Modelling Canonical Computations in Brains and Machines with the Free Energy Principle

DISSERTATION

zur Erlangung des akademischen Grades

Doktoringenieur (Dr.-Ing.)

angenommen durch die Fakultät für Informatik
der Otto-von-Guericke-Universität Magdeburg

von M.Sc. André Ofner

geb. am 10.09.1993 in München

Gutachterinnen/Gutachter

Prof. Dr. Sebastian Stober
Prof. Dr. Christopher L. Buckley
Dr. Arthur Flexer

Magdeburg, den 20.07.2023

ANDRÉ OFNER

MODELLING CANONICAL COMPUTATIONS IN BRAINS AND MACHINES WITH THE FREE ENERGY PRINCIPLE

Dissertation

Supervisors
Prof. Dr.-Ing. Sebastian Stober
Prof. Dr. rer. nat. habil. Myra Spiliopoulou

André Ofner: *Modelling Canonical Computations in Brains and Machines with the Free Energy Principle,* Dissertation

ABSTRACT

The Free Energy Principle in neuroscience explains brain function based on the minimization of Bayesian surprise. Under mathematical constrains on the underlying optimisation process, the FEP can be used to derive a process theory that describes neuronal dynamics as a hierarchy of descending local predictions and ascending prediction errors that drives learning of an internal generative model of the world. This inversion of a hierarchical generative model in the brain is known as predictive coding (PC). Simultaneously, the minimization of Bayesian surprise is a central objective for generative models in the context of deep artificial neural networks (ANNs), such as the Variational Autoencoder (VAE). Deep generative models are trained on high-dimensional inputs from large datasets using exact backpropagation of errors to the parameters of the model. Such global and exact optimisation is in contrast to the approximate, iterative and locally informed learning in PC. Apart from similar computational objectives it is still unclear how modelling constrains under the FEP, such as locally optimised free energy or the use of error uncertainty relate to deep generative models in terms of performance and comparability to human brain function.

In this thesis, we aim at filling some of these gaps by designing and evaluating ANN models under the constrains of the FEP and hypothesize that such models implement canonical computations that are present across scales in the human brain.

We contribute a dynamical PC model with a hierarchy of latent representations that learns by predicting probabilistic sequences of latent states using exact error backpropagation. We also contribute a generalized PC model (GPC) that is designed from first principles under the FEP. GPC replaces global backpropagation of error with local optimisation, performs uncertainty estimation under the Laplace approximation and generates dynamical predictions local in time using generalized coordinates of motion. We demonstrate that GPC scales to complex sensory inputs and is comparable to VAEs in terms of inference.

Next to evaluating unsupervised learning, we test and discuss possibilities to retrieve sensory processing related information in brain activity using FEP based models. In this context, we evaluate the possibility to learn shared representations of brain activity recorded in electroencephalograms (EEG) and auditory stimuli using a multiview VAE architecture. We demonstrate the possibility to retrieve temporal locations of evoked brain responses (ERPs) in EEG from the error response of a PC model processing audio stimuli. We also

demonstrate the possibility to predict EEG signals directly and discuss options to actively infer temporal ERP locations from EEG.

With our proposed methods we take a step towards performant, yet biologically plausible, ANNs and provide means of explainability through comparison to human brain function and model design from first principles.

ZUSAMMENFASSUNG

Das Free Energy Principle (FEP) in den Neurowissenschaften erklärt Gehirnfunktion basierend auf der Minimierung von Überraschung im bayesschen Sinne. Unter mathematischen Einschränkungen des zugrunde liegenden Optimierungsprozesses kann das FEP verwendet werden, um eine Prozesstheorie abzuleiten, die neuronal Interaktionen als eine Hierarchie von absteigenden lokalen Vorhersagen und aufsteigenden Vorhersagefehlern beschreibt, die zum Erlernen eines internen generativen Modells der Welt dient.

Diese Invertierung eines hierarchischen generativen Modells im Gehirn ist auch als Predictive Coding (PC) bekannt. Gleichzeitig ist die Minimierung von bayesscher Überraschung ein zentrales Berechnungsziel für generative Modelle im Zusammenhang mit tiefen künstlichen neuronalen Netzen (KNNs), wie dem Variational Autoencoder (VAE). Tiefe generative Modelle werden mit komplexen Eingaben aus großen Datensätzen trainiert und nutzen eine exakte Rückpropagierung von Vorhersagefehlern zu Modellparametern.

Eine solche globale und exakte Optimierung steht im Kontrast zum approximativen, iterativen und lokalen Lernen in PC Modellen. Abgesehen vom ähnlichen globalen Ziel der Modelloptimierung ist noch unklar, wie Modellierungsbeschränkungen unter dem FEP, wie zum Beispiel lokal optimierte freie Energie oder die Verwendung von Fehlerunsicherheit, mit tiefen generativen Modellen in Bezug auf Leistung und Vergleichbarkeit mit der menschlichen Gehirnfunktion zusammenhängen.

In dieser Dissertation zielen wir darauf ab, einige dieser Lücken zu schließen, indem wir KNNs unter den Einschränkungen des FEP entwerfen, trainieren und evaluieren. Dabei gehen wir von der Hypothese aus, dass solche Modelle kanonische Berechnungen implementieren, die über verschiedenen Skalen verteilt auch im menschlichen Gehirn vorhanden sind.

Wir stellen ein dynamisches PC Modell mit einer Hierarchie latenter Repräsentationen vor, das lernt, indem es Sequenzen von probabilistischen Zuständen unter Verwendung exakter Fehlerrückpropagierung vorhersagt. Wir stellen weiterhin ein Generalized PC Modell (GPC) vor, das von Grund auf unter dem FEP entwickelt wurde. GPC ersetzt die globale Fehlerrückpropagierung durch lokale Optimierung, führt Unsicherheitsschätzung unter der Laplace Annäherung durch und generiert strikt zeitlich lokale dynamische Vorhersagen mithilfe von generalisierten Koordinaten. Wir zeigen, dass GPC mit komplexen statischen und sequentiellen Informationen umgehen kann und in Bezug auf Inferenz mit VAEs vergleichbar ist.

Neben dem unüberwachten Lernen testen und diskutieren wir Möglichkeiten, FEP-basierte Modelle im Sinne des Information Retrieval zum Auffinden von wahrnehmungsbezogener Informationen im menschlichen Gehirn zu nutzen. In diesem Zusammenhang evaluieren wir die Möglichkeit, gemeinsame Repräsentationen von mittels Elektroenzephalografie (EEG) aufgezeichneter Gehirnaktivität und auditiven Stimuli unter Verwendung einer Multi-View-VAE Architektur zu lernen. Wir demonstrieren die Möglichkeit, zeitliche Positionen von evoked potentials (ERPs), ereignisbezogene Potentialen im EEG, aus der Fehlerantwort eines PC-Modells abzurufen, das Audiostimuli verarbeitet. Wir demonstrieren weiterhin die Möglichkeit, EEG Signale direkt vorherzusagen und diskutieren Möglichkeiten, die zeitliche Position von ERPs aktiv aus EEG-Vorhersagefehlern abzuleiten.

Mit unseren vorgeschlagenen Methoden machen wir einen Schritt in Richtung leistungsfähiger, aber biologisch plausibler KNNs und bieten Mittel zur Erklärbarkeit durch den Vergleich mit menschlicher Gehirnfunktion und der Modellentwicklung auf Basis fundamentaler Prinzipien.

PUBLICATIONS

Many ideas and figures presented in this thesis have appeared previously in the following conference publications and a book chapter:

[pub:1] A. Ofner, B. Millidge, and S. Stober. "Generalized Predictive Coding: Bayesian Inference in Static and Dynamic models." In: *4th Shared Visual Representations in Human and Machine Intelligence workshop (SVRHM) at NeurIPS*. 2022.

[pub:2] A. Ofner, J. Schleiss, and S. Stober. "Hierarchical Predictive Coding and Interpretable Audio Analysis-Synthesis." In: *Proceedings of the 15th International Symposium on Computer Music Multidisciplinary Research (CMMR)*. 2021, pp. 225–234.

[pub:3] A. Ofner and S. Stober. "Shared Generative Representation of Auditory Concepts and EEG to Reconstruct Perceived and Imagined Music." In: *19th International Society for Music Information Retrieval Conference (ISMIR)*. 2018, pp. 392–399.

[pub:4] A. Ofner and S. Stober. "Balancing Active Inference and Active Learning with Deep Variational Predictive Coding for EEG." In: *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE. 2020, pp. 3839–3844.

[pub:5] A. Ofner and S. Stober. "Modeling perception with hierarchical prediction: Auditory segmentation with deep predictive coding locates candidate evoked potentials in EEG." In: *21st International Society for Music Information Retrieval Conference (ISMIR)*. 2020, pp. 566–573.

[pub:6] A. Ofner and S. Stober. "Deep Neural Networks and Auditory Imagery." In: *Music and Mental Imagery*. Ed. by M. B. Küssner, L. Taruffi, and G. A. Floridou. Routledge, 2022, pp. 112–122.

[pub:7] R. P. Rane, E. Szügyi, V. Saxena, A. Ofner, and S. Stober. "Prednet and predictive coding: A critical review." In: *Proceedings of the 2020 International Conference on Multimedia Retrieval (ICMR)*. 2020, pp. 233–241.

Several additional publications were created during the course of writing, but are not directly used in this thesis. They are briefly summarized in Section A.1 of the Appendix.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

FEP Free Energy Principle

PC predictive coding

VFE variational free energy

ELBO evidence lower bound

GPC generalized predictive coding

BPTT backpropagation through time
VAE variational autoencoder
DNN deep neural network
NN neural network
RNN recurrent neural network
CNN convolutional neural network
VCCA Variational Canonical Correlation Analysis
MIR Music Information Retrieval
EEG electroencephalogram
fMRI functional magnetic resonance imaging
ICA independent component analysis
CCA Canonical Correlation Analysis
ReLU rectified linear unit
SSIM structural similarity index measure
KL Kullback-Leibler
LPC Linear Predictive Coding
EOG electrooculography
DVCCA Deep Variational Canonical Correlation Analysis
VLAE Variational Laplace Autoencoder
MS multiple shooting
MAP maximum a posteriori probability
PCN predictive coding network
LA Laplace approximation
LSTM long short-term memory
ML machine learning
FFT fast Fourier transform
DCCAE Deep Canonically Correlated Autoencoders
BCCA Bayesian Canonical Correlation Analysis
MSE mean squared error
FRP fixation related potential
SSEP steady-state evoked potentials
ERP event-related potentials
MIR Music Information Retrieval
PCA Principal Components Analysis
VRNN Variational Recurrent Neural Network
RSSM recurrent state space models

CPC Contrastive Predictive Coding

SSM state space model

IIR Infinite Impulse Response

DSP Digital Signal Processing

ODE Ordinary Differential Equation

CorrNet Correlational Neural Network

ConvLSTM convolutional LSTM

ANN artificial neural network

EM expectation-maximisation

MAP maximum a posteriori

1 INTRODUCTION

1.1 MOTIVATION AND SCOPE

The Free Energy Principle (FEP) in neuroscience provides a unifying theoretical account of brain function [50, 52, 62]. At its core, it states that the human brain, and self-organising systems more generally, constantly minimize Bayesian surprise [50, 91].

As a principle, the FEP itself is not a theory and does not provide empirically falsifiable claims per se [62]. When no further assumptions on the implementation are made, the FEP essentially describes brain function as making exact Bayesian inference, in line with the Bayesian brain hypothesis [62].

The FEP, however, can be used to derive specific process theories when assumptions are made regarding the structure of the underlying probabilistic model and optimization scheme [62]. Such process theories specify concrete and testable process models, that allow to falsify theoretical claims using empirical experimentation. A particular influential instantiation of the FEP is predictive coding (PC). PC rests on the assumption that the brain optimises, via gradient descent on prediction errors, a hierarchically structured generative model that encodes factorized and Gaussian distributions, typically using a mean-field and Laplace approximation [49, 51, 172]. This effectively turns exact perception as Bayesian inference into approximate inference based on the optimisation of variational free energy. Such free energy minimisation is a candidate for canonical computations in the brain, as the same set of computations is hypothesized to be repeated across brain regions, across scales and between different modalities [11, 79]. In this context, free energy minimisation via PC has generated large quantities of empirical evidence, especially in the context of a mapping onto canonical cortical microcircuits in the brain [5, 11, 146, 193, 215].

CURRENT MODELS AND THEIR LIMITATIONS Variational free energy can be interpreted as the negative of the evidence lower bound (ELBO), which is a common objective function for many generative models in machine learning [12, 104]. As such, the central objective underlying perceptual inference in the FEP, the variational free energy, is already a central component of many highly influential deep learning models that perform variational inference on static observations, maybe most prominently the variational autoencoder (VAE) [104] or deep recurrent latent variable models that infer sequential data [20].

These high performing models resort to deep neural networks (DNNs) trained via backpropagation of a global error signal. Backpropagation involves computing the gradient of the chosen objective function with respect to the optimised weights using the chain rule [177]. While DNNs per se are inspired by biological neurons and neural connectivity, the biological plausibility of training multi-layer networks using globally informed and exact error signals is highly debated [25, 120, 174]. Nevertheless, the computational goal in generative DNN based models, such as VAEs, and more directly biologically plausible process models, like PC, is directly comparable [128]. When searching for canonical patterns of computation that relate process models to brain function, it can be useful to separate the computational objective (minimizing free energy) from the algorithmical or implementational details and analyse their mutual influence, in spirit of Marr's levels of analysis [130]. This thesis focuses on the interplay between algorithmic constrains posed on predictive coding networks (PCNs) under the FEP and their influence on the overall computation in DNNs as well as the mapping to human brain signals.

Research on process models under the FEP is a highly active field in neuroscience, where elaborate and biologically plausible models, such as generalized predictive coding, have been developed that describe learning in hierarchical-dynamical architectures [11, 51]. Such models in neuroscience often, however, are evaluated on data with relatively low complexity. Influential process models, such as hierarchical PC do not directly scale to machine learning applications, such as unsupervised learning from complex image or video observations [139]. In contrast, their DNN based counterparts in machine learning, e.g. Contrastive Predictive Coding (CPC) or PredNet, are often only loosely inspired by computational or algorithmic aspects of process models in neuroscience [84, 124, 139, 156]. Often times, this results in architectures that are difficult to interpret with respect to their role as a model of canonical computations in the brain. Only little research has focused explicitly on developing process models derived from the FEP that scale to tasks usually approached with DNNs and error backpropagation, such as gradient-based predictive coding [208, 225]. These models are still substantially less elaborate than process models in neuroscience, and still have to scale to complex sequential or spatio-temporal inputs, like audio or video. Similarly, they often still lack aspects of uncertainty estimation and hierarchical-dynamical abstraction, which are central components of the respective process models in neuroscience.

In this thesis, we aim at filling some of these gaps between free energy optimization in neuroscientific and machine learning models by designing and evaluating ANN models under the constrains of the FEP. The design and evaluation of the proposed models is based on the following hypotheses that underlie the presented research:

Research hypotheses

1. PC, and in particular Generalized PC under the FEP, is a candidate canonical computational mechanism for self-organization in brains and machines
2. Biological and artificial PCNs can be directly related with respect to their functional architecture and prediction error responses during inference
3. Biologically plausible PCNs in machine learning can process complex signals to enable information retrieval from sensory, brain and cross-modal data

1.1.1 *Research aims and contributions*

Based on aforementioned general assumptions, we focus on the following core research aims:

Research aims

1. Design and evaluate DNN based PCNs with coherence to the Free Energy Principle
2. Characterize PCN responses in comparison to human brain activity and human annotation of sensory data
3. Design PCNs with local learning rules and evaluate their performance on unsupervised representation learning and in relation to DNNs with exact backpropagation of errors

In this context, we introduce novel PCN models and perform qualitative and quantitative evaluation from several perspectives:

1. We contribute a dynamical deep PC model with a hierarchy of latent representations that learns by predicting probabilistic sequences of latent states using exact error backpropagation.
2. We test and discuss possibilities to retrieve sensory processing related information in brain activity using FEP based models. In this context, we evaluate the possibility to learn shared representations of brain activity recorded in electroencephalograms and auditory stimuli using a multi-view VAE architecture.
3. We investigate the possibility to retrieve temporal locations of evoked brain responses in EEG from the error response of the proposed dynamical deep PC model in the context of audio stimulus processing.

4. We further demonstrate the possibility to predict EEG signals directly using the proposed model and discuss options to actively infer temporal evoked response locations from EEG.

5. We contribute a generalized PC model that is designed from first principles under the FEP. The model uses locally informed error optimisation, performs uncertainty estimation under the Laplace approximation and generates dynamical predictions local in time using generalized coordinates of motion. We demonstrate that GPC scales to complex sensory inputs and is comparable to VAEs in terms of inference.

1.1.2 *Thesis structure*

thesis is structured into eight chapters that cover context, proposed models, conducted experiments and discussion of empirical results:

- Chapter 2 discusses the context and fundamental mathematical background for this thesis.
- Chapter 3 presents a VAE based architecture for shared representation learning from electroencephalogram (EEG) and auditory data.
- Chapter 4 analyses and reviews PredNet, a popular DNN model that considers ideas from predictive coding.
- Chapter 5 introduces a hierarchical deep PC network that learns by predicting probabilistic sequences of latent states and covers its application to information retrieval from EEG data.
- Chapter 6 presents and evaluates a variant of gradient-based PC with local learning rules for the prediction of audio.
- Chapter 7 presents a generalized PC network and evaluates the performance and scalability of a biologically plausible implementation of the FEP.
- Chapter 8 recapitulates the contributions of thesis and discusses limitations and potential future research directions.

