learning Representations, ICLR 2015*.

Liu, J. N. et al. (2015). “Deep neural network modeling for big data weather forecasting”. In: *Information Granularity, Big Data, and Computational Intelligence*. Springer, pp. 389–408.

Moritz, S. et al. (2015). “Comparison of Different Methods for Univariate Time Series Imputation in R”. *arXiv:1510.03924 [cs, stat]*. arXiv: 1510.03924 [cs, stat].

Shaikhina, T. et al. (2015). “Machine Learning for Predictive Modelling based on Small Data in Biomedical Engineering”. *IFAC-PapersOnLine* 48:20, pp. 469–474. DOI: https://doi.org/10.1016/j.ifacol.2015.10.185.

Abdullah, S. et al. (2016). “Automatic Detection of Social Rhythms in Bipolar Disorder”. *Journal of the American Medical Informatics Association* 23:3, pp. 538–543. DOI: 10.1093/jamia/ocv200.

Cui, Z. et al. (2016). “Multi-Scale Convolutional Neural Networks for Time Series Classification”. *arXiv:1603.06995 [cs]*. arXiv: 1603.06995 [cs].

Eleftheriadis, S. et al. (2016). “Variational Gaussian Process Auto-Encoder for Ordinal Prediction of Facial Action Units”. In: *Computer Vision - ACCV 2016 - 13th Asian Conference on Computer Vision*. Vol. 10112. Lecture Notes in Computer Science, pp. 154–170. DOI: 10.1007/978-3-319-54184-6_10.

Faurholt-Jepsen, M. et al. (2016). “Electronic Self-Monitoring of Mood Using IT Platforms in Adult Patients with Bipolar Disorder: A Systematic Review of the Validity and Evidence”. *BMC Psychiatry* 16:1, p. 7. DOI: 10.1186/s12888-016-0713-0.

Gutiérrez, P. A. et al. (2016). “Ordinal Regression Methods: Survey and Experimental Study”. *IEEE Transactions on Knowledge and Data Engineering* 28:1, pp. 127–146. DOI: 10.1109/TKDE.2015.2457911.

Myin-Germeys, I. et al. (2016). “Ecological Momentary Interventions in Psychiatry”. *Current Opinion in Psychiatry* 29:4, pp. 258–263. DOI: 10.1097/YCO.0000000000000255.

Sathyanarayana, A. et al. (2016). “Sleep Quality Prediction From Wearable Data Using Deep Learning”. *JMIR mHealth and uHealth* 4:4, e125. DOI: 10.2196/mhealth.6562.

Weiss, K. et al. (2016). “A Survey of Transfer Learning”. *Journal of Big Data* 3:1, p. 9. DOI: 10.1186/s40537-016-0043-6.

Blei, D. M. et al. (2017). “Variational Inference: A Review for Statisticians”. *Journal of the American Statistical Association* 112:518, pp. 859–877. DOI: 10.1080/01621459.2017.1285773. arXiv: 1601.00670.

Chekroud, A. M. et al. (2017). “Computational Psychiatry: Embracing Uncertainty and Focusing on Individuals, Not Averages”. *Biological Psychiatry* 82:6, e45–e47. DOI: 10.1016/j.biopsych.2017.07.011.

Durstewitz, D. (2017a). “A state space approach for piecewise-linear recurrent neural networks for identifying computational dynamics from neural measurements”. *PLOS Computational Biology* 13:6, pp. 1–33. DOI: 10.1371/journal.pcbi.1005542.

Durstewitz, D. (2017b). *Advanced Data Analysis in Neuroscience*. Bernstein Series in Computational Neuroscience. Springer, Cham. DOI: 10.1007/978-3-319-59976-2.

Kim, S. (2017). "Ordinal Time Series Model for Forecasting Air Quality Index for Ozone in Southern California". *Environmental Modeling & Assessment* 22. DOI: 10.1007/s10666-016-9521-7.

Mikelsons, G. et al. (2017). "Towards Deep Learning Models for Psychological State Prediction Using Smartphone Data: Challenges and Opportunities". *arXiv:1711.06350 [cs, stat]*. arXiv: 1711.06350 [cs, stat].