2

PRELIMINARIES

The following sections will briefly introduce the Bayesian brain hypothesis and compare its central claims about perceptual processing with those posed by the FEP. After clarifying the overall computational objectives, this chapter covers the mathematical preliminaries for describing PC models under the FEP, including variational inference, the mean-field and the Laplace approximation. Finally, we will relate PC models to PC inspired deep neural networks and cover possibilities to relate FEP based models to electroencephalographic recordings of brain signals.

2.1 FREE ENERGY PRINCIPLE AND PREDICTIVE CODING

2.1.1 *The Bayesian brain hypothesis*

From a high-level perspective, the FEP can be related to the Bayesian brain hypothesis [38, 91, 106]. In particular, without specific assumptions on the structure and type of optimisation proces of the underlying generative model, the FEP is even equivalent to the Bayesian brain hypothesis [62].

The Bayesian brain hypothesis makes several fundamental claims about brain function: Firstly, it claims that the brain maintains an internal, generative model of the world that specifies how observations are generated. Typically, such a generative model explains sensations stemming from the physical environment of the agent, including the behavior of intelligent agents, but can cover internal, bodily, sensations as well [15, 188]. In this context it is important to note that the Bayesian brain hypothesis does not make claims about the exact type of encoding of this generative model, or whether the generative model is even explicitly encoded at all. Instead, the central claim is that the brain operates "as if" it had an internal model [62].

Secondly, following the Bayesian brain hypothesis, the internal generative model can be structured into two types of variables that model unobservable "hidden" states and observable sensory states respectively. Hidden variables ν have a prior distribution $p(\nu)$ that can be drawn from. For example, a concrete hidden state might refer to the size of an object and its prior distribution $p(\nu)$ could cover a range of possible sizes. The second type of variable, sensory observations o are drawn from a distribution of observations that is conditional on this hidden state $p(o|\nu)$. This means that the internal generative model is equipped with the capacity to express a specific hypothesis

about a hidden state $p(v)$ and to map it to concrete sensory observations o that are possible given this hypothesis. This allows to compute the likelihood of the observation under the current hypothesis, i.e. a quantification of how likely it is to encounter a specific observation under the assumption that a chosen hypothesis is correct [91]. Typically, the mapping between hidden and observed state is assumed to be complex, noisy or even ambiguous, such that multiple hypotheses could explain the same observation [62, 91]. Following Bayes' rule, the posterior distribution $p(v \mid o)$ of a specific hypothesis given an observation can be computed by inverting the structure of the generative model:

$$p(v \mid o) = \frac{p(o \mid v)p(v)}{p(o)} \tag{1}$$

which depends on the hidden state prior distribution $p(v)$, the conditional distribution of observations given hidden states $p(o|v)$ and the marginal likelihood of observations $p(o) = \sum_v p(o \mid v)p(v)$. If continuous states are inferred, summation turns into integration. Thus in perception, under the Bayesian brain hypothesis, the brain updates beliefs in its hypotheses based on the inference of posterior distributions (after observing new data) given their prior distributions (before observing new data) using Bayes' rule [91].

Such belief updates require new data, i.e. observations, to be acquired. Depending on the complexity of modelled agent, new data from the environment can be treated as strictly passively observed or influenced by the agent's own actions a, or sequences of actions π. In some models, new data might also be generated entirely internally, e.g. during imagination or sleep [57].

2.1.2 *Bayesian surprise*

Focusing on passive perception, we can now talk about a "surprise" about newly encountered sensory observations o in the Bayesian sense: A new observation brings Bayesian surprise when the posterior distribution after the observation is different to the prior [91]. The quantity of surprise, the distance between prior and posterior distribution, is often expressed using the Kullback-Leibler (KL) divergence that measures the relative entropy between the distributions:

$$D_{KL}(p(v \mid o) \| p(v)) = \sum_v p(v \mid o) \log \frac{p(v \mid o)}{p(v)} dv \tag{2}$$

This thesis is concerned primarily with modelling passive perception, in the absence of explicit actions or policies over actions. However, for the sake of of completeness, we briefly talk about the role

of active intervention on Bayesian surprise. When actions and policies π (referring to sequences over actions) are considered, Bayesian surprise is equivalent to the information gain $\mathcal{I}(\pi)$ from following a particular policy:

$$\mathcal{I}(\pi) = \sum_{o} p(o \mid \pi) D_{KL}(p(v \mid o, \pi) \| p(v \mid \pi)) \tag{3}$$

where the KL divergence $D_{KL}(p(v \mid o, \pi) \| p(v \mid \pi))$ now measures the divergence of the posterior distribution after following a policy from the prior distribution before taking any action [62]. The mutual influence between the expected surprise of observations encountered in the future when following a policy and and the minimization of long-term surprise in perception is the starting point for various models in Bayesian decision theory. A prominent example is information gain maximization that treats the utility of a possible policy π simply as equivalent to the information gain $\mathcal{I}(\pi)$. Finally, expected observations under a particular policy can be modelled as leading to positive or negative rewards, such that the expected utility becomes a function that combines exploitation, i.e. taking actions that lead to reward, with exploration, i.e. taking actions that maximise information gain [23, 200].

2.1.3 *Free Energy Principle and variational inference*

The central statement of the FEP with respect to self organising systems is:

The Free Energy Principle and self-organisation

The Free Energy Principle states that self-organising systems minimize a quantity called free energy when they are at an equilibrium with their environment [50, 105].

Biological agents (or the brain in particular) are embedded in an environment that constantly changes, but only have a limited set of *internal* states. This implies that agents must avoid (the long-term average of) surprise about the *external* in order to maintain the stability of their internal states. From the perspective of the brain, the external consists of the human body as well as the physical environment. As we will discuss in this section, the free energy is an upper bound on surprise and is computed with respect to those quantities that the agent can control, namely its internal states and the actions it generates. The influence of the internal on the external is mediated via actions, while sensations depend only on the external and drive changes of the internal representations in order to provide better explanations. Figure

1 schematically displays these structural dependencies between the brain and its environment.

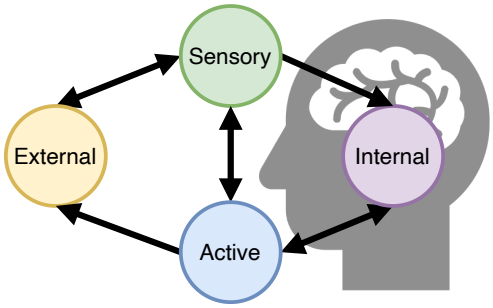


Figure 1: Structural dependencies that separate the *internal* states of the brain from the *external* environment under the FEP. Internal states are in exchange with the environment through *action* and *sensations*. Internal (brain) states and action minimize free energy, which is a function of sensory observations and the represented generative model that explains the causes of observations.

A Markov blanket describes such separation of internal and external states in a self-organising system in the statistical sense. More precisely, the active and sensory states that separate the internal from the external are referred to as the Markov blanket or the blanket states. Biological organisms, such as the brain, self-organise at various levels. This Markov blanket structure can thus be applied at different levels, for example at the level of cortical microcircuits, or even at the level of individual neurons [79]. Section 2.1.7.2 provides more details on Markov blankets in the brain.

For passive perception without additional assumptions on the structure of the underlying generative model, the central claims of the FEP are equivalent to those of the Bayesian brain hypothesis [62]. A crucial conceptual difference is that the FEP treats inference as a process of optimisation, where an additionally introduced distribution q *approximates* the posterior distribution $p(v|o)$ of the hidden states [14, 50, 51]. Computing an approximation is desirable here, since in real-world problems, such as perception of natural scenes, the exact posterior is generally very expensive or intractable to compute.

The resulting optimisation problem can be noted as:

$$\begin{aligned} q^*(v) &= \underset{q(v)}{\operatorname{argmin}} \quad D_{KL}(q(v) \| p(v \mid o)) \\ &= \underset{q(v)}{\operatorname{argmin}} \sum_v q(v) \log \frac{q(v)}{p(v \mid o)} \end{aligned} \tag{4}$$

where the optimal approximate posterior $q^*(v)$ minimizes the distance between approximate posterior $q(v)$ and the true posterior $p(v|o)$ of the hidden states. Intuitively, if the true posterior is contained in the variational family of the approximating distributions,

$p(\nu \mid o) \in Q$, then the approximation is exact, and the predictions of the FEP reduce to those of exact Bayesian inference.

The FEP proceeds by expressing this optimisation problem in a form that does not depend directly on the exact posterior $p(\nu \mid o)$, which is typically not known. The core idea is to reformulate the optimisation problem by unpacking the KL divergence between variational and true posterior:

$$\begin{aligned} D_{KL}(q(\nu)\|p(\nu \mid o)) &= \sum_{\nu} q(\nu) \log \frac{q(\nu)}{p(z \mid o)} \\ &= \sum_{\nu} q(\nu)(\log q(\nu) - \log p(z \mid o)) \\ &= \sum_{\nu} q(\nu) \left(\log q(\nu) - \log \frac{p(o,\nu)}{p(o)} \right) \\ &= \log p(o) + \sum_{\nu} q(\nu)(\log q(\nu) - \log p(o,\nu)) \\ &= \log p(o) + \sum_{\nu} q(\nu) \log \frac{q(\nu)}{p(o,\nu)} \\ &= \log p(o) + F(o,\nu) \end{aligned} \tag{5}$$

and noting the following relationship between the marginal likelihood $p(o)$ and the KL divergence:

$$\log p(o) = D_{KL}(q(\nu)\|p(\nu \mid o)) - F(q(\nu)) \tag{6}$$

where $F(o,\nu)$ is the variational free energy (VFE), also known as the negative of the generative model's evidence lower bound, the ELBO [12, 104]

$$F(o,\nu) = \sum_{\nu} q(\nu) \log \frac{q(\nu)}{p(o,\nu)} \tag{7}$$

which depends only on the approximate hidden state distribution $q(\nu)$ and the joint distribution of observations and approximate hidden states $p(o,\nu)$, both of which are accessible. Here, the marginal likelihood $p(o)$, or model evidence, is the probability of generating the observation o with the prior of the learned internal generative model, after integration over parameters.

Crucially, since the KL divergence in equation 6 is always positive, the surprise $\log p(o)$ from observations o is always bound by the VFE:

$$\underbrace{F(o,\nu)}_{\text{free energy}} = \underbrace{D_{KL}(q(\nu)\|p(\nu|o))}_{\text{KL divergence}} \underbrace{-\log(p(o))}_{\text{surprise}} >= \underbrace{-\log(p(o))}_{\text{surprise}} \tag{8}$$

This means that, under the FEP, we can characterize surprise minimization with the minimization of free energy with respect to the currently observed input.

Another useful decomposition of the VFE separates it into and accuracy and a complexity term:

$$\underbrace{F(o, v)}_{\text{free energy}} = \underbrace{D_{KL}(q(v) \| p(v))}_{\text{complexity}} - \underbrace{E_q(\log p(o \mid v))}_{\text{accuracy}} \tag{9}$$

This drives the optimization process towards accurate predictions of encountered observations o while keeping the complexity, or divergence between approximate posterior and its prior distribution as small as possible.

2.1.4 *Gaussian states and mean-field approximations*

In the previous sections, we have seen how the FEP turns Bayesian inference into a process of optimisation by choosing a parametric family of probability distributions Q for the approximate distribution. If the variational family is unrestricted, i.e. contains the exact posterior, then inference is exact. The following sections will talk about PC models, that restrict the variational family using a mean-field approximation and often resort to the Laplace approximation to infer states and parameters of the model. We will also cover generalized coordinates of state motion, which allow to express not only the states themselves but also their (expected) motion. In contrast to unrestricted Bayesian inference, PC models make specific and testable predictions about the underlying algorithmic and implementational structure.

The mean field approximation for hidden states simplifies the structure of the approximate posterior by assuming that it factorizes over components of the hidden state space:

$$q(v_1, \cdots, v_m) = \prod_{i=1}^{m} q(v_i) \tag{10}$$

This implies that no interactions between the Gaussian random variables are allowed, i.e. they are modelled as independent parts [163]. This sort of simplification will typically not contain the true posterior when latent variables are mutually dependent.

An approximation that is often made for PC models is to model continuous distributions $q(v)$ with a Gaussian form, such that the variational distribution is parameterized by a mean μ and a covariance Σ parameter [51, 54]:

$$q(v) = \mathcal{N}(v; \mu, \Sigma) \tag{11}$$

In many cases the structure of the generative model is organized hierarchically, i.e. contains several layers of hidden states, where each hidden state is generated as a function of the next higher hidden state. In this hierarchical case, a factorization of the variational posterior can be made not only across the components of a hidden state, but also over the hierarchical levels of the entire model [49, 51]. For N layers, the corresponding hierarchical generative model has the form

$$q(\nu) = \prod_{l=1}^{N} q\left(\nu_l \mid \nu_{i+l}\right) = \prod_{l}^{N} q\left(\nu_l; \mu_l, \Sigma_l \mid \nu_{l+1}; \mu_{l+1}, \Sigma_{l+1}\right) \quad (12)$$

and models each hierarchical layer as only dependent on the respective higher layer. Again, we assumed the hidden states to be modelled as multivariate Gaussians, parameterized by mean and covariance. This hierarchical structure is the basis for hierarchical PC models, which infer optimal hierarchical states from sensory observations using locally interacting layers.

Even more generally, another mean-field approximation that is typically assumed more implicitly in many process models is to model the effects of model parameters in terms of timescales as independent. A particularly prominent approach is to divide model parameters λ into three sets $\lambda = \lambda_u, \lambda_\gamma, \lambda_\theta$ that model rapidly, slowly and very slowly changing parameters [52]:

$$q(\nu) = \prod_{l=1}^{N} q\left(\nu_l; \lambda_l\right) = q\left(\nu_u; \lambda_u\right) q\left(\nu_\gamma; \lambda_\gamma\right) q\left(\nu_\theta; \lambda_\theta\right) \quad (13)$$

Rapid changing parameters λ_u are typically compared to neuronal activity in the brain, while the more slowly changing parameters λ_γ resemble molecular signalling and neural modulation [52]. Finally, the very slowly changing parameters λ_θ are often related to experience-based and long-term changes in neural connectivity [52]. This sort of mean-field approximation across parameter time-scales is found in many neural process models under the FEP as well as in machine learning models, such as in the separation between state inference, attentional processing and weights learning with deep neural networks.

2.1.5 *The Laplace approximation*

Next to the mean-field approximations over parameter structure and their temporal scales, another important approximation for many PC models is the Laplace approximation (LA). The LA is present in many PC models in neuroscience that have explicitly been mapped to canonical (cortical) microcircuits in the human brain [11, 51]. The core implication of the Laplace approximation is that, assuming continuous

Gaussian distributions, the covariance Σ of approximate hidden states ν is not directly inferred, but instead is a function of the mean.

Technically, this is based on a second-order Taylor series expansion around the mode of the variational posterior, i.e. around a local maximum or peak of the probability density function. The conditional covariance can then be approximated using the curvature of the free energy, more specifically the negative Hessian of the log joint probability at the mode [54, 162, 170].

We first explicitly specify the previously introduced generative model with respect to model parameters θ, that mediate the influence of hidden variables ν on observed variables o via their joint density $p(o, \nu \mid \theta)$. In the context of artificial neural networks, θ often refers to the learnable weights of the model. The free energy can be expressed as

$$\begin{aligned} F(o, \nu, \theta) &= \sum_{\nu} q(\nu) \log \frac{q(\nu)}{p(o, \nu \mid \theta)} \\ &= E_q(\log q(\nu)) - E_q(\log p(o, \nu \mid \theta)) \end{aligned} \tag{14}$$

where the first term $E_q(\log q(\nu))$ is the entropy that generally is tractable. Again, we assume that hidden states ν are Gaussian with respect to mean and covariance parameters μ and Σ. The Laplace approximation proceeds by approximating the second term $E_q(\log p(o, \nu \mid \theta))$ using a second-order Taylor series expansion around the mode of the variational posterior:

$$\log p(o, \nu \mid \theta) \approx \log p(o, \mu^* \mid \theta) - \frac{1}{2}(\nu - \mu^*)^T H (\nu - \mu^*) \tag{15}$$

where first the variational posterior $\mu^* = \operatorname{argmax}_{\mu} \log p(o, \nu \mid \theta)|_{\nu=\mu}$ needs to be determined. The Hessian at the mode $H = -\nabla_\nu \nabla_\nu \log p(o, \nu \mid \theta)|_{\nu=\mu^*}$ can then be determined by differentiating the log joint probability twice. We can now express the approximating distribution $q(\nu \mid o) = \mathcal{N}(\mu, \Sigma) = \mathcal{N}(\mu, H^{-1})$ as a function that depends only on the mean parameter μ [162, 170].