Pedersen, A. et al. (2017). "Missing Data and Multiple Imputation in Clinical Epidemiological Research". *Clinical Epidemiology* Volume 9, pp. 157–166. DOI: 10.2147/CLEP.S129785.

Rodriguez, S. S. et al. (2017). "Mobile Sensing at the Service of Mental Well-being: a Large-scale Longitudinal Study". In: *Proceedings of the 26th International Conference on World Wide Web, WWW 2017*. ACM, pp. 103–112. DOI: 10.1145/3038912.3052618.

Shumway, R. H. and Stoffer, D. S. (2017). *Time Series Analysis and Its Applications: With R Examples*. Springer Texts in Statistics. Springer International Publishing, Cham. DOI: 10.1007/978-3-319-52452-8.

Suhara, Y. et al. (2017). "DeepMood: Forecasting Depressed Mood Based on Self-Reported Histories via Recurrent Neural Networks". In: *Proceedings of the 26th International Conference on World Wide Web, WWW 2017*. ACM, pp. 715–724. DOI: 10.1145/3038912.3052676.

Wen, C. K. F. et al. (2017). "Compliance With Mobile Ecological Momentary Assessment Protocols in Children and Adolescents: A Systematic Review and Meta-Analysis". *Journal of Medical Internet Research* 19:4, e132. DOI: 10.2196/jmir.6641.

Zhao, B. et al. (2017). "Convolutional Neural Networks for Time Series Classification". *Journal of Systems Engineering and Electronics* 28:1, pp. 162–169. DOI: 10.21629/JSEE.2017.01.18.

Blazquez, D. and Domenech, J. (2018). "Big Data Sources and Methods for Social and Economic Analyses". *Technological Forecasting and Social Change* 130, pp. 99–113. DOI: 10.1016/j.techfore.2017.07.027.

Che, Z. et al. (2018a). "Hierarchical Deep Generative Models for Multi-Rate Multivariate Time Series". In: *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*. Vol. 80. Proceedings of Machine Learning Research. PMLR, pp. 783–792.

Che, Z. et al. (2018b). "Recurrent Neural Networks for Multivariate Time Series with Missing Values". *Scientific Reports* 8:1, p. 6085. DOI: 10.1038/s41598-018-24271-9.

Chen, T. Q. et al. (2018). "Neural Ordinary Differential Equations". In: *Advances in Neural Information Processing Systems 31, NeurIPS 2018*.

Christensen, R. (2018). *Cumulative Link Models for Ordinal Regression with the R Package Ordinal*. Available at https://cran.r-project.org/web/packages/ordinal/vignettes/clm_article.pdf.

Cremer, C. et al. (2018). "Inference Suboptimality in Variational Autoencoders". In: *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*. Vol. 80. Proceedings of Machine Learning Research. PMLR, pp. 1086–1094.

Dwyer, D. B. et al. (2018). "Machine Learning Approaches for Clinical Psychology and Psychiatry". *Annual Review of Clinical Psychology* 14:1, pp. 91–118. DOI: 10.1146/annurev-clinpsy-032816-045037.

Guo, Z. et al. (2018). "A deep learning model for short-term power load and probability density forecasting". *Energy* 160, pp. 1186–1200. DOI: https://doi.org/10.1016/j.energy.2018.07.090.

Hirk, R. et al. (2018). "Multivariate Ordinal Regression Models: An Analysis of Corporate Credit Ratings". *Statistical Methods & Applications* 28, pp. 1–33. DOI: 10.1007/s10260-018-00437-7.

Hyndman, R. J. and Athanasopoulos, G. (2018). *Forecasting: Principles and Practice*. 3rd ed. OTexts: Melbourne, Australia. OTexts.com/fpp3.

Jaskari, J. and Kivinen, J. J. (2018). "A Novel Variational Autoencoder with Applications to Generative Modelling, Classification, and Ordinal Regression". *arXiv:1812.07352 [cs, stat]*. arXiv: 1812.07352 [cs, stat].

Liddell, T. M. and Kruschke, J. K. (2018). "Analyzing Ordinal Data with Metric Models: What Could Possibly Go Wrong?" *Journal of Experimental Social Psychology* 79, pp. 328–348. DOI: 10.1016/j.jesp.2018.08.009.

Myin-Germeys, I. et al. (2018). "Experience Sampling Methodology in Mental Health Research: New Insights and Technical Developments". *World Psychiatry* 17:2, pp. 123–132. DOI: 10.1002/wps.20513.

Wang, Y. et al. (2018). "Big Data Analytics: Understanding Its Capabilities and Potential Benefits for Healthcare Organizations". *Technological Forecasting and Social Change* 126, pp. 3–13. DOI: 10.1016/j.techfore.2015.12.019.

Zhang, Y. and Ling, C. (2018). "A Strategy to Apply Machine Learning to Small Datasets in Materials Science". *npj Computational Materials* 4:1, pp. 1–8. DOI: 10.1038/s41524-018-0081-z.

Belkin, M. et al. (2019). "Reconciling Modern Machine-Learning Practice and the Classical Bias–Variance Trade-Off". *Proceedings of the National Academy of Sciences* 116:32, pp. 15849–15854. DOI: 10.1073/pnas.1903070116.

Cearns, M. et al. (2019). "Recommendations and Future Directions for Supervised Machine Learning in Psychiatry". *Translational Psychiatry* 9:1, pp. 1–12. DOI: 10.1038/s41398-019-0607-2.

Colombo, D. et al. (2019). “Current State and Future Directions of Technology-Based Ecological Momentary Assessment and Intervention for Major Depressive Disorder: A Systematic Review”. *Journal of Clinical Medicine* 8:4, p. 465. DOI: 10.3390/jcm8040465.

Durstewitz, D. et al. (2019). “Deep Neural Networks in Psychiatry”. *Molecular Psychiatry* 24:11, pp. 1583–1598. DOI: 10.1038/s41380-019-0365-9.

Kim, J.-C. and Chung, K. (2019). “Prediction Model of User Physical Activity Using Data Characteristics-based Long Short-term Memory Recurrent Neural Networks”. *KSII Transactions on Internet and Information Systems* 13:4, pp. 2060–2077.

Kingma, D. P. and Welling, M. (2019). “An Introduction to Variational Autoencoders”. *Foundations and Trends in Machine Learning* 12:4, pp. 307–392. DOI: 10.1561/2200000056. arXiv: 1906.02691.

Koppe, G. et al. (2019a). “Identifying Nonlinear Dynamical Systems via Generative Recurrent Neural Networks with Applications to fMRI”. *PLOS Computational Biology* 15:8. Ed. by L. Isik, e1007263. DOI: 10.1371/journal.pcbi.1007263.

Koppe, G. et al. (2019b). “Recurrent Neural Networks in Mobile Sampling and Intervention”. *Schizophrenia Bulletin* 45:2, pp. 272–276. DOI: 10.1093/schbul/sby171.

Rubanova, Y. et al. (2019). “Latent ODEs for Irregularly-Sampled Time Series”. *arXiv:1907.03907 [cs, stat]*. arXiv: 1907.03907 [cs, stat].

Seppälä, J. et al. (2019). “Mobile Phone and Wearable Sensor-Based mHealth Approaches for Psychiatric Disorders and Symptoms: Systematic Review”. *JMIR mental health* 6:2, e9819. DOI: 10.2196/mental.9819.

Tran, T. D. et al. (2019). “Modeling local dependence in latent vector autoregressive models”. *Biostatistics* 22:1, pp. 148–163. DOI: 10.1093/biostatistics/kxz021.

Triantafillou, S. et al. (2019). “Relationship Between Sleep Quality and Mood: Ecological Momentary Assessment Study”. *JMIR Mental Health* 6:3, e12613. DOI: 10.2196/12613.

Umematsu, T. et al. (2019). “Improving Students’ Daily Life Stress Forecasting Using LSTM Neural Networks”. In: *2019 IEEE EMBS International Conference on Biomedical Health Informatics (BHI)*, pp. 1–4. DOI: 10.1109/BHI.2019.8834624.

Amigó, E. et al. (2020). “An Effectiveness Metric for Ordinal Classification: Formal Properties and Experimental Results”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*, pp. 3938–3949. DOI: 10.18653/v1/2020.acl-main.363.