2.1.6 *Predictive coding under the Free Energy Principle*

Previously, we introduced the idea that the brain optimizes an internal generative model whose organization is assumed to explain how the observations that humans encounter in their environment are generated. we have seen that specific assumptions about the structure of this internal generative model, such as Gaussian distributions and a factorization over hierarchical layers can be made. With these assumptions in place, we now can describe a specific process model that optimises the parameters of this generative model. The process of

empirically inferring model parameters from observations requires a "inversion" of the hierarchical internal generative model and the integration of observations that arrive over time. For the approximations discussed before, PC is such a process model. In PC, the optimisation of fast changing parameters, i.e. inference and slower changing parameters, i.e. attention and learning is achieved via a simple algorithmic scheme that is repeated for each hierarchical layer: Feedback information from a respective higher hierarchical layer conveys predictions, while feedforward information is propagated back towards the higher layer and carries prediction errors, i.e. the divergence between prediction and data [51, 172]. A core implication of such locally informed learning is that no global error signal is necessary. This is in contrast to the backpropagation of error algorithm for deep neural networks, that propagates errors from a global objective function back to all parameters of the model [177]. In particular, such locally informed learning can be implemented in the context of biologically plausible, Hebbian update rules, where the change applied to a weight only depends on the activity of presynaptic and postsynaptic neurons. In the context of artificial neural networks, this means that the weight update depends strictly on information that is locally available to the optimized weight parameter, the activity at its input and output [225]. It should be noted that PC models are not the only possible process models that can be derived from the FEP [58]. Still, they are a prominent implementation of the FEP as they allow a straightforward mapping onto biological structures, which is more difficult for other variants, such as (marginal) belief propagation [11, 51].

The following sections will explain how hierarchical and dynamical inference is achieved in PC. In order to introduce all relevant aspects, we will focus on a general model described by [51]. Many existing PC variants can be interpreted as a special case of such generalized PC. For example, when predictions over time are excluded, we recover hierarchical PC. In chapter 7 we compare hierarchical PC to VAEs in deep learning, which have a similar computational goal, but use global backpropagation of error [128]. Another special case of generalized predictive coding (GPC) is dynamical PC (i.e. without hierarchical organisation), which generalizes from many popular models in engineering, such as the Kalman filter or Linear Predictive Coding (LPC) in signal processing [95, 158, 163]. Indeed, the PC theory has a its historical roots in efficient signal processing and data compression [40, 158, 172]. In this thesis we will relate dynamical PC to recurrent neural networks (RNNs) trained with exact error backpropagation through time (BPTT). In particular, in chapter 5 we design and evaluate a RNN based deep PC model that predicts sequences of probabilistic latent states using BPTT. The model presented in chapter 7 is based strictly on the temporal filtering described by GPC, i.e. does not require prop-

agating exact errors sequentially through time. It is important to note that these differences focus on the algorithmic level, i.e. the models differ in *how* free energy is optimised.

2.1.6.1 *Dynamical models and generalized coordinates*

Predictive coding as a process model under the FEP rests on several publications by Karl Friston and colleagues, which slightly differ in the assumptions for the underlying optimisation process [49, 51, 52, 56]. Here, we focus on the general model discussed in [51], which rests on the mean-field approximation and the Laplace approximation, which we have covered before.

In dynamical PC, inference with respect to sequentially arriving sensory data o is achieved by defining a generative model that contains model parameters θ and state variables. Crucially, the state variables are now split up into hidden state variables x and cause state variables v, which model temporal and hierarchical dependencies respectively:

$$y = g(x, v, \theta) + z \tag{16}$$

In analogy to the static generative models discussed previously, the function $y = g(x, v, \theta)$ relates hypotheses from latent distributions x, v to model responses y. We have intentionally used the variable y instead of o, to stress that it resembles observations from the generative model that we want to infer. The (possibly nonlinear) function g is parameterized by θ. Hypotheses about the temporal changes $\dot{x}$ of hidden states x are expressed via another (nonlinear) function f: The variable z explicitly represents observation noise, i.e. noise in the mapping from states to sensory predictions.

$$\dot{x} = f(x, v, \theta) + w \tag{17}$$

where w explicitly represents any noise that underlies the motion of hidden states over time. The presence of the function f turns this generative model into a *dynamical* generative model, since it models changes over time with respect to hypotheses generated from x and v.

During inference, this dynamical model can be used for sequential Bayesian filtering, i.e. the estimation of variables that evolve over time using only the current and past observations. For many practical applications, like Kalman filtering, dynamical predictions of form $\dot{x} = f(x, v, \theta)$ are already sufficient to process sequential observations. In the general case, however, observed sequential information can be highly complex, especially in the context of fast changing neural signals in the brain, and the noise underlying temporal changes might

have strong (temporal) correlation. For these reasons, generalized PC models suggest that the brain explicitly represents temporal changes not only with respect to the motion $\dot{o} = \frac{do}{dt}$ of observations, but also with respect to higher order derivatives over time, i.e. with respect to acceleration $\ddot{o} = \frac{do^2}{dt^2}$, jerk $\dddot{o} = \frac{do^3}{dt^3}$ and so on [49, 51].

This idea is reflected in generalized PC by explicitly modelling a hierarchy of dynamics. It uses a local linearity assumption, i.e. assumes that observed changes in states are linear or smooth when looking at very small time increments. The assumption of local linearity allows to ignore higher order derivatives of the functions f and g [49]. This assumption allows for "generalized" model responses $\tilde{y} = \tilde{g}(\tilde{v}, \tilde{x}, \theta)$ that are expressed via "generalized coordinates" of state motion

$$\begin{aligned} y &= g(x, v) \\ y' &= g_x x' + g_v v' \\ y'' &= g_x x'' + g_v v'' \\ &... \end{aligned} \tag{18}$$

where we now explicitly model states and *instantaneous* temporal derivatives of their motion, e.g. $\tilde{x} = [x, x', x'', ...]$ for hidden states and analogously for observations $\tilde{y}$ and cause states $\tilde{v}$. It is important to note that specifically the local trajectory of states, i.e. the motion of the state at a particular point in time is modelled. In effect, the observer equation $\tilde{y} = \tilde{g}(\tilde{v}, \tilde{x}, \theta)$ still refers to only a specific point in time [51].

Similarly, the motion of hidden states is modelled as

$$\begin{aligned} \dot{x} = x' &= f(x, v) \\ \dot{x'} = x'' &= f_x x' + f_v v' \\ \dot{x''} = x''' &= f_x x'' + f_v v'' \\ &... \end{aligned} \tag{19}$$

where the generalized motion of the hidden state $\tilde{x} = \tilde{f}(\tilde{v}, \tilde{x}, \theta)$ is generated using function $\tilde{f} = [f, f', f'', \ldots]$. This requires the function f itself and the derivative with respect to its inputs. Equation 18 is usually referred to as "observer equation" and equation 19 is referred to as "state equation". Crucially, the cause states v act as control inputs that perturb the dynamics of hidden states x, allowing to separate between the modelled dynamics themselves and possible causes for these dynamics.

We can interpret this structure as a generalized state space model, that generalizes from standard state space models that only consider first order, like the Kalman filter [49, 95]. Similarly, RNNs and variants like the long short-term memory (LSTM) typically model only first order changes of their inputs explicitly [42, 80, 177]. Another difference to standard space models is the assumption that the noise on

each order of generalised motion is correlated. Under generalized PC, the generalized observation noise $\tilde{z} = [z, z', z'', \ldots]$ and the noise in the transition function $\tilde{w} = [w, w', w'', \ldots]$ are assumed to be analytic and have a well-defined covariance between orders of motion [49].

By assuming that observation noise z and transition noise w as well as all state priors are Gaussian, we can now define the probabilistic dynamical latent variable model that underlies PC when no hierarchy is considered. The model defines a joint distribution of possible observations (also called model response) y, cause states v and hidden states x:

$$p(\tilde{y}, \tilde{x}, \tilde{v}) = p(\tilde{y} \mid \tilde{x}, \tilde{v}) p(\tilde{x}, \tilde{v}) = p(\tilde{y} \mid \tilde{x}, \tilde{v}) p(\tilde{x} \mid \tilde{v}) p(\tilde{v}) \tag{20}$$

Figure 2 visualizes such a dynamical generative model.

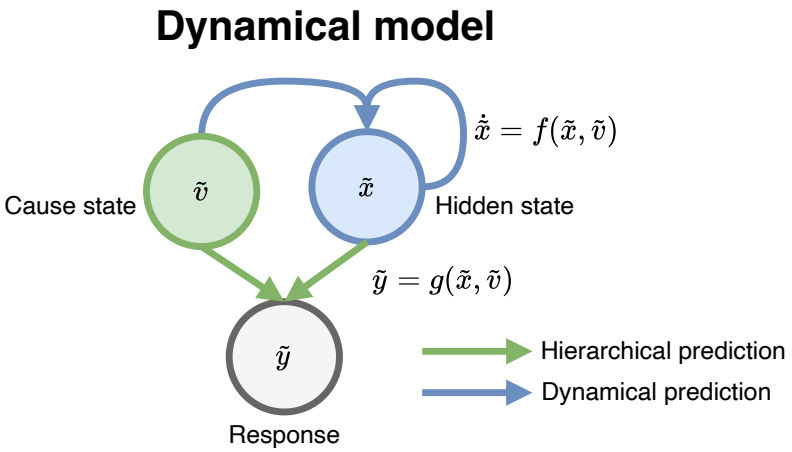


Figure 2: Dynamic generative model. Model responses are generated from cause and hidden states that represent estimated causes and temporal changes in causes respectively.

2.1.6.2 *Hierarchical-dynamical models*

The dynamical model of the previous section can generate hypotheses about the cause for possible observations and their motion over time. However, it does not yet account for a hierarchical abstraction of these possible causes. Hierarchical PC rests on the assumption that the brain maintains a generative model that factorizes into m conditionally independent layers

$$\begin{aligned} y &= g^{(1)}\left(x^{(1)}, v^{(1)} \mid \theta\right) \\ v^{(i-1)} &= g^{(i)}\left(x^{(i)}, v^{(i)} \mid \theta\right) \\ v^{(m)} &= v^{(m),p} \end{aligned} \tag{21}$$

where, for each pair of adjacent layers, a separate function g links the states of a higher layer to the cause states of a lower layer [51]. Like in the single-layer model in the previous section, each hierarchical layer i also maintains a function that models the motion of its hidden states $\dot{x}^{(i)} = f\left(x^{(i)}, v^{(i)}\right)$. For the sake of clarity, we have omitted the specification of generalized coordinates for states and functions in each hierarchical layer.

Since hierarchical layers are assumed to be conditionally independent, the overall hierarchical-dynamical model can be expressed as:

$$p(\tilde{x}, \tilde{v}) = p\left(\tilde{v}^{(m)}\right) \prod_{i=1}^{m-1} p\left(\tilde{x}^{(i)} \mid \tilde{v}^{(i)}\right) p\left(\tilde{v}^{(i)} \mid \tilde{x}^{(i+1)}, \tilde{v}^{(i+1)}\right) \quad (22)$$

where the prior on hidden states $p(\tilde{x}^{(i)}, \tilde{v}^{(i)})$ in 21 is now expressed as a empirical prior that depends on the next higher hierarchical layer. Since we deal with states in generalized coordinates of motion, these empirical priors refer to priors on the instantaneous dynamics of the cause states of the respective lower hierarchical layer.

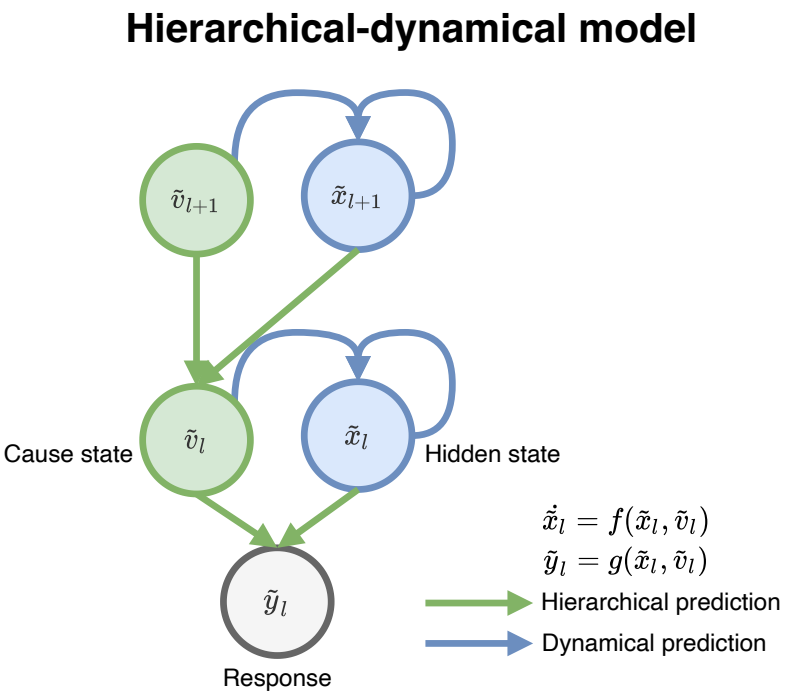


Figure 3: Hierarchical-dynamical generative model. Model responses are generated from cause and hidden states that represent estimated causes and temporal changes in causes respectively. In contrast to the dynamical model in Figure 2, the causes in each hierarchical layer are generated by the hierarchical prediction from the respective higher layer.

Figure 3 shows such a hierarchical-dynamical generative model with two hierarchical layers. Since in each hierarchical layer the cause

states model perturbations to the motion of hidden states, this overall structure allows to formulate quite complex abstractions over hierarchical and temporal dynamics between hierarchical layers.

2.1.6.3 *Variational inference, learning and amortization*

The previous sections motivated an approximate variational approach to Bayesian inference by approximating a complex target distribution with a simpler family of distributions and minimizing the distance between the approximation and target distribution. As we have seen, the approximating distribution can be organized hierarchically and even cover temporal dependencies. This leads to quite complicated generative models, such as the hierarchical-dynamical model in Figure 3. Such variational inference is a general method that has found widespread use in complex applications, such as statistics, machine learning or computational biology [12, 232].

As we will see in the next section, PC models under the FEP take an iterative approach to estimating the approximating distribution as well as the parameters of the generative model. Early formulations of PC were derived as a variant of the expectation-maximisation (EM) algorithm, which finds maximum a posteriori (MAP) estimates of the model parameters [31]. MAP estimation aims at (iteratively) maximising the posterior probability of a parameter, given observed data and a prior distribution over the parameter. These PC models iteratively switch between an expectation (E) step, which computes the optimal values of the states given the currently learned model parameters, and a maximisation (M) step, which optimises the parameters given the inferred states. Different variations to this scheme exist, often including the additional estimation of precision parameters [14, 51, 139]. Generally speaking, the E step can be associated with (state) inference, while the M step can be seen as (weights) learning. As detailed in the next section, both inference and learning in PCNs is done using a simple gradient descent on the variational free energy of the model, expressed as precision weighted prediction errors. This implies that in PCNs, next to the gradual optimisation of weights parameters, the inference at every datapoint is also an iterative procedure.

In the context of DNNs, the VAE is simple, yet effective and widely used architecture, that takes a slightly different approach to variational inference. Here, the focus lies on learning the weight parameters of the model, while inference of the parameters of state v given data o is amortized and simplifies to a single pass through the network [104]. Usually, the latent state is modelled as normal distribution with diagonal covariance:

$$v \sim q_\phi(v \mid o) = \mathcal{N}\left(v; \mu, \sigma^2 I\right) \tag{23}$$

In this context, amortization refers to the idea of using an additional set of parameters ϕ, that are shared across data points and express the inferred state as a function of the data. In VAEs, the parameters of ϕ are represented by a deep encoder network. Next to this encoder network, VAEs use a generative model θ to map from latent states to the expected observation, rendering the model an autoencoder. This amortization of the inference process is highly effective, since the encoder can be re-used for inference on new datapoints, instead of iteratively inferring the state distribution v for each new datapoint. Like PCNs, VAEs optimise an ELBO objective, as introduced in 2.1.3, usually in the following form:

$$\text{ELBO} = \mathbb{E}_{q(v|o)}[\log p(o \mid v)] - \text{KL}(q(v \mid o) \| p(v)) \tag{24}$$

where the first part measures the accuracy, or reconstruction error and the KL divergence in the second part measures the divergence of the encoded state distribution $q(v \mid o)$ from the prior distribution $p(v)$. VAEs optimise this by back propagating the error from the objective throughout the entire network [104]. In VAEs, propagation of errors through samples from the random latent variable $\tilde{v} \sim q_\phi(v \mid o)$ is done using the "reparameterization" trick. The trick refers to expressing the sampled distribution as a *differentiable* function $g_\phi(\epsilon, o)$ with respect to an additional noise variable ϵ:

$$\tilde{v} = \mu + \sigma \odot \epsilon \quad \text{with} \quad \epsilon \sim \mathcal{N}(0, I) \tag{25}$$

This sampling step is necessary, since VAEs explicitly encode the state covariance parameter. As mentioned in section 2.1.5, this is unlike PCNs which deal with it implicitly, e.g. using the Laplace approximation. In summary, both inference and learning in PCNs results in a gradient descent on the variational Free Energy, while VAEs reduce state inference to a single step using amortization. This difference between PCNs and VAEs has seen relatively little attention in the literature so far. However, there have been some attempts to include iterative inference in VAEs [128, 162]. Similarly, a version of amortized inference has recently been suggested for PCNs [208]. Chapter 7 of this work provides a similar approach to this, by showing that higher hierarchical layers in PCNs can be interpreted as performing amortized inference, when they have access to the observation. The next section will cover details about how generative models are inverted in PCNs using a gradient descent on prediction errors.