Bärwolff, G. (2020). “Numerische Lösung stochastischer Differentialgleichungen”. In: *Numerik für Ingenieure, Physiker und Informatiker*. Springer, Berlin, Heidelberg, pp. 361–388. DOI: 10.1007/978-3-662-61734-2_10.

Brown, T. B. et al. (2020). “Language Models are Few-Shot Learners”. In: *Advances in Neural Information Processing Systems 33, NeurIPS 2020*.

Fortuin, V. et al. (2020). “GP-VAE: Deep Probabilistic Time Series Imputation”. In: *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020*. Vol. 108. Proceedings of Machine Learning Research. PMLR, pp. 1651–1661.

Keverne, J. and Binder, E. B. (2020). “A Review of Epigenetics in Psychiatry: Focus on Environmental Risk Factors”. *Medizinische Genetik* 32:1, pp. 57–64. DOI: 10.1515/medgen-2020-2004.

Li, S. C.-X. and Marlin, B. M. (2020). “Learning from Irregularly-Sampled Time Series: A Missing Data Perspective”. In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020*. Vol. 119. Proceedings of Machine Learning Research. PMLR, pp. 5937–5946.

Little, R. J. and Rubin, D. B. (2020). *Statistical analysis with missing data*. John Wiley & Sons.

Monfared, Z. and Durstewitz, D. (2020a). “Existence of N-Cycles and Border-Collision Bifurcations in Piecewise-Linear Continuous Maps with Applications to Recurrent Neural Networks”. *Nonlinear Dynamics* 101:2, pp. 1037–1052. DOI: 10.1007/s11071-020-05841-x.

Monfared, Z. and Durstewitz, D. (2020b). “Transformation of ReLU-based Recurrent Neural Networks from Discrete-Time to Continuous-Time”. In: *Proceedings of the 37th International Conference on Machine Learning*. PMLR, pp. 6999–7009.

Nazábal, A. et al. (2020). “Handling incomplete heterogeneous data using VAEs”. *Pattern Recognit.* 107, p. 107501. DOI: 10.1016/j.patcog.2020.107501.

Sayer, R. (2020). “Bayesian Variational Inference for Piecewise-Linear Recurrent Neural Networks”. MA thesis. Heidelberg: University of Heidelberg.

Sezer, O. B. et al. (2020). “Financial Time Series Forecasting with Deep Learning : A Systematic Literature Review: 2005–2019”. *Applied Soft Computing* 90, p. 106181. DOI: 10.1016/j.asoc.2020.106181.

Thieme, A. et al. (2020). “Machine Learning in Mental Health: A Systematic Review of the HCI Literature to Support the Development of Effective and Implementable ML Systems”. *ACM Transactions on Computer-Human Interaction* 27:5, 34:1–34:53. DOI: 10.1145/3398069.

Wang, L. et al. (2020). “Sex Trafficking Detection with Ordinal Regression Neural Networks”. *arXiv:1908.05434 [cs, stat]*. arXiv: 1908.05434 [cs, stat].

Bommer, P. L. et al. (2021). “Identifying Nonlinear Dynamical Systems from Multi-Modal Time Series Data”. *arXiv:2111.02922 [cs, q-bio, stat]*. arXiv: 2111.02922 [cs, q-bio, stat].

Brenner, M. et al. (2021). *Tractable Dendritic RNNs for Identifying Unknown Nonlinear Dynamical Systems*. Available at https://openreview.net/forum?id=AVShGWiL9z.

Dai, Z. et al. (2021). “CoAtNet: Marrying Convolution and Attention for All Data Sizes”. *arXiv:2106.04803 [cs]*. arXiv: 2106.04803 [cs].

Durstewitz, D. et al. (2021). "Psychiatric Illnesses as Disorders of Network Dynamics". *Biological Psychiatry: Cognitive Neuroscience and Neuroimaging* 6:9, pp. 865–876. DOI: 10.1016/j.bpsc.2020.01.001.