2.1.6.4 *Predictive coding*

In the previous sections, we covered the hierarchical and dynamical generative models that underlie PC under the FEP. Here we want

to specify the neuronal dynamics that are used to invert generative models, in order to empirically infer model parameters given sensory observations. Figure 4 visualizes the inversion of a hierarchical-dynamical generative model with two hierarchical layers.

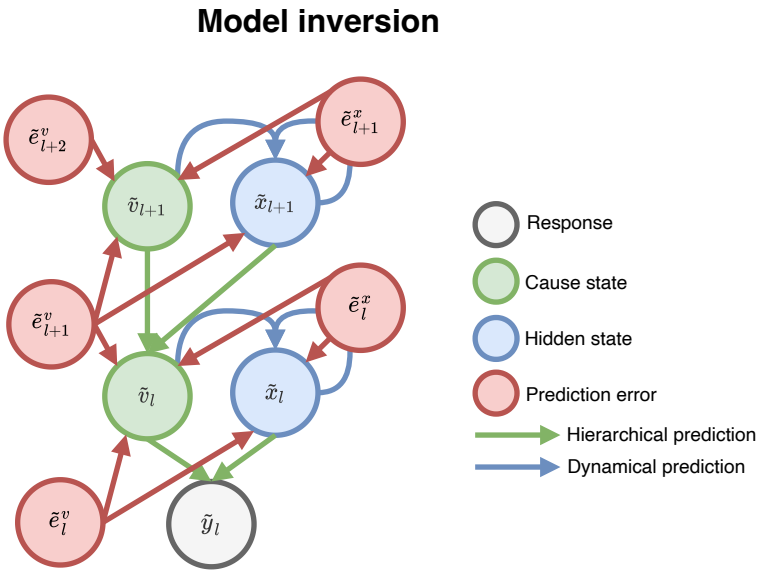


Figure 4: Predictive coding describes an inversion of the hierarchical-dynamical generative model shown in Figure 3 and allows to infer model parameters from observed data using locally computed prediction errors ϵ. Each layer is updated with respect to dynamical prediction errors ϵ^x on the predicted motion of their hidden states and hierarchical prediction errors ϵ^v from the outgoing and incoming hierarchical prediction.

Given the hierarchical-dynamical model described in section 2.1.6.2, GPC assumes that the brain generates locally informed predictions $\tilde{y} = \tilde{g}(\tilde{x}, \tilde{v})$ and $\dot{\tilde{x}} = f(\tilde{x}, \tilde{v})$ about lower states $\dot{\tilde{v}}$ and the motion of hidden states $\tilde{x}$ [49, 51]. At the lowest layer predictions are made about possible sensory observations $\tilde{y}$.

The distance between true causes (or sensory states) and their top-down prediction results in a prediction error ϵ^v for each hierarchical layer:

$$\tilde{\epsilon}^v = \begin{bmatrix} \tilde{y} \\ \tilde{v}^{(1)} \\ \vdots \\ \tilde{v}^{(m)} \end{bmatrix} - \begin{bmatrix} \tilde{g}^{(1)} \\ \tilde{g}^{(2)} \\ \vdots \\ \tilde{v}^{(m),p} \end{bmatrix} \tag{26}$$

In many implementations, the lowest state $\tilde{y}$ is initialised simply with an observed sample $\tilde{o}$ of the environment, while the intermediate states v are often initialised using their top-down predicted prior. However, other options, such as an initialization with zeros or the

mean of multiple samples are possible. For the highest layer m, there is no corresponding top-down prediction and the prediction error measures the divergence from a prior $\nu^{(m),p}$.

Similarly, dynamical prediction errors ϵ^x measure the distance between true hidden state motion Dx and the motion predicted within each hierarchical layer:

$$\tilde{\epsilon}^x = \begin{bmatrix} D\tilde{x}^{(1)} \\ D\tilde{x}^{(2)} \\ \vdots \\ D\tilde{x}^{(m)} \end{bmatrix} - \begin{bmatrix} \tilde{f}^{(1)} \\ \tilde{f}^{(2)} \\ \vdots \\ \tilde{f}^{(m)} \end{bmatrix} \tag{27}$$

where D is a temporal derivative operator, such that for hidden states $x = [x, x', x'', x''', ...]$ their temporal derivative can compactly be expressed by $Dx = [x', x'', x''', ...]$.

Given the hierarchical-dynamical latent variable model in 22, we can, like in the previous sections, resort to variational Bayes and focus on optimizing the variational free energy. The VFE of the hierarchical-dynamical model is:

$$F(\tilde{y}, \tilde{v}, \tilde{x}) = E_q(\log q(\tilde{v}, \tilde{x})) - E_q(\log p(\tilde{y}, \tilde{v}, \tilde{x})) \tag{28}$$

Generalized PC makes a mean field approximation over latent states and between hierarchical layers and uses the Laplace approximation to simplify neural dynamics as a function of the mean [49, 51].

Ignoring constants, one can express the energy compactly with respect to prediction errors:

$$\log p(\tilde{y}, \tilde{x}, \tilde{v} \mid \theta) = \frac{1}{2} \ln |\tilde{\Pi}| - \frac{1}{2} \tilde{\epsilon} \tilde{\Pi} \tilde{\epsilon} \tag{29}$$

where the generalized precision $\tilde{\Pi}$ refers to the overall unexplained variance of the prediction error, or the uncertainty that arises in the model during inference [51]. It can be separated into the precision of the observation and the transition function, representing hierarchical and dynamical predictions respectively:

$$\tilde{\Pi} = \begin{bmatrix} \tilde{\Pi}^z & \\ & \tilde{\Pi}^w \end{bmatrix} \tag{30}$$

This precision $\tilde{\Pi}$ is usually estimated empirically from the observed prediction errors, typically by estimating the variance and covariance components of the error. The prediction errors themselves simply measure the difference between predicted and observed mean parameters, as we will see next.

The prediction errors $\tilde{\varepsilon}$ summarize hierarchical and dynamical prediction errors:

$$\tilde{\varepsilon} = \begin{bmatrix} \tilde{\varepsilon}^{v} \\ \tilde{\varepsilon}^{x} \end{bmatrix} = \begin{bmatrix} \tilde{y} - \tilde{g}(\tilde{x}, \tilde{v}, \theta) \\ D\tilde{x} - \tilde{f}(\tilde{x}, \tilde{v}, \theta) \end{bmatrix} \tag{31}$$

The updates for mean μ of states x, v can then be described by computing the influence of the states on the energy at each timestep:

$$\dot{\tilde{\mu}} - D\tilde{\mu} = (\log p(\tilde{y}, \tilde{v}, \tilde{x} \mid \theta))_{\tilde{\mu}} \tag{32}$$

Here, $\dot{\tilde{\mu}} - D\tilde{\mu}$ denotes the difference between the change $\dot{\tilde{\mu}}$ of the parameter and the encoded (i.e. already expected) change $D\tilde{\mu}$.

Under the Laplace approximation, the updates for cause and hidden states can be expressed with respect to the corresponding prediction errors based only on the inferred and predicted mean parameters:

$$\begin{aligned} \dot{\tilde{\mu}}_v^{(i)} &= D\tilde{\mu}_v^{(i)} - \tilde{\varepsilon}_v^{(i)} \tilde{\Pi}^{z,(i)} \tilde{\varepsilon}_v^{(i)} - \tilde{\Pi}^{z,(i+1)} \tilde{\varepsilon}_v^{(i+1)} \\ \dot{\mu}_x^{(i)} &= D\tilde{\mu}_x^{(i)} - \tilde{\varepsilon}_x^{(i)} \tilde{\Pi}^{w,(i)} \tilde{\varepsilon}_x^{(i)} \end{aligned} \tag{33}$$

2.1.6.5 *Predictive coding and precision*

In the previous section, we have discussed that the weighting applied to the prediction error depends on the precision $\tilde{\Pi}$ of observation and transition noise. As mentioned before, this precision is usually estimated from the observed prediction error covariance. In many models that implement a variant of predictive coding, the precision is assumed to be constant in practise, which simplifies the ensuing gradient descent on prediction errors further. Predictive coding models used in machine learning are trained on static observations, or on discrete samples from sequential data at a low frequency. This often makes it difficult or even impossible to estimate the prediction error covariance from a single observation, which would require sampling at high frequencies [14]. In these cases, the prediction error covariance could still be estimated based on the differences between multiple observations, e.g. across a batch of examples or an entire dataset. Especially early formulations of PC models stress the role of the prediction error precision as part of the empirical prior that higher hierarchical layers provide for lower layers during inference [48, 51]. Furthermore, the prediction error precision has been associated with complex attentional processing, primarily due to its role in mediating the gain on prediction errors during inference [8, 46]. In some cases the underlying models include additional structure in the generative model, e.g. allowing to learn state-dependent precision [46]. In summary, while

the prediction error precision during inference is often neglected in practise, it has important cognitive interpretations.

As explained in section 2.1.5, under the Laplace approximation, the state covariance parameter does not need to be inferred explicitly, but is a function of the mean, given by the curvature around the inferred mode, i.e. the optimal posterior of the state mean *after* inference. However, similar to the precision of the prediction error, this curvature is often not computed in practise and is used primarily to motivate a gradient descent on the mean of the states as the only relevant quantity during inference. Including the Laplace approximated covariance of the states, however, provides relevant information on the uncertainty of the states during *learning*. In chapter 7 we implement and analyse a predictive coding model that includes such curvature in the context of rectified linear nonlinearity, where it is easy to compute.

2.1.7 *Canonical computations*

2.1.7.1 *Predictive coding and canonical microcircuits*

Hierarchical and dynamical PC models, as discussed in the previous sections provide a relatively simple algorithmic scheme that describes how a Bayesian generative model can be inverted in order to infer parameters from sensory observations. An appeal of PC as a process theory under the FEP is that it can be mapped to the physical structure of cortical hierarchies in the brain and allows to relate evoked brain responses to inferential processes [11, 49]. The cortical canonical microcircuit describe a neural circuitry, i.e. connectivity scheme between multiple types of neural populations in cortical columns, that is replicated throughout the entire cortex [11].

Research on canonical circuitry in the cortex has a long history and initially focused on the role of repeated functional elements throughout the neocortex in visual processing [36, 37]. A central insight that underlies these early as well as more recent formulations of canonical circuits is that cortical columns are organized hierarchically, each spanning multiple horizontal layers of the cortex. While each of these layers contains different cell population types, their function appears to be repeated between columns, indicating that the functional properties in each column are shared. As part of the cerebral cortex, the neocortex has only relatively recently evolved and covers several higher level cognitive functions in mammals, such as spatial reasoning, language or motor control [122, 171]. The neocortex is often divided into six layers with different cell types, in ascending order from the most superficial (or outermost) layer towards the deeper (or more inwards) layers. Figure 5 shows an overview of these six layers, including their intrinsic and extrinsic connectivity. While the exact form of these six layers within cortical columns differs between dif-

ferent areas in the cortex, the overall functionality of the individual layers can be summarized into a canonical form [11].

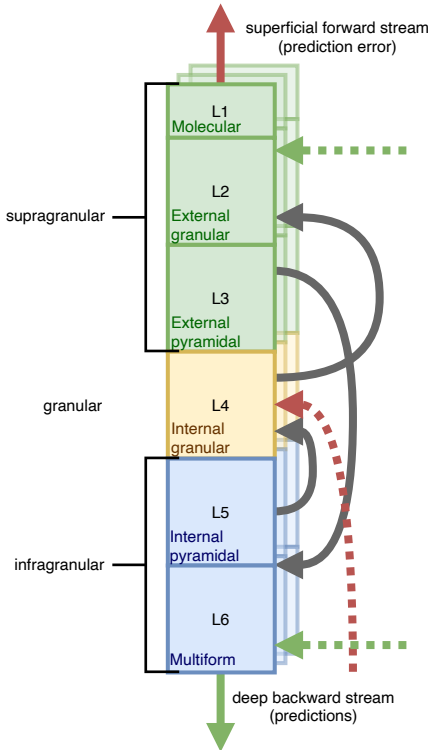


Figure 5: Simplified overview over intrinsic (gray) and extrinsic (green, red) connectivity in the cortical canonical microcircuit, as discussed by Bastos es et al. [11]. Shown is a vertical column that spans six horizontal layers and is repeated horizontally.

Within each column, information is processed (recurrently) via intrinsic connections. Superficial cells in the supragranular layer send feedforward information towards higher cortical areas, while the infragranular layer provides a feedback stream towards lower cortical areas. Maybe most strikingly, the cells in the granular layer (L4) receive the majority of feedforward information, while superficial and deep layers receive information from a backward stream. Research has shown that columns along the cortical layers are organized hierarchically, where lower layers provide feedforward signals for higher layers [11, 79]. Afferent, i.e. incoming, feedforward signal can be related to prediction errors in predictive coding and is reciprocated with a feedback signal towards the respective lower area. In the context of predictive coding models, this feedback can be related to the prediction signal.

The most prominent version of the cortical canonical microcircuit differentiates between the following cell types: Superficial pyramidal cells, deep pyramidal cells, spiny stellate cells and excitatory or inhibitory interneurons [11]. Pyramidal cells are found in the supragranular and infragranular layers, while interneurons are found in

the granular layer and supragranular layer [11]. Finally, spiny stellate cells are located in the granular layer. In the context of predictive coding, each of these cell types, including their extrinsic and intrinsic connectivity can be related to different parts of the generative model. With the PC model of the previous section in mind, the superficial pyramidal cells are interpreted as encoding and propagating precision weighted prediction errors on cause states v. Deep pyramidal cells are thought to represent conditional expectations on the cause v and hidden states x and to propagate top-down predictions towards lower layers in the cortical hierarchy. Spiny stellate cells encode precision weighted prediction errors on the cause states v of the lower hierarchical layer. The inhibitory interneurons of the granular layer are assigned to represent precision weighted prediction errors on the hidden states x, while interneurons in the supragranular layer represent expectations on hidden states x and cause states v [11]. In summary, the inferential structure of PC models can directly be mapped onto canonical microcircuits in the brain, supporting the notion of PC as driving a canonical pattern of computation in the brain. The next section will resort to this canonical cortical microcircuit in a further simplified form.

2.1.7.2 *Markov blankets throughout the brain*

Next to identifying canonical neural correlates of PC as a process model, it is possible to identify free energy minimising, self-organising structures across scales in the brain. One approach rests on the mapping of Markov blankets, i.e. functional boundaries marked by statistical independence, to structures in the brain across scales [79, 105]:

Markov blanket

The boundaries of a system can be defined in a statistical sense, by separating (complex, dynamical) systems into internal and external states that are separated by a Markov blanket, which consists of active and sensory states [105].

The Markov blanket thus refers specifically to the blanket states, the active and sensory states that make external states conditionally independent from internal states and vice versa. More precisely, internal (i) and external (e) are conditionally independent, when their joint probability conditioned on the Markov blanket (b) is equal to the product of their marginal probability conditioned on the blanket [79]:

$$i \perp e \mid b \Leftrightarrow p(i, e \mid b) = p(i \mid b)p(e \mid b) \tag{34}$$

The FEP implies that, when such a Markov blanket exists for a complex random dynamical system, then it will appear to minimize its free energy over time [105]. From a modelling perspective, this view on self-organisation stresses the importance of the chosen factorization in the generative model, such as discussed in 2.1.4. Markov blankets often are mentioned in the context of *active inference*, a corollary of the FEP which focuses on the role of action in free energy minimization. Since we here focus on perceptual inference, active inference using physical actions is outside of the scope in this work. However, active states can also be mapped to non-physical actions, and even to the process of prediction itself [79].

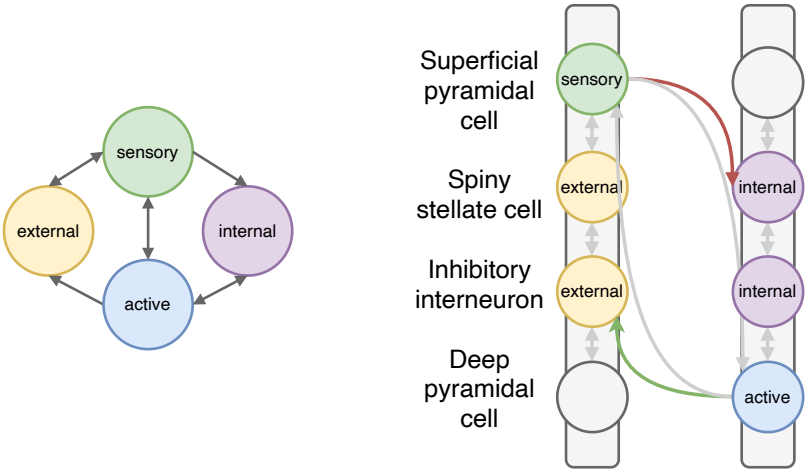


Figure 6: Left: Markov blanket structure including *internal* states that are conditionally independent from *external* states. Right: Regional Markov blanket mapped to the canonical cortical microcircuit, as discussed by [79]. In the canonical microcircuit, cells emitting feedforward signals (red arrow) and feedback signals (green arrow) can be interpreted as the sensory and active states of a Markov blanket that separates individual columns.

In this case, actions appear more implicitly: Top-down predictions, as active states, influence the activity of lower layers. This is particularly evident in the hierarchical layout of PCNs, where predicted states try to minimize their complexity based on their top-down prediction. Similarly, top-down predictions can be weighted by precision, thereby actively sampling only those parts of the input where the precision is high. In contrast, bottom-up prediction errors can be interpreted as sensory blanket states, which drive updates of internal states [105]. In PC, predictions and prediction errors thus can be thought to separate the internal (higher) from the external (lower) levels. A schematic mapping of a Markov blanket structure onto a simplified canonical microcircuit is shown in Figure 6.

Next to the level of cortical circuits, Markov blankets also cover individual neurons and coarser structures, such as entire brain regions or the overall function of the brain [79, 105]. Crucially, these

self-organising sub-systems can still be identified as minimizing free energy. In summary, the neural mechanisms underlying PC describe a specific level of self-organisation, a special case of the Markov blanket as a canonical structure that can be found across scales within the brain.

2.2 DEEP NEURAL NETWORKS

After covering the conceptual and mathematical background of the FEP and PC, the following sections cover technical fundamentals for deep neural network architectures, their application in artificial intelligence and the backpropagation of error algorithm. The content in this section is partially based on the book chapter:

> [pub:6] A. Ofner and S. Stober. "Deep Neural Networks and Auditory Imagery." In: *Music and Mental Imagery*. Ed. by M. B. Küssner, L. Taruffi, and G.A. Floridou. Routledge, 2022, pp. 112–122

2.2.1 *Supervised and unsupervised learning*

A central aspect of machine learning algorithms is their capacity to represent information. Many machine learning (ML) algorithms, such as logistic regression or naïve Bayes, work on representations that are defined before solving a particular task. Within the field of ML, the set of representations that are available to the algorithm to make inference are called features. For many applications it might be sufficient to provide specifically chosen features. Often times, however, it is difficult to find the right set of predefined features, especially when dealing with complex data. These considerations have led to the field of representation learning in ML. Representation learning algorithms no longer project points from a predetermined feature space to outcomes. Instead, they allow to learn the representation itself. Such representation learning methods can extract complex sets of features, which can be more complex and predictive than hand-crafted features. Many representation learning algorithms furthermore allow to find sets of features without human supervision, allowing to tackle large and diverse amounts of data. An important example for such representation learning algorithms are autoencoders. Autoencoders exist in deterministic or stochastic variants, such as the VAE and learn representations of data by converting inputs into an internal representation, which in turn is decoded back into the original data [104]. This way, autoencoders are trained to preserve as much information as possible, even if the internal representation is reduced in size. A good representation representation explains variations in the data well and is disentangled. For example, an algorithm working on images might

learn that an object's color is affected by the time of day. An important problem that arises here is the aspect of extracting high-level features from raw data. This challenge has been addressed with neural networks (NNs) and particularly the class of DNNs, which are based on the idea of expressing complex representations as weighted combinations of simpler representations. These computational models excel at statistical pattern recognition, especially when trained on large datasets, often times containing millions of (labelled) data points. A vast selection of different network architectures has been developed, often tailored to deliver high accuracy on a particular task [116, 194]. Common to deep learning models is that they are built from multiple layers that learn representations with increasing levels of abstraction as information propagates towards deeper, hidden neural layers.

2.2.2 *Neural networks and backpropagation of error*

Feed-forward neural networks relate their inputs x to outputs y via a function $y = \sigma(Wx + b)$ that depends on learnable weights W and biases w and are activated via differentiable, nonlinear functions σ, such as a sigmoid or rectified linear unit (ReLU) [148]. Multiple feed-forward layers can be stacked to form DNNs that can learn complex nonlinear transformations from inputs to outputs. Next to fully connected feedforward NNs, other variants, such as convolutional neural networks or recurrent neural networks exist, which will be discussed later.

DNNs are usually trained using a combination of gradient descent and the backpropagation of error algorithm [177]. Generally speaking, a backpropagation of error throughout neural networks requires that the entire model is differentiable. For inputs x, targets y and outputs $\hat{y}$, the internal parameters of the network are changed based on an error signal, where the magnitude of the error usually is parameterized by a a scalar objective function, the loss function $L(\hat{y}, y)$.

Using the chain rule, the gradients, i.e. first-order partial derivatives of the loss function L with respect to all trainable parameters of the model need to be computed, although optimising using higher order methods is theoretically possible. The parameters can then be adjusted using gradient descent. Generally speaking, there is no guarantee that training via backpropagation reaches a global minimum of the loss function.

While inference is made with respect to individual data points, weights training (i.e. learning) is ideally made with respect to the entirety of a dataset. In practise, however, this is often infeasible. Instead of requiring entire datasets to be processed in parallel, stochastic gradient descent allows to make weight updates with respect to batches of data [99, 175].

Gradient descent iteratively updates trainable parameters w using a learning rate η

$$w \leftarrow w - \eta \nabla_w L(y, \hat{y}) \tag{35}$$

where $\nabla_w L(y, \hat{y})$ is the gradient of the loss function with respect to the parameters.

A variety of efficient methods have been introduced that allow to train large neural networks, e.g. using approximate second order methods, such as Adam or RMSprop [103]. These typically scale the learning rate during gradient descent, e.g. based on the variance of the gradients.

2.2.3 *Convolutional neural networks*

Several aspects underlying deep neural networks bear resemblance with neural architectures found in the biological brain. For example, convolutional neural networks (CNNs) show connectivity similar to the organization of neurons in the visual cortex, where individual neurons respond only to activation within their respective receptive fields [121, 159, 211]. The CNN architecture is in turn a refinement of the "neocognitron" published in [60]. Initially designed for relative simple and small inputs, like handwritten digit recognition, CNNs have been employed in increasingly complex architectures, such as LeNet or AlexNet, that featured more convolutional layers and larger inputs [111, 117, 118]. In CNNs, spatial receptive fields are efficiently computed across input dimensions and increasingly complex representations are learned through combining features in overlapping receptive fields.

The core idea behind CNNs is a to use a combination of *convolution* and *pooling*. Convolutional kernels, also referred to as filters, are applied by sliding over spatial dimensions of inputs and computing the dot product between filter weights and the input. This allows to share the same weights across different regions of the input. The output activation of the convolution operation are feature maps that can be trained to activate for specific spatial features in the input. With learnable weights w and biases b, a CNN layer applies a function of form $y = \sigma(w * x + b)$ for input feature maps x and output feature maps y, where σ is a nonlinear activation function and $*$ refers to the convolution operation. Typically, each filter is applied to the entire depth of the input, restricting local connectivity to the spatial domain. These feature maps can be down sampled using pooling operations. Pooling increases the size of the receptive field by summarizing neighboring values, e.g. by computing the maximum ("max pooling"), or the average ("average pooling") value. This allows CNNs to compute abstract representations with receptive fields that can cover the entire input. CNNs are particularly suited for spatial processing, as they excel at

aggregating increasingly abstract features and their spatial relations. They have significantly improved state-of-the-art results in a wide range of fields, such as visual object detection or image segmentation [111, 116].

2.2.4 *Recurrent neural networks*

Next to CNNs, RNNs are an important class of deep neural networks [42, 191]. In contrast to feedforward neural networks, RNNs feature recurrent or feedback connections between internal states in the network, making them a good choice for processing sequential data. Simple RNNs, like Elman RNNs [42], process discrete input sequences x one timestep at a time by updating their hidden state h_t with respect to the past hidden state h_{t-1} and the current input x_t:

$$h_t = \sigma\left(W^{hx}x_t + b_x + W^{hh}h_{t-1} + b_h\right) \tag{36}$$

where σ refers to a nonlinear activation function. At each timestep, a outputs $\hat{y}_t$ in the feedforward pass can be generated using another function:

$$\hat{y}_t = \sigma\left(W^{yh}h_t + b_y\right) \tag{37}$$

In this context, W^{hx}, W^{hh} and W^{yh} are the trainable weights for input, recurrence and output function respectively and b_x, b_h, b_y are the respective learnable biases.

For weight updates using backpropagation of errors, the recurrent connections between internal states of the model are "unfolded" in time. Unfolding refers to the process of making the time-steps in the network explicit, resulting in a structure that allows to update network weights analogous to feedforward networks. In particular, this unfolding involves treating the recurrent network as a feedforward network, with one layer per timestep, and weights shared between all timesteps. The resulting feedforward network can be trained using backpropagation of error in the conventional way. This algorithm is referred to as BPTT [224]

2.3 EEG PROCESSING WITH NEURAL NETWORKS

2.3.1 *Decoding auditory information*

Two different strategies have been pursued to explore the links between cognitive processes and brain activity in the neuroimaging domain. The typical procedure in forward inference is the manipulation of a subject's psychological state followed by a calculation of the probabilities of observing specific brain signals given this state, that is, a

process from psychological state to expected brain activity [63]. In contrast, reverse inference is used to reason in the other direction, that is, from brain activity to cognitive processes. Reverse inference from brain activity is a complex endeavour, especially as it requires knowledge about the information the analysed brain signals actually can account for. This is especially problematic when reasoning from specific regions of the brain, as their activation could be the result of reused activations within different cognitive processes. In reverse inference, "decoding" approaches typically model the space of brain signal as multivariate and the space of cognitive processes as univariate. In contrast, "encoding" approaches typically model a univariate brain space and a multivariate space of cognitive processes. Some methods, such as canonical correlation consider both neural and psychological space as multivariate. These multivariate methods allow to perform inference with increased accuracy, especially as they consider spatio-temporal relations across brain regions. In recent years, the use of machine-learning algorithms to decode information from brain activity has gained widespread attention and several studies have started to apply them to the analysis of audio and imagery related neural processes. Brain activity decoding in the presence of auditory imagery promises insights into the contributing cognitive functions during music listening and imagination while also shining a light on the multi-modal mapping to (imagined) sensory inputs.

A general consensus in these studies is that a listener's brain shows responses that are correlated with presented auditory stimuli and that the relationship between brain response and stimulus can be exploited to classify or reconstruct auditory stimuli. Examples for such correlations can be found in the modulation of neural oscillation magnitude and frequency by perceived tempo, rhythm and accents [22, 153]. Intracranial recordings show precise phase-locking to click train stimuli and frequency-following responses for speech stimuli [110, 150]. Auditory event-related potentialss (ERPs) refer to repeatable and distinguishable neural responses to auditory events, such as onsets or changes in pitch [184]. ERPs can be related to fine grained aspects of audio, reflecting even differences in timbre or harmonics [189]. Neural responses are modulated by individual aspects such as expertise or attention [207]. This means that methods aiming at reverse inference benefit from modelling both the subject's particular behaviour as well as taking into account the structure of auditory stimuli and environment. The idea of mapping auditory features to brain signals has found traction in a variety of studies applying machine learning to reconstruct aspects such as the loudness envelope, tempo or pitch [199, 203].

These studies are focused primarily on audio perception and show promising results, even if the quality of stimulus decoding is rather unsatisfactory. [160] describe a reconstruction of speech stimulus en-

velopes directly from recorded EEG signal based on the correlations between EEG channels and the stimulus envelope. However, as described by [201] the application of the same approach to more complex musical signal results in poor results and demonstrates the low correlation between recorded signal and auditory stimulus, which is typical for non-invasive imagining methods. A common way to deal with such low correlation is to average across a large number of trials and focus on relatively simple stimuli, a technique particularly popular in ERP based experiments [228]. Neural activity recording with EEG or functional magnetic resonance imaging (fMRI) provide relatively accurate and inexpensive access to brain signal, making it particularly interesting for machine learning. While EEG has high temporal resolution, the spatial accuracy is inferior to the three-dimensional fMRI signal. Multi-modal recording techniques like simultaneous EEG-fMRI capture multiple views on brain activity in spatially and temporally overlapping regions and allow more fine-grained analysis of the signal [86].

Deep learning, especially CNN and RNN based approaches have found use in the analysis of neural activity underlying auditory perception and imagination. They have demonstrated improved results for tasks such as tempo estimation or classification of imagined musical pieces [143, 201]. However, often times neuroimaging datasets on auditory data simply do not contain sufficient quantitative amounts data to train deep neural networks. Typical datasets in the domain contain 20 or less subjects and cover only short excerpts of musical pieces or imagined stimuli. Only very few of the available datasets contain larger amounts of subjects or provide more than one hour of recording time per subject. These more extensive datasets typically cover the perception of full songs, movies or are recorded in the context of auditory brain-computer interfacing [26, 123]. In theory, this shortage could be counteracted with more and longer recording sessions and detailed metadata.

2.3.2 *From perception to imagery decoding*

Most of aforementioned studies focus primarily on the perception of audio and music, in contrast to music imagination or combinations of both conditions [203]. However, multiple studies using EEG and MRI indicate that large parts of the neural processes underlying music perception can also be observed in absence of the actual stimulus [76, 183]. The literature on auditory imagery in the context of decoding neuroimaging data tends to deal with imagery and imagination as interchangeable terms. Often times, authors are interested in the analysis or reconstruction of mental imagery that takes place in the process of active imagination. In these cases, this lack of separation is not too problematic. However, there might be cases where a more

fine-grained differentiation is necessary, such as in auditory imagery during memory recall or active imagination.

Here we refer to auditory imagery as the process of experiencing auditory information without it being present at the senses. Furthermore, we treat auditory imagination as a process that can employ mental auditory images, such as when actively imagining a specific song. Auditory imagery generally retains the structure of actual auditory content, which is shaped by aspects such as tempo and pitch. Individual aspects of the auditory signal might be more or less accurately retained when recalling previously heard stimuli, for example, when the timbre but not the exact tempo is preserved.

Auditory imagery has been shown to be interacting with other imagery modalities, such as motor imagery. For example, being skilled in motor as well as auditory imagery has positive impact on the learning performance in the respective other domain [16]. This indicates that auditory imagery can be seen as a process that reactivates the neural pathways underlying auditory perception and recreates an internalized sensory experience. This implies that features extracted from brain signal in perception across different modalities should be useful for guiding the decoding of brain activity during imagination. Still, only a small number of studies have tackled the problem of classifying brain states in auditory imagery or reconstructing the imagined stimuli directly. For example, [65] demonstrated the possibility to use auditory imagery decoding in a Brain Computer Interface setup using EEG hardware. The study focused on the recognition of white noise imagery and reported 93% accuracy. According to the authors, these results open up the possibility to use auditory imagery as a complementary approach to motor-imagery interfaces. This aspect is especially interesting given the close connection between motor and auditory imagery. At the time of writing, there is still a lack of machine learning studies bridging between different imagery domains and exploring these inter-dependencies in greater depth.

3

SHARED REPRESENTATION OF AUDIO AND EEG

The variational autoencoder is a simple, yet highly efficient, DNN architecture that optimises VFE for a latent variable with deep neural networks and the backpropagation of error algorithm. VAEs provide a useful reference point to investigate models that optimize the computational objective of the Free Energy Principle, since they scale up to complex sensory data. Here, we want to investigate the possibility to employ VAEs as a model of canonical free energy computation in the context of learning shared representations between sensory observations, i.e. a stimulus domain and brain signals evoked by these sensory observations. In particular, we evaluate the possibility to reconstruct perceived and imagined musical stimuli from EEG recordings based on two datasets. One dataset contains multichannel EEG of subjects listening to and imagining rhythmical patterns presented both as sine wave tones and short looped spoken utterances. These utterances leverage the well-known speech-to-song illusory transformation which results in very "catchy" and easy to reproduce motifs. A second dataset provides EEG recordings for the perception of 10 full length songs. Using a multi-view VAE model we demonstrate the feasibility of learning a shared latent representation of brain activity and simple auditory concepts, such as rhythmical motifs appearing across different instrumentations. Upon qualitative inspection, the model allows continuous interpolation between representations of different observed variants of the presented stimuli and enables to reconstruct perceived and imagined music. However, our results indicate that the stimulus complexity and the quality of training data shows a strong effect on the reconstruction quality. Focusing on passive perception, we also quantitatively evaluate the proposed multi-view VAE model in terms of stimulus reconstruction performance.

The content in this section is based on the following publication:

> [pub:3] Ofner, André and Sebastian Stober (2018). "Shared Generative Representation of Auditory Concepts and EEG to Reconstruct Perceived and Imagined Music." In: *19th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 392–399.

3.1 INTRODUCTION

Studying the human brain's response to music gained a lot of attention in recent years. Many studies in the field rely on EEG recordings, as they provide better temporal resolution than other techniques,

such as functional magnetic resonance imaging functional magnetic resonance imaging (fMRI). Previous research suggests that a listener's brain response is modulated in correlation to the perceived auditory stimuli on many different levels and that these modulations can be detected within EEG. One of these effects is the correlation between the frequency and magnitude of neural oscillation patterns, which are modulated by accents and rhythmical patterns in music [22, 152, 153]. Other studies indicate that tracking auditory attention towards a specific sound source in EEG recordings is possible [6, 207].

EEG data has been used to research ERPs as a repeatable and distinguishable response to aspects of perceived music. The characteristic brain activity patterns underlying ERPs can be specific, for example, to the structure of musical events, such as note onsets or rhythm and pitch patterns [147, 185]. Other ERPs are related to the timbre of sound and can be modulated even by differences within timbre, such as changes in harmonics [136, 190]. While many ERP components show similar activation across subjects, studies suggest that some are caused by more fine-grained aspects of music, especially within trained musicians [190]. These brain activity patterns extend over the temporal, spatial and frequency domain of the EEG signal.

Motivated by the existence of such features, EEG recordings have been used in several music information retrieval studies based on EEG, such as perceived rhythm or tempo classification [203]. First attempts have been made to reconstruct the loudness envelope of perceived and imagined musical stimuli, but with unsatisfying accuracy [199, 202]. Some of these studies use deep neural networks for classification and regression. The achieved results hint at their usefulness in decoding complex brain signals. Outside of music cognition, recent studies have shown the possibility to use generative models to reconstruct perceived visual stimuli both from fMRI and EEG recordings [39, 98].