Girin, L. et al. (2021). "Dynamical Variational Autoencoders: A Comprehensive Review". *Foundations and Trends in Machine Learning* 15:1-2, pp. 1–175. DOI: 10.1561/2200000089. arXiv: 2008.12595.

Koppe, G. et al. (2021). "Deep Learning for Small and Big Data in Psychiatry". *Neuropsychopharmacology* 46:1, pp. 176–190. DOI: 10.1038/s41386-020-0767-z.

Lim, B. and Zohren, S. (2021). "Time-Series Forecasting with Deep Learning: A Survey". *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 379:2194, p. 20200209. DOI: 10.1098/rsta.2020.0209.

Monfared, Z. et al. (2021). "How to Train RNNs on Chaotic Data?" *arXiv:2110.07238 [cs, math, stat]*. arXiv: 2110.07238 [cs, math, stat].

Nakkiran, P. et al. (2021). "Deep Double Descent: Where Bigger Models and More Data Hurt". *Journal of Statistical Mechanics: Theory and Experiment* 2021:12, p. 124003. DOI: 10.1088/1742-5468/ac3a74.

Ortiz, A. et al. (2021). "The Futility of Long-Term Predictions in Bipolar Disorder: Mood Fluctuations Are the Result of Deterministic Chaotic Processes". *International Journal of Bipolar Disorders* 9:1, p. 30. DOI: 10.1186/s40345-021-00235-3.

Rauschenberg, C. et al. (2021a). "Living Lab AI4U - Artificial Intelligence for Personalized Digital Mental Health Promotion and Prevention in Youth". *European Journal of Public Health* 31:Supplement_3, ckab164.746. DOI: 10.1093/eurpub/ckab164.746.

Rauschenberg, C. et al. (2021b). "A Compassion-Focused Ecological Momentary Intervention for Enhancing Resilience in Help-Seeking Youth: Uncontrolled Pilot Study". *JMIR Mental Health* 8:8, e25650. DOI: 10.2196/25650.

Sakai, T. (2021). "Evaluating Evaluation Measures for Ordinal Classification and Ordinal Quantification". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021*, pp. 2759–2769. DOI: 10.18653/v1/2021.acl-long.214.

Schick, A. et al. (2021). "Effects of a Novel, Transdiagnostic, Hybrid Ecological Momentary Intervention for Improving Resilience in Youth (EMIcompass): Protocol for an Exploratory Randomized Controlled Trial". *JMIR research protocols* 10:12, e27462. DOI: 10.2196/27462.

Schmidt, D. et al. (2021). "Identifying nonlinear dynamical systems with multiple time scales and long-range dependencies". In: *9th International Conference on Learning Representations, ICLR 2021*.

Shi, Y. et al. (2021). “Relating by Contrasting: A Data-efficient Framework for Multimodal Generative Models”. In: *9th International Conference on Learning Representations, ICLR 2021*.

Sükei, E. et al. (2021). “Predicting Emotional States Using Behavioral Markers Derived From Passively Sensed Data: Data-Driven Machine Learning Approach”. *JMIR mHealth and uHealth* 9:3, e24465. DOI: 10.2196/24465.

Tombolini, C. (2021). “Non-Linear Dynamical System Identification from Multimodal Time Series”. MA thesis. Heidelberg: University of Heidelberg.

Warkentin, P. A. (2021). “Oscillatory Pre-Training for the Reconstruction of Dynamical Systems in Recurrent Neural Networks”. MA thesis. Heidelberg: University of Heidelberg.

Zhang, C. et al. (2021). “Understanding Deep Learning (Still) Requires Rethinking Generalization”. *Communications of the ACM* 64:3, pp. 107–115. DOI: 10.1145/3446776.

Lu, F. et al. (2022). “Continuously Generalized Ordinal Regression for Linear and Deep Models”. *arXiv:2202.07005 [cs]*. arXiv: 2202.07005 [cs].