A recent study has demonstrated the possibility to learn such shared latent embeddings for EEG recordings of music perception and use them as a continuous semantic space representation of the audio [173]. This suggests that elaborate generative models could learn a shared encoding of music and brain signals, leading to a conjoint representation of auditory features and musical concepts that are perceived and processed by the brain. As previous research indicates, these concepts span a spectrum of complexity, starting on the level of the subject-specific ERP responses to low-level features, such as changes in loudness up to high-level semantic or emotional meanings of music.

3.1.1 *Auditory concepts*

Our approach relies on three assumptions for auditory concepts:

1. Coupled auditory and conceptual processing
2. Shared neural representation of music perception and imagination
3. Hierarchical structure of music

Firstly, we assume that there is a tight coupling between auditory and conceptual processing [101]. Several studies suggest that auditory stimuli are processed in a conceptual system that is shared with other modalities, such as visual perception [213]. Furthermore, music processing is based on concepts inherent to the auditory stimuli as well as on external factors, such as visual and social environment or musical training [81]. Secondly, following the ideas of embodied cognition, we assume that the human conceptual system is essentially grounded in perception and that through its interplay with action and cognitive states, music perception at least partially shares conceptual and neural representation with musical imagination [100]. Previous research suggests that auditory concept formation can be traced back to specific ERPs and that the magnitude of some ERP component can be controlled by the presence of an auditory concept in the listeners mind [205]. Thirdly, we follow the idea that music is essentially hierarchical in structure and that auditory concepts equivalently exist on a spectrum of abstraction levels, reflecting and augmenting this structure. They can range from concepts related to single sounds or rhythm to concepts within the emotional or aesthetic processing of music. Together with the previous two assumptions this means that basic elements of perceptual musical processing, such as ERPs related to note onset expectancy, are influenced by their integration into conceptual processing. Music cognition and concept formation can be highly subjective, stimulus-driven as well as context-dependent, e.g. on visual and social aspects of a performance [144]. For these reasons, we hypothesize that a simultaneous retrieval of auditory concepts from multiple sources aids the reconstruction of the processed stimuli while further deepening our understanding of music cognition.

3.2 RELATED WORK

Various approaches exist to learning a shared embedding from two or more datasets. One method is Canonical Correlation Analysis (CCA) [85]. CCA is non-probabilistic and enables the extraction of linear components to optimize the correlations between two multivariate datasets. CCA in combination with convolutional neural networks has recently been used by Raposo et al. to learn a shared semantic space between audio and EEG signal [173]. Based on CCA, Fujiwara et al. have introduced Bayesian Canonical Correlation Analysis (BCCA), a probabilistic interpretation of CCA [59]. However, BCCA still contains linear observation models, while EEG data is very complex and

noisy and requires non-linear computation. To surpass this limitation, Deep Canonically Correlated Autoencoders (DCCAE) were proposed by Wang et al. [216]. DCCAEs maximize the correlation between the latent embeddings of two separate autoencoders, but do not enable cross-reconstruction between their inputs. While this problem is solved by Correlational Neural Networks (CorrNets), the unregularized latent embeddings of both DCCAE and CorrNet are prone to overfitting, especially in combination with the representational power of non-linear observation models [18].

Inspired by [216] we use a multi-view VAE architecture that can be interpreted as a deep, generative and probabilistic latent variable interpretation of CCA, called Deep Variational Canonical Correlation Analysis (DVCCA) [216]. A similar approach tailored specifically to a missing view reconstruction for visual stimuli in fMRI data has successfully been demonstrated recently [39]. Here, we show that we can derive a general multi-view generative model capable of joint EEG and stimulus processing that allows multi-modal learning from physiological data as well as directly from the stimuli. To our knowledge, no comparable framework for EEG-based audio stimulus reconstruction or for shared auditory concept learning exists.

3.3 METHODS AND DATA

We use two datasets, the OpenMIIR speech and the Naturalistic Music EEG Dataset - Tempo (NMED-T) dataset. They are similar in experimental setup but differ in focus and size.

3.3.1 *OpenMIIR speech dataset*

One dataset contains EEG of subjects listening to and imagining four rhythmical patterns presented both as sine wave tones and short looped spoken utterances. It stems from the Open Music Imagery Information Retrieval (OpenMIIR) initiative [203] and features four different “catchy” and easy to reproduce motifs superimposed on a constant metronome click. We refer to it as "OpenMIIR speech dataset". The trials are annotated for containing either speech or sine wave tones and can be used to train and evaluate model performance for the perception and imagination of the same rhythmical trials within two timbres. The metronome clicks serve as cues that are present during perception as well as imagination. The main intention behind this dataset is to reduce stimulus complexity as far as possible while still retaining enough musical structure for building and evaluating models. This dataset contains data from seven subjects with normal hearing and no history of brain injury. It was recorded with 64 EEG channels, horizontal and vertical electrooculography (EOG) channels sampled at 512 Hz. All perception stimuli have equal tempo and du-

ration of 12 s. Presentation was done in randomized order after 2 s of metronome clicks. They were immediately followed by another 12 s of metronome cues. Participants were asked to imagine the perceived stimulus directly after presentation using these subsequent cue clicks. The concatenated perception-imagination trials sum up to 26 s of recorded EEG data for each trial. As each trial was presented 6 times, this sums up to a total of 96 presented trials. In total, the dataset contains about 2500 s (42 min) of EEG recordings per subject. We performed common-practice preprocessing steps using the MNE-python toolbox by Gramfort et al. including manual bad channel removal and interpolation after visual inspection [67]. All EEG data was bandpass filtered between 0.5 and 50 Hz. Extended Infomax independent component analysis (ICA) was used to remove EEG artifacts using the EOG signal.

3.3.2 *NMED-T dataset*

The NMED-T dataset provides EEG recordings for the perception of 10 naturalistic full length songs. The songs are in Western musical tradition, have durations between 4:30 and 5:00 min in length and contain vocals. They are real-world musical works with pronounced rhythmical properties. 125 channel EEG at 1 kHz sampling rate was recorded for all of the 20 subjects with normal hearing and no history of brain injury. We used the preprocessed version of the dataset, which features EEG down-sampled to 125 Hz and bandpass filtered between 0.3 and 50 Hz. Ocular and cardiac artifacts were removed using the additional EOG channels with ICA after manual bad channel removal. A more detailed description of the preprocessed dataset can be found in [123].

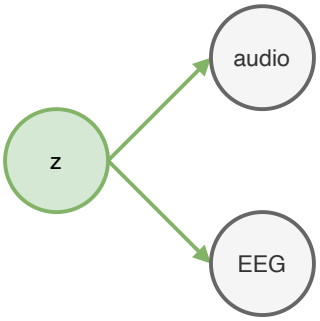
Subjects in both experiments were not required to have musical training, nor did they execute a particular task during listening or imagination. All EEG channels were normalized to zero mean and range [-1, 1]. For training, EEG data was split into excerpts of 1 s length, resulting in 512 samples (OpenMIIR) and 125 samples (NMED-T) length.

We computed Mel spectrograms of audio targets at their full sample-rate of 44100 Hz using the librosa library [135] with 64 frequency bands between 0 and 2000 Hz, fast Fourier transform (FFT) window size of 2048 and hop length of 1024. Furthermore, we generated loudness envelopes for each stimulus using Hilbert transform of the scipy library at the full sample rate [92]. We then down-sampled the Mel spectrograms and loudness envelopes to the sample rates of the EEG (512 Hz for OpenMIIR and 125 Hz for NMED-T) before splitting into excerpts of 1 s length.

3.3.3 *Learning shared representations*

We propose an adaptation of Variational Canonical Correlation Analysis (VCCA) as proposed by Wang et al. [216] to perform VAE based multi-view learning on audio and EEG signal. We start by defining EEG and audio to be two views that can be generated independently from a shared latent embedding z:

$$p(audio, eeg, z) = p(z)p(audio|z)p(EEG|z). \tag{38}$$



$$z \sim N(0, I)$$
$$audio \mid z \sim N\left(W_{audio} z, \phi_{audio}\right)$$
$$EEG \mid z \sim N\left(W_{EEG} z, \phi_{EEG}\right)$$

Figure 7: Probabilistic latent variable model of the multi-view VAE model used for shared representation learning from audio and EEG.

As we are essentially interested in the auditory information within EEG signal, we formulate a default model with a single encoder, which processes EEG. Here, z is a learnable space of auditory concepts which are contained implicitly both in the audio and the EEG signal and which generate significant parts of both views. Effectively, this projects both audio and EEG signal into the shared latent space z, while amortized inference on z is done exclusively using the EEG input. By declaring the prior $p(z)$, $p(audio \mid eeg)$, and $p(EEG \mid z)$ to be Gaussian, we ensure that the projections $E[z \mid audio]$ and $E[z \mid eeg]$ of the maximum likelihood solution are in the same space as the projections through CCA. As we deal with the reconstruction of complex EEG data, we parameterize the mean of $p_\theta(EEG \mid z)$ with DNNs and apply the same procedure for the mean of $p_\theta(audio \mid z)$. The approximate posterior $q_\phi(z \mid eeg)$ is optimized by a third DNN. The model is trained by sampling from $q_\phi(z \mid eeg)$ and optimizing the lower bound of the log likelihood $L(eeg, audio; \theta, \phi)$. Like in conventional VAEs, we optimize the reconstruction loss of audio and EEG decoder and the KL divergence between the learned $q_\phi(z \mid eeg)$ and $p(z)$ using the reparameterization trick [104].

3.3.4 *Multimodal data and additional views*

The model can be extended to arbitrary amount of decoders to reconstruct multiple views, as long as they are dependent mainly of a shared latent variable. Here, we use several decoders to reconstruct

different aspects of the audio signal: Mel spectrograms of the audio stimuli, their loudness envelope as well as an additional decoder to classify the trial types. Our main interest is the quality of the stimulus and EEG reconstructions and we use the remaining decoders only to enhance the training quality. Optionally, we could add additional encoders, by making use of additional private latent variables introduced with the VCCA model. They could provide view-specific aspects of additional input, e.g. from other modalities, such as fMRI, audio or EEG signal during imagination. Figure 8 shows the modified VCCA architecture with one EEG decoder and two audio decoders. Here, we test the model with a single EEG encoder and multiple decoders.

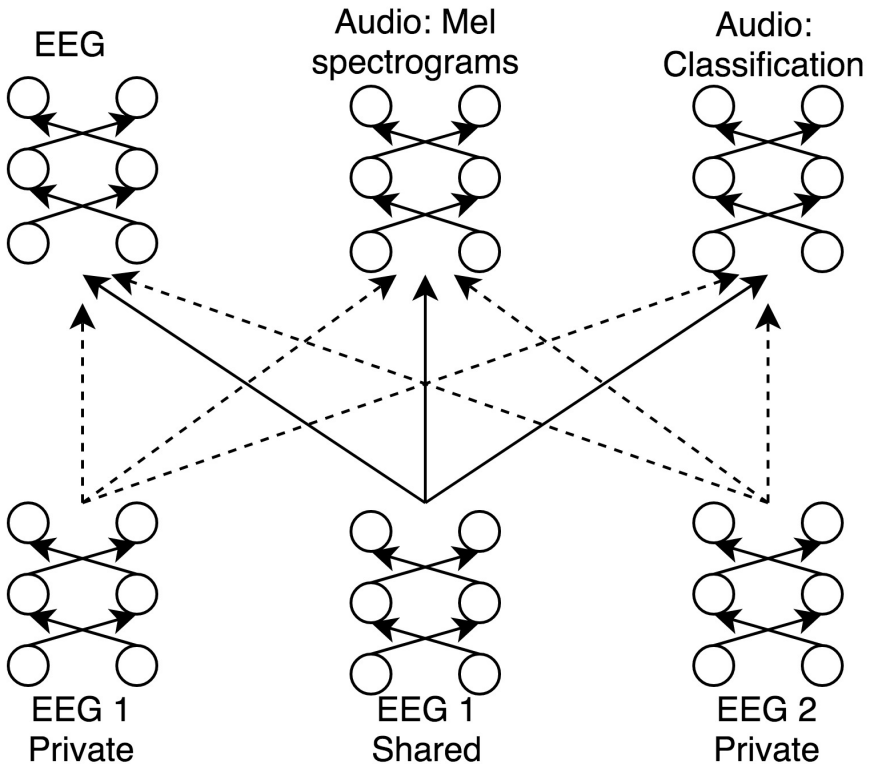


Figure 8: Model architecture for the proposed multi-view model from Figure 7. Latent variables parameterized by optional private encoders are indicated with dashed lines.

3.3.5 *EEG encoder architecture*

Both NMED-T and OpenMIIR speech EEG encoders featured 4 convolutional layers with filter numbers linearly ascending from 64 to 512 per layer. Convolution was performed on two dimensional inputs. Each column of the input represented the same linear concatenation of EEG channels for a single sample within the inputs of 1 s length. This resulted in inputs of size 512*64 for the OpenMIIR speech and

125*125 channels for the NMED-T inputs. The kernel size was set to [2x2] for all layers. Here and for all further kernel dimensions, we define the first index to be within the channel domain (or frequency for spectrograms) and the second within the temporal domain. Each convolutional layer was followed by 30 % dropout.

3.3.6 *EEG and audio decoder architectures*

We used similar EEG decoder architectures for both datasets. The OpenMIIR speech EEG decoder featured 6 hidden deconvolution layers with three layers of 16 and another three layers of 32 filters. The kernel size was set uniformly to [2x16] with stride 2 except for a [2x1] kernel in the third layer with stride 1. A final dense output layer consisted of 512*64 units. The decoder for the NMED-T dataset followed the same deconvolution architecture, except for kernels with dimension of [4x16] and [4x1] instead of [2x16] and [2x1]. A final dense layer consisted of 125*125 units. Both OpenMIIR speech and NMED-T decoders for Mel spectrograms consisted of four layers: Two deconvolution layers of 32 filters and two layers with 64 filters. As the length of Mel spectrograms mirrors those of the EEG excerpts, but in combination with a frequency resolution of 64 bins, the final dense layer featured 512*64 and 125*64 units respectively. The kernel dimensions were set to [4x8] uniformly, except for the fourth deconvolution layer of the OpenMIIR speech decoder, with a [2x8] kernel. The decoder for loudness envelope reconstruction consisted of a bidirectional LSTM layer with 128 hidden units, followed by a dense layer of size equal to the length of the audio excerpt. Finally, the decoder used for classification of the OpenMIIR speech dataset consisted of two hidden dense layers with 32 filters and a dense output layer of 1 unit. All internal units used ReLU activations, all output units had sigmoid activation. The size of the latent embedding was 128 units.

3.4 EXPERIMENT

3.4.1 *Model training and prediction*

The model was trained both intra-subject and cross-subject in an end-to-end fashion purely on the perception trials using Adam optimization with a constant learning rate of 0.0001 [103]. For both datasets we used 60 % of available perception trials for training and another 20 % for validation. The remaining 20 % and the imagination trials were used for testing. All trials were shuffled randomly before training. For tests on imagination data, we evaluated both imagination trials whose corresponding perception trials were included in the training as well as entirely unknown trials. All models were trained up to convergence of the Mel spectrogram reconstruction loss, between 1000-

2000 epochs. Reconstruction loss was computed as the mean squared error between reconstructions and targets.

3.4.2 *Introspection*

After training we inspected the learned latent space by linearly interpolating between multiple existing EEG inputs extracted either from the training or testing dataset. This way, we received embeddings for the given inputs as well as a fixed number of embeddings that connect them in the learned projection space. We then used the model to reconstruct the Mel spectrogram and EEG signal for the embeddings.

3.5 RESULTS

3.5.1 *Perceived stimulus reconstruction*

We were able to use the model to reconstruct the Mel spectrograms of perceived audio within both datasets at various levels of accuracy. Figure 9 shows reconstructions of speech and sine wave tone patterns for intra-subject training and testing on both trial types of the OpenMIIR speech dataset. The reconstructions are characterized by rhythmical and timbral alignment with the target. In some cases we noticed erroneous temporal shifts of the whole predicted rhythmical pattern within a reconstructed excerpt. Additional tests with smaller window sizes lead to a decrease in amount and size of such errors, while increasing the amount of false positive predictions of both sine wave and speech patterns. In some cases speech and sine wave patterns were mixed up, but still with correct temporal alignment of note onset positions between target and predictions. Figure 10 shows reconstructions after training on all subjects of the OpenMIIR speech dataset. Multi-subject training lead to results with improved temporal alignment of targets and predictions. Here, in more cases the two timbres (sine wave and speech pattern) were confused. This indicates that the correct prediction of the timbre is more subject-specific than the temporal and rhythmical aspects. Increasing the amount of training data for both trials enhanced the overall reconstruction quality, training only on the speech trials still lead to temporally meaningful reconstructions of the sine wave tone patterns. We found the stimulus reconstruction quality to be best when including 4 subjects for cross-subject training and testing.

Increasing the amount of dropout within the EEG decoder (up to 40 %) turned out to be beneficial for reconstructions of comparable quality for trials in subjects that were excluded entirely from the training procedure. Training with randomized window start positions and using overlapping overlapping temporal windows proved to enhance the reconstruction quality. This indicates that the spectrogram recon-

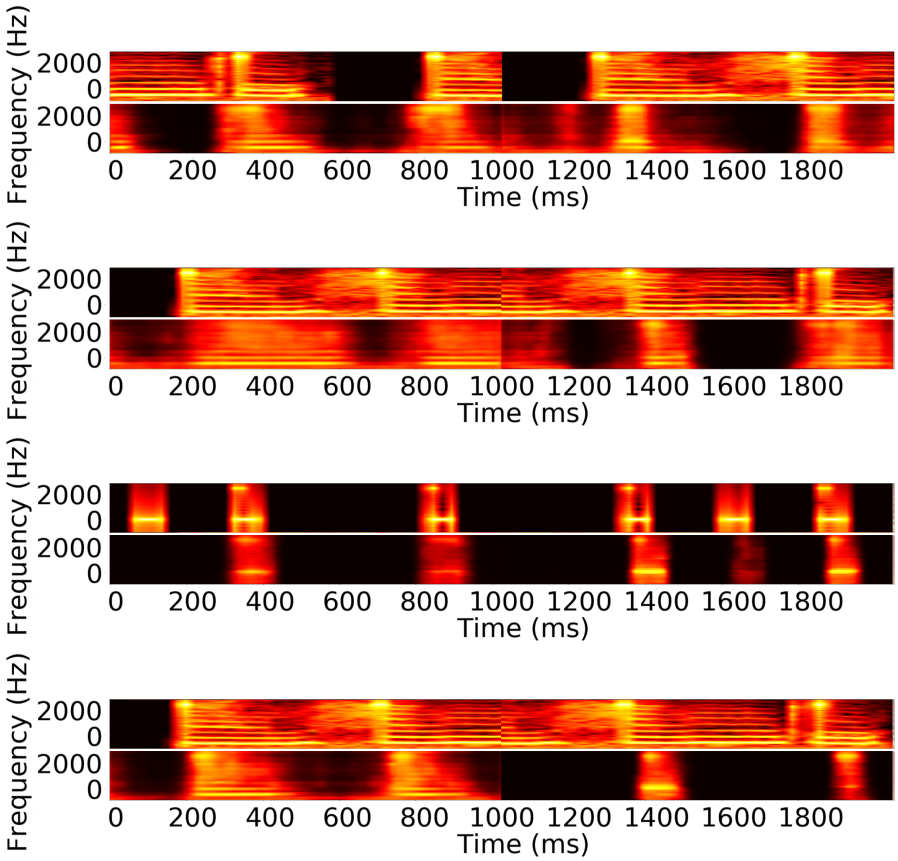


Figure 9: Mel spectrogram reconstructions of perceived rhythmical trials for model trained on subject 'P13' of the OpenMIIR speech dataset. Target stimuli are presented above their reconstructions.

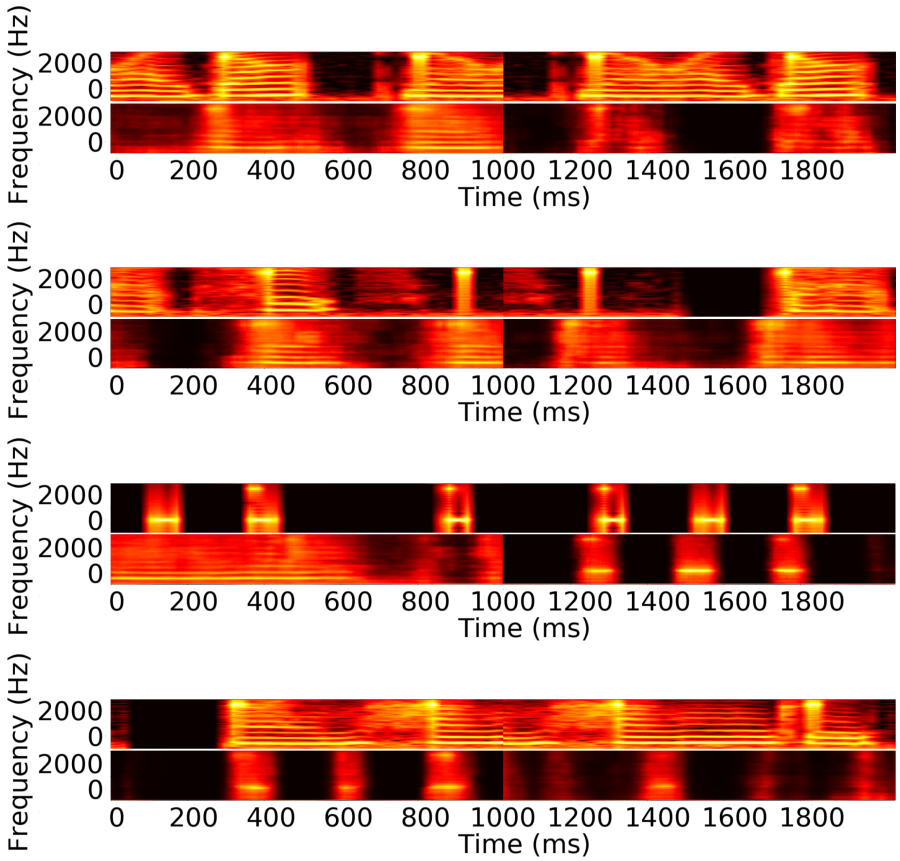


Figure 10: Mel spectrogram reconstructions of perceived rhythmical trials for model trained on all subjects of the OpenMIIR speech dataset. Target stimuli are presented above their reconstructions.

struction quality for this dataset is limited by the amount of available training data suggests the use of more elaborate data augmentation techniques or increased data set sizes in future work.

Compared to the OpenMIIR dataset, the NMED-T dataset provided more training data with increased target complexity. The reconstructions showed different characteristic in visual inspection. Often times, the timbre reconstruction dominated the reconstruction of temporal aspects, especially in parts that featured multiple instruments or singing voice. In fewer cases, but within all songs, noticeable onsets of percussion, speech or other sounds were reconstructed. For all trained models, timbre reconstruction was visible after around 500 epochs, while temporal aspects were learned at later stages. Figure 11 provides examples for reconstructed excerpts of the perceived full-length songs contained in the NMED-T dataset. We found no substantial difference in the quality of reconstructions within subjects included into training and those from subjects excluded during training. This might be due to the small amount and long duration of 10 stimuli in combination with a single presentation per stimulus. Increasing the dropout rate after each convolutional layer in the EEG encoder

over 30 % increased the models tendency to reconstruct temporal aspects, such as percussion onsets. Training sets with a larger amount of subjects generally improved reconstruction quality. Furthermore, the introduction of overlapping EEG input windows increased the amount of reconstructed temporal features. Models trained for more than 2000 epochs showed more sparse reconstruction within the frequency domain. This indicates that adding more data and increasing training length can further increase the reconstruction quality for naturalistic music. Often times, the size of temporal misalignment was equal at all positions within reconstructed excerpts. This indicates that the reconstruction quality is dependent of the window size. Future work could test this assumption by simultaneously training on EEG or audio excerpts of various sizes within different encoders of the model. This would furthermore allow the representation of the latent concepts to include contexts of various size. For example, in the audio domain, such contexts could range from single note onsets to changes in song structure.

3.5.2 *Imagined stimulus reconstruction*

VCCA models trained on perceptual OpenMIIR speech data could be applied to imagination trial reconstruction. The reconstructed stimuli showed the same typical rhythmical patterns and could be divided into speech and sine wave predictions. However, the correct rhythmical predictions were less often visible and more blurry. It is important to note that the imagination was performed superimposed on a constant metronome click. This means that only the difference between the rhythmical structure and timbre was based on imaginative processes, while there were still perceptual cues for temporal alignment. Models trained on multi-subject perceptual data showed less blurry reconstructions. Adding private encoders with imagination based EEG signal did not cause a visible increase in reconstruction quality.

3.5.3 *Qualitative analysis of learned auditory concepts*

We found musically meaningful representations of the OpenMIIR speech stimuli in the latent space of models trained intra-subject as well as cross-subject. Both EEG signal from training and testing subsets could be used to produce continuous interpolation. Processing EEG inputs from both testing and training data sets and using the target audio stimuli as validation, we found continuous representation across the temporal, rhythmical and timbral domain. For any given input, we could change the temporal position of the rhythmical pattern as well as the timbre (within speech and sine wave tones). Furthermore, the latent space enabled interpolation between

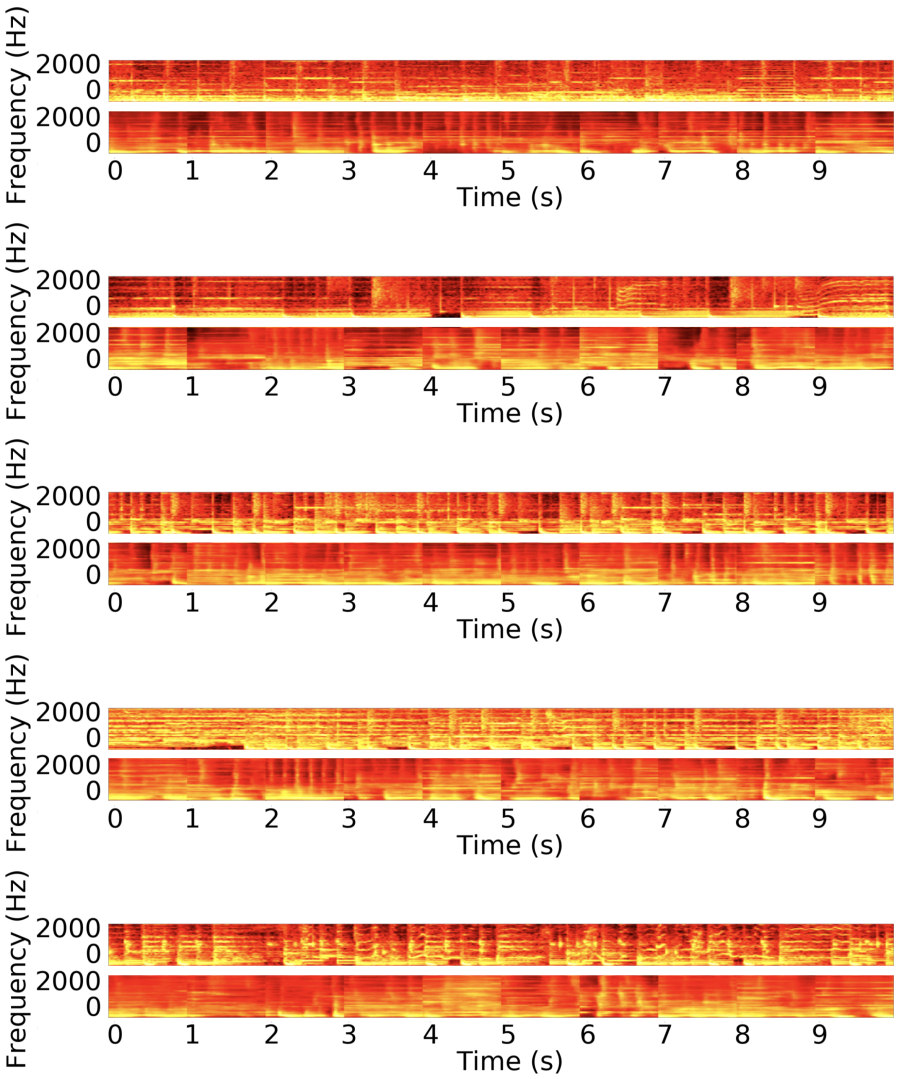


Figure 11: Excerpts of reconstructed Mel spectrograms from the NMED-T dataset. The target stimuli are shown above their reconstructions. The two top rows are based on training on all subjects. The three bottom rows are based on training on 10 subjects and testing on subjects that were excluded during training.

metronome clicks and the superimposed sine wave tones, although with decreased clarity on the test set. Figure 12 (a) shows an example for the interpolation between 3 embeddings based on EEG inputs of the OpenMIIR speech training data set. Here, interpolation between a syncopated and non-syncopated part of the rhythm was done while simultaneously shifting the temporal position of the rhythmical pattern within the reconstructed excerpt. The non-syncopated excerpt was further interpolated into its representation with speech signal. Figure 12 (b) shows topographic projections of the brain activity reconstructed for each embedding that was computed in Subfigure (a). For the sake of clarity we show six topographic plots out of the total amount of 512 per embedding. Qualitative comparison of the EEG signal with the original inputs indicated that overfitting the EEG data is not possible when we stop training when the audio reconstruction loss is saturated. For other use cases, higher quality EEG reconstructions could be achieved with different training procedures, such as unsupervised EEG reconstruction pretraining. Models with smaller latent embeddings sizes (down to 8 latent units) did still produce meaningful and continuous interpolations, but with more blurring across the temporal and frequency domains.

3.6 TRAINING ON AVERAGED EEG DATA

The previous sections covered a mostly qualitative evaluation of the reconstructed stimuli and the structure of the learned latent space. We found that a reduction of encoder and decoder complexity still leads to meaningful results, while reducing the need to regularize the encoder model via high dropout rates. For this reason, in this section we present a simplified architecture with less parameters that features only two decoder networks (for EEG and audio) and report results for the OpenMIIR and NMED-T dataset. In particular, we focus on training with averaged EEG inputs and the influence of the relative and absolute weighting of the reconstruction losses and the size of the latent space. For this, we evaluate the model on EEG signal after averaging across subjects (within individual trials) or across trials (within the same subject). This averaging of temporally aligned EEG data is a common technique in the analysis of evoked responses [77, 147, 185, 190]. Next to analysing model predictions by averaging over inputs, we also include averaged EEG data in the training set, under the hypothesis that it improves the generalization of the model. For model evaluation we resort to the mean squared error $\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ between the true data y and predictions $\hat{y}$. Additionally, in order to assess the perceptual quality of reconstruction in terms of texture, we resort to the structural similarity index measure (SSIM) with a window size of 7 [218].

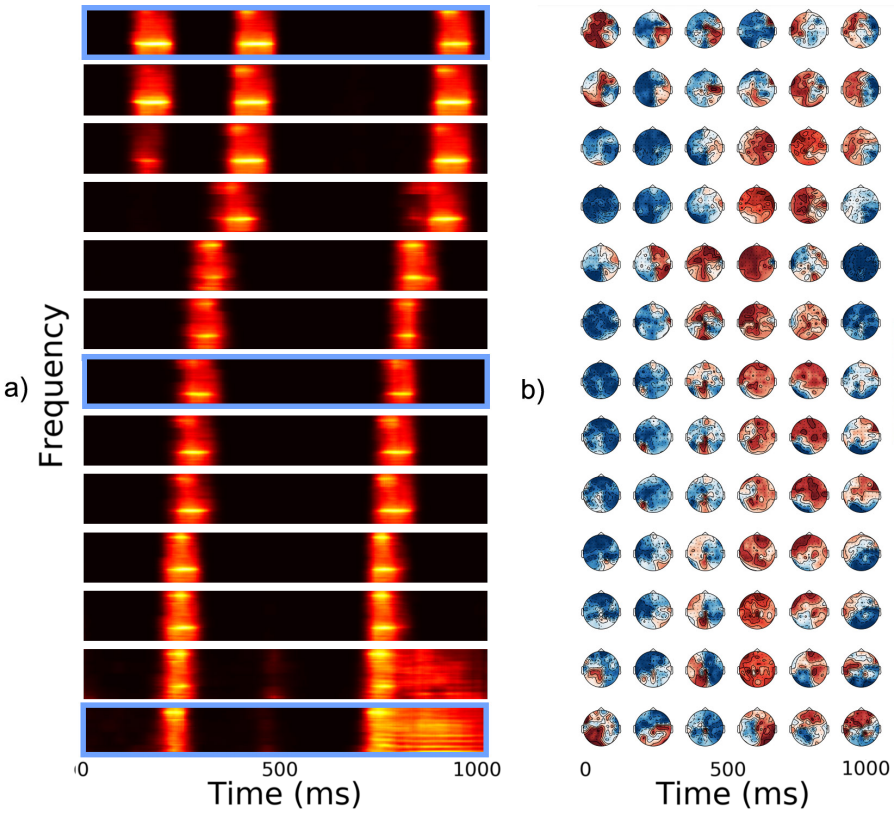


Figure 12: (a) Reconstructed Mel spectrograms after interpolation in the learned latent space learned for Subject 'P13' of the OpenMIIR speech dataset. Embeddings that correspond to real EEG inputs are indicated with blue frames. (b) Topographic visualization of the reconstructed temporal brain activity. Each row represents the brain activity reconstructed for the embedding in the same row of Subfigure (a).

3.6.0.1 *Model structure and training*

The simplified model uses similar decoders for both OpenMIIR and NMED-T dataset: A dense layer projects the latent state (with size 32 or 128) to a hidden dimension of [1x512], followed by 4 convolutional layers, featuring 64 and 32 filters of size [4x2], respectively. A final convolutional layer with the same filter size maps the output towards a single channel prediction. All hidden layers used a ReLU activation, followed by a Sigmoid at the output layer. We adjusted for the differently sized EEG shape in the OpenMIIR dataset by simply cropping the respective decoder's output. The encoders for both datasets featured six convolutional layers, each with a filter size of [4,2]. Again, the hidden layers are ReLU activated. The final convolutional layer maps its input to either 1024 units (NMED-T) or 512 units (OpenMIIR). For the NMED-T dataset, we used linearly ascending channels, ranging from 32 at the input layer to 512 at the fifth layer. The OpenMIIR model was significantly smaller and used two convolutional layers with 32 followed by another three convolutional layers with 64 channels. For the following experiments we did not use dropout in encoder or decoder networks. We trained all models for $3e^5$ steps using the Adam optimiser at a learning rate of $1e^{-4}$ and a batch size of 64.