Appendix

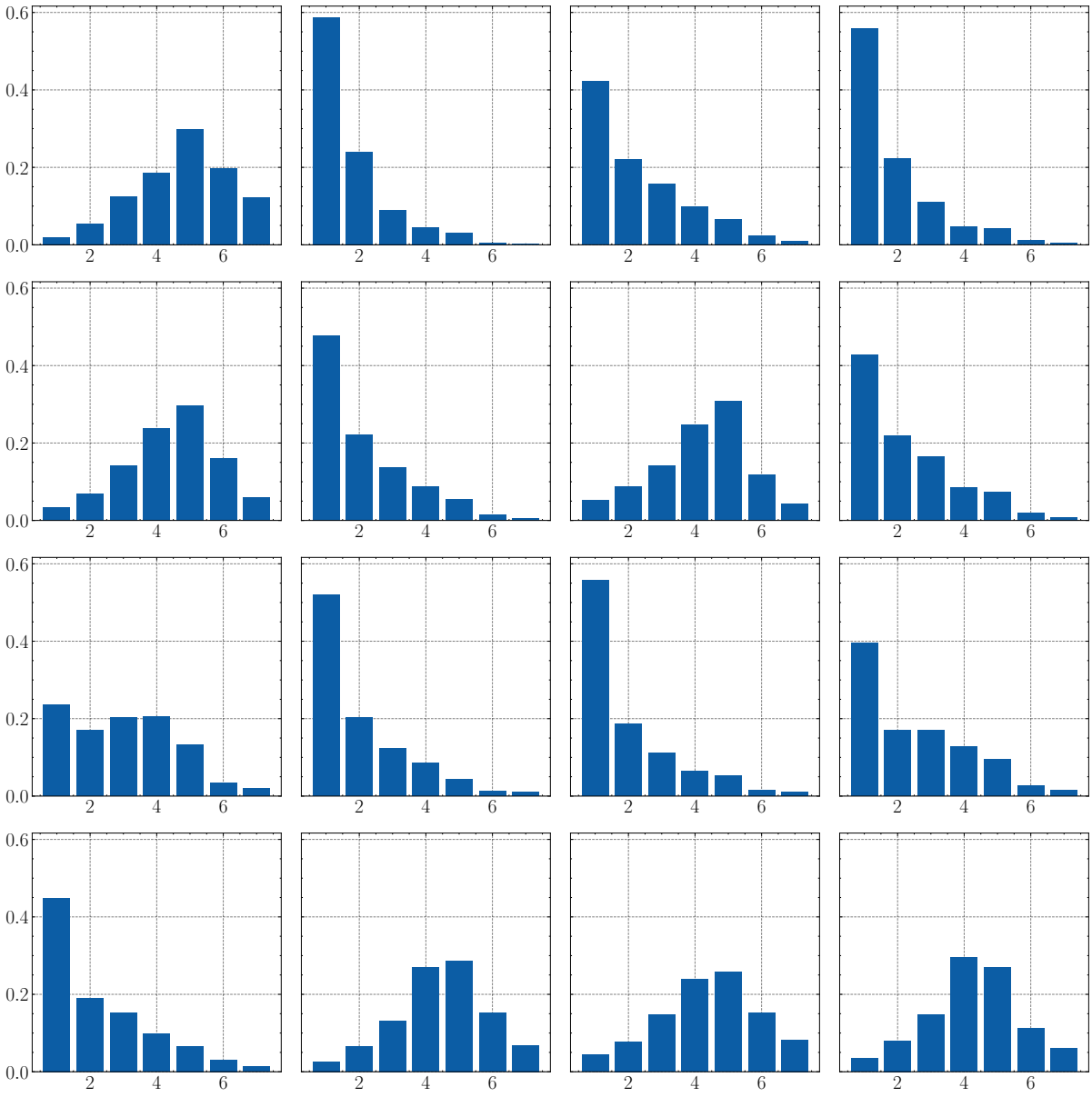


Figure 26: The histograms show that the overall distribution of the ordinal features in the benchmark data perfectly matches with the empirically observed distributions of the Likert items in the EMIcompass data, see Figure 9.

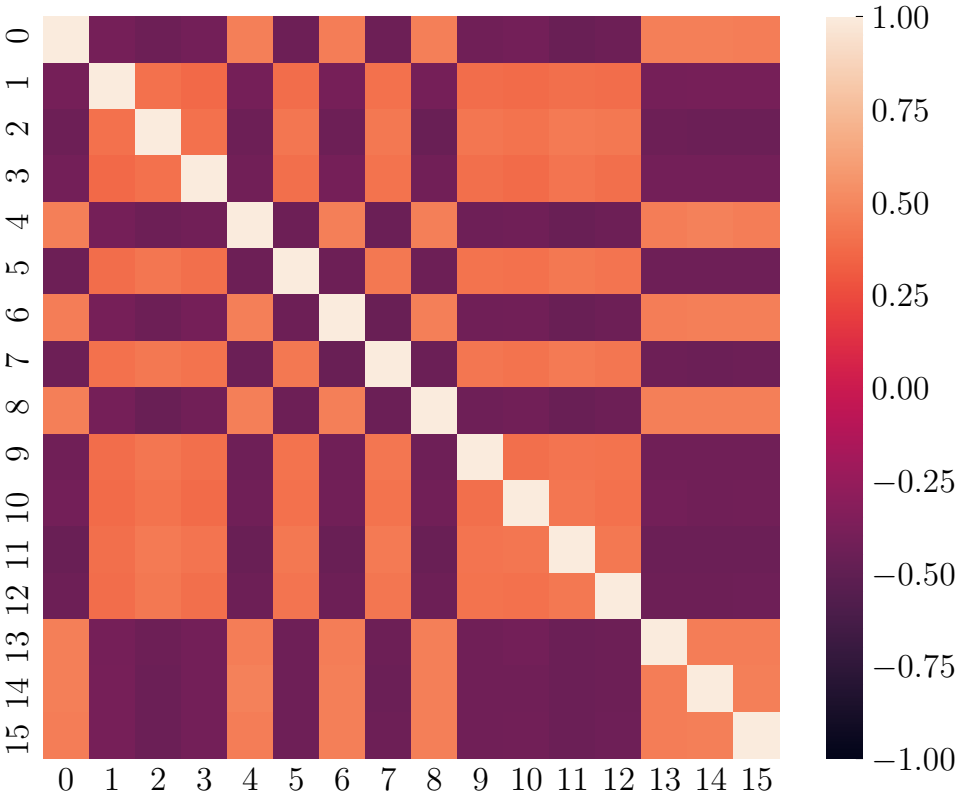


Figure 27: The Spearman rank order correlation matrix of the benchmark data shows good agreement with the correlation structure extracted from the EMIcompass data, see Figure 12.

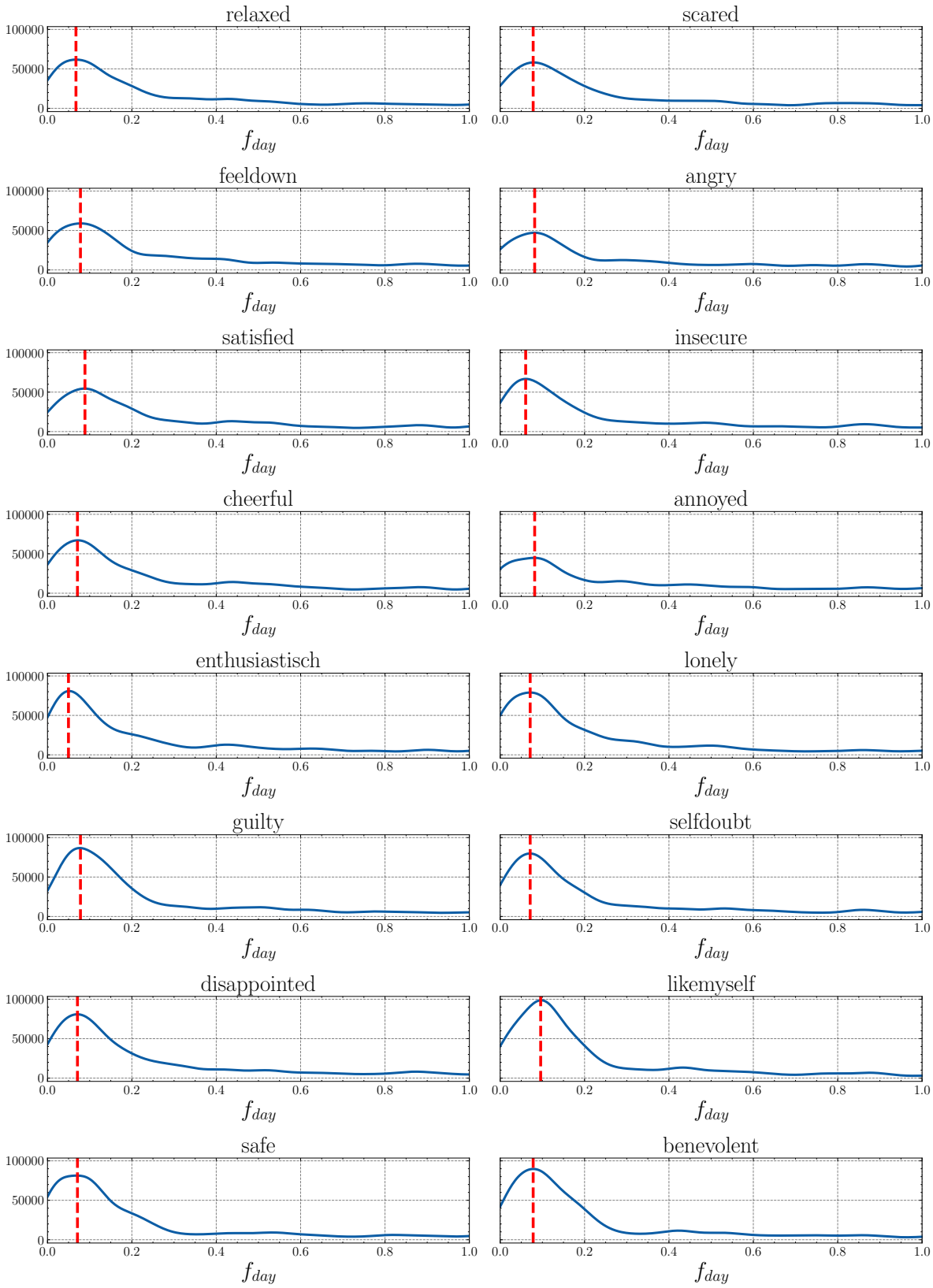
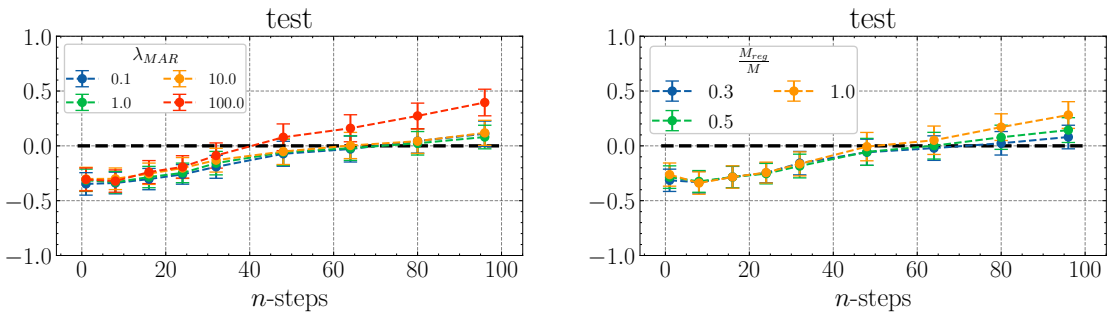


Figure 28: Power spectrum of the EMIcompass data smoothed by a Gaussian kernel. The dotted red line indicates the maximum frequency component.



(a) $\text{RMSE}_n^{\text{model}} - \text{RMSE}_n^{\text{mean}}$ **(b)** $\text{RMSE}_n^{\text{model}} - \text{RMSE}_n^{\text{mean}}$

Figure 29: Model performance on benchmark data ($T_{\text{train}} = 10.000$) for different MAR settings. *Left*: $\frac{M_{\text{MAR}}}{M} = 0.3$, *Right*: $\lambda_{\text{MAR}} = 1.0$.

Erklärung

Ich versichere, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Heidelberg, 27.03.2022
(Ort, Datum)

Unai Fischer Abaigar
(Unai Fischer Abaigar)