3.6.1 *Results on NMED-T*

We trained the larger of the two models on the NMED-T dataset after averaging the input EEG signal over subjects, keeping the separation into songs (i.e. intra-subject trials) intact. We held out every tenth input window from the training set, resulting in a test set that covers portions of the entire song. We will refer to this test set split as "windowed test set". We also held out the final 10% of all songs, resulting in a second subset of the dataset that we refer to as "continuous test set". It should be noted that averaging across subject further decreases the size of the dataset, but possibly provides inputs that are easier to process, due to the elimination of noise in the signal. Table 1 provides an overview of the results on continuous and windowed test set for a model with 32 latent units.

Table 2 lists the same quantities for a larger model with 128 latent units. Both report the mean and standard deviation of three independent runs with randomly initialised weights and biases. We found that the variant with a higher weight $\gamma_{EEG} = 10$ for the EEG reconstruction loss consistently scored better in terms of mean squared error (MSE) on the EEG reconstruction and SSIM scores for both audio and EEG (while the MSE on audio generally decreased). This indicates that increasing the weighting of the EEG reconstruction term in comparison to the KL regularization term and the audio input

		z = 32			
γ_{EEG}	split	MSE_{audio}	MSE_{EEG}	$SSIM_{audio}$	$SSIM_{EEG}$
1.0	C	**0.027±0.016**	0.009±0.001	0.184±0.054	0.242±0.036
10.0	C	0.029±0.018	**0.007±0.001**	**0.194±0.051**	**0.284±0.039**
1.0	W	**0.03±0.027**	0.008±0.001	0.178±0.055	0.256±0.039
10.0	W	0.032±0.026	**0.007±0.001**	**0.183±0.048**	**0.3±0.042**

Table 1: Mean squared error (MSE) and Structural Similarity (SSIM) results on the continuous and windowed test sets of the NMEDT-T dataset for a model with 32 latent units. Reported are mean and standard deviation for three runs.

generally improves the learned representations. Performance generally degraded when raising the γ weighting of audio reconstructions instead of EEG reconstructions. We found that training models the significantly longer than $3e^5$ weight updates, e.g. up to $1e^6$ steps does still increase reconstruction quality on the training set, but does not improve the test results, i.e. the model overfits the training data. We found that, in terms of MSE and SSIM accuracy, the audio reconstruction works slightly better on the continuous test set compared to the windowed test set. A possible explanation for this difference is the lack of familiarity and entrainment of the human brain response at the beginning of the songs. Since the continuous test set is extracted from the end of each song, repetitive aspects of the song are already known. Another possible explanation is the simpler musical structure towards the end of pop songs, such as included in the NMED-T dataset, which often include a "fade-out" with reduced instrumentation towards the end. Still, the overall differences between model variants is consistent between both test sets.

		z = 128			
γ_{EEG}	split	MSE_{audio}	MSE_{EEG}	$SSIM_{audio}$	$SSIM_{EEG}$
1.0	C	0.027±0.019	0.008±0.001	0.189±0.061	0.243±0.036
10.0	C	**0.026±0.018**	**0.005±0.001**	**0.192±0.055**	**0.472±0.035**
1.0	W	**0.03±0.025**	0.008±0.001	0.177±0.06	0.258±0.039
10.0	W	0.03±0.028	**0.005±0.0**	**0.185±0.054**	**0.485±0.036**

Table 2: Mean squared error (MSE) and Structural Similarity (SSIM) results on the continuous and windowed test sets of the NMEDT-T dataset for a model with 128 latent units. Reported are mean and standard deviation for three runs.

3.6.2 *Results on OpenMIIR*

We also trained and evaluated the previously best performing model (with 128 latent units $\gamma_{EEG} = 10$) on the OpenMIIR speech dataset, by including averaged and individual data in the training set. Here we focused on the trials of the perception condition. Again, we we held out the final 10% of all trials for testing. The OpenMIIR dataset has significantly shorter trial lengths. In order to guarantee as much training data as possible, we did not reserve windows throughout the entire trial for testing. We trained two variants of the model: One variant that is trained exclusively EEG signals on a per-subject and per-trial basis, i.e. without any averaging of the inputs. The second variant was trained on an extended training set that included averaged EEG inputs representing the mean across subjects (keeping separate trials) and the mean across trials (keeping individual subjects).

		z = 128			
Test	Model	MSE_{audio}	MSE_{EEG}	$SSIM_{audio}$	$SSIM_{EEG}$
S	S	**0.044±0.025**	**0.011±0.003**	0.216±0.126	**0.716±0.058**
S	S&M	0.046±0.027	0.012±0.003	**0.229±0.141**	0.703±0.057
M	S	0.044±0.028	**0.003±0.0**	0.175±0.083	**0.718±0.02**
M	S&M	**0.042±0.027**	0.004±0.0	**0.265±0.175**	0.701±0.019

Table 3: Mean squared error (MSE) and Structural Similarity (SSIM) results on the test set of the OpenMIIR dataset. The test set is divided into reconstructions from per-subject and per-trial EEG (S) and from EEG inputs after averaging across subjects (M). The model is trained either with (S&M) or without (S) including EEG data that has been averaged across subjects. Reported are mean and standard deviation for three runs.

Table 3 reports the corresponding MSE and SSIM scores on the OpenMIIR test set with respect to mean and standard deviation of three runs with randomly initialised model parameters. Here, we report results on two subsets of the training data: A subset with per-subject and per-trial EEG (S) and another subset with EEG inputs after averaging across subjects (M). Similarly, we denote the model trained on the extended dataset with $model_{S\&M}$ and the model without averaged training data as $model_S$. As expected, the accuracy of audio reconstruction from $model_{S\&M}$ improve over $model_S$ on the test set with mean EEG. The EEG reconstruction accuracy however was slightly better for $model_S$ on both subsets of the test set. Interestingly, the audio SSIM score of $model_{S\&M}$ improved over $model_S$ on the per-subject test set (S). This improvement in audio SSIM however is in contrast to a worse performance with respect to the MSE for this subset.

3.6.2.1 *Reconstructions from averaged EEG*

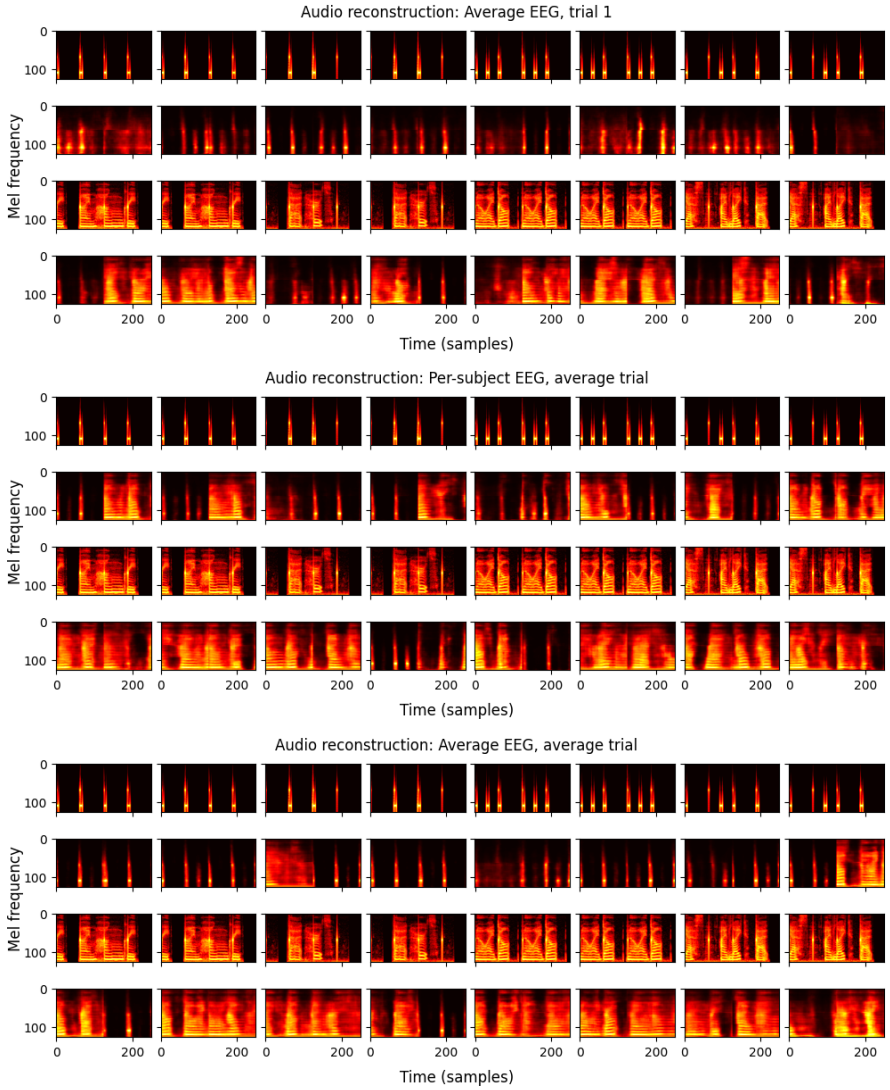


Figure 13: Reconstructing audio stimuli from averaged EEG inputs after training the model the OpenMIIR dataset with additional averaged EEG data. Shown are all 16 different trial types, i.e. 8 rhythmic (top row) and the corresponding 8 speech trials (bottom row). EEG inputs are either averaged across subjects, over trials or across both dimensions.

After evaluating the influence of adding averaged EEG data to the OpenMIIR training set quantitatively, we now qualitatively inspect the predictions made from averaged EEG inputs. Here we report results from the model trained on individual and averaged EEG inputs. Additional results from a model trained without averaged EEG data can be found Figure 46 in the Appendix. In this context, we infer EEG inputs that have been averaged across the following the subject and trial domain:

1. average subject and per-trial EEG
2. per-subject and average trial EEG
3. average subject and average trial EEG

Figure 13 shows the corresponding reconstructions from the model. Shown are all 16 different trial types, i.e. 8 rhythmic and the corresponding 8 speech trials. All three evaluated conditions showed a separation between rhythmic and speech trials in terms of stimulus sparsity in the timbral domain. A significant difference is noticeable between the reconstructions made from averaged EEG (keeping individual trials separated) and those from averaged trials (keeping individual subjects separated). In the second condition, significantly more speech-like reconstructions were assigned to EEG inputs from rhythmic trials. In contrast, the temporal alignment of speech reconstructions was visually improved in the second condition. Averaging over both subjects and trials led to more pronounced reconstructions with increased sparsity especially for the rhythmic trials. Upon visual inspection, in this configuration the temporal alignment further increased. Interestingly, in some cases a temporally aligned rhythmic reconstruction was replaced with a speech reconstruction. This indicates that the model aligns stimulus reconstructions with onsets detected in the EEG input, even if the timbral reconstruction sometimes fails entirely.

3.6.2.2 *Averaging audio reconstructions*

Next to averaging the inputs to the model and evaluating the corresponding reconstructions, we can also average the audio reconstructions themselves. In this context we first infer audio reconstructions from

1. per-subject and per-trial EEG
2. average subject and per-trial EEG
3. per-subject and average trial EEG

and average over audio reconstructions subsequently. Figure 14 shows the corresponding averaged audio reconstructions from the model. In contrast to the previous section, the audio reconstructions now refer to the mean across multiple reconstructions instead of a single reconstruction from an averaged input. We found that the included variance of multiple audio reconstructions was reflected strongly in the mean. In particular, for per-subject and per-trial predictions, the corresponding reconstructions showed strong temporal smearing in the timbral domain, although a temporal alignment with the metronome clicks was still noticeable. This indicates that the model generates predictions with a large variance between trials. The

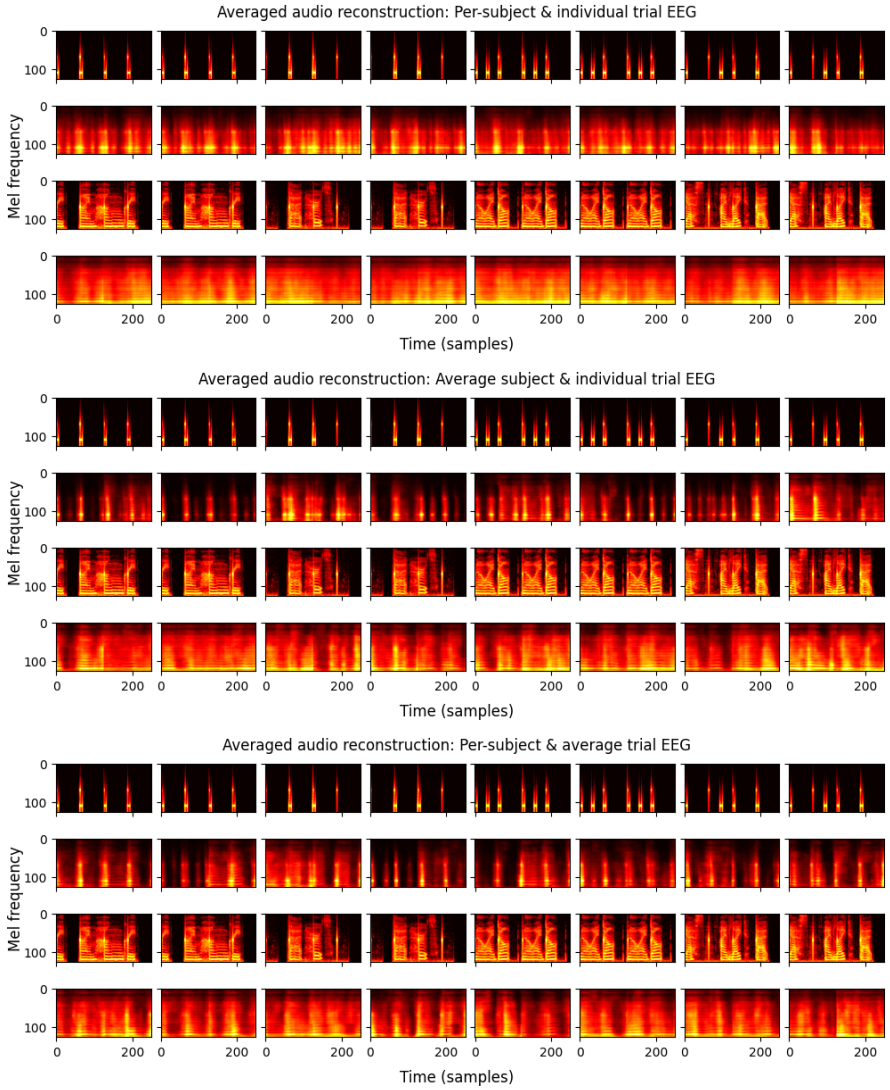


Figure 14: Averaged audio reconstructions from a model trained on the OpenMIIR dataset with additional averaged EEG data. First, predictions are made from individual EEG inputs or from EEG that has been averaged across subjects (within the same trial) or across trials (within the same subject). After inference, the mean of the audio reconstructions is computed for each trial.

temporal smearing was reduced for averaged EEG inputs. Averaging across subjects or across trials leads to reconstructions that reflect the variance in only one of these domains. As visible in Figure 14, the corresponding reconstructions show an increased sparsity and a more pronounced temporal alignment to the target stimuli, although worse than the individual reconstructions presented in the previous section. A visualization of the results from the model trained without averaged EEG data can be found in Figure 45 in the Appendix.

3.7 DISCUSSION

Despite having only a single hierarchical latent variable, VAEs are a useful model to approach shared representation learning across perceptual signals and related evoked responses in the human brain. From the perspective of Bayesian inference, the evaluated model learns a joint generative model, where surprise connected to sensory data y_{eeg} is minimized jointly with surprise about the corresponding brain signal y_{eeg} under a hypothesis generated from the joint latent distribution z. Our results indicate that learning such joint representations, although a relatively naive approach, allows to retrieve structured information about sensory data from complex and noisy brain signal.

Throughout the conducted experiments, EEG and audio signals (due to their two-dimensional representation as mel spectrograms) are processed using deep neural networks that process spatial aspects of the input explicitly. This allows for efficient processing using convolutions. From a modelling perspective, the expressiveness of a unit-wise squared-error signal is relatively limited, especially when the variance of the decoder network is ignored, which is a simplification taken in many VAE based studies [104, 178]. Such a unit-wise reconstruction loss considers only the relative difference between the predicted and observed mean, without considering (expected) uncertainty. EEG signals contain significant amounts of noise, primarily due to the recorded signal itself being a complex signal from many different neural sources and due the additional noise induced by the physical electrodes being attached to the skin. Estimating the uncertainty at the output of the VAE's decoders (i.e. independently for each modality) is possible, but increases model complexity.

This means that the presented model might be limited by the simplified approach to temporal processing. Treating the temporal axis of inputs (within temporal windows of fixed size) like a spatial domain allows to use a static model. From a Bayesian filtering perspective, the optimization of model weights is usually done with respect to integrals over time, i.e. with respect to temporal averages over multiple inference steps. Inference on temporally changing states, however, is usually expressed in terms of inference on individual (discrete) timesteps, such as in the Kalman filter or in recurrent neural networks [95, 177]. This means that from a high level perspective, the static approach taken here is meaningful in terms of overall parameter learning, but makes a strong simplification regarding temporal inference. If we interpret the brain as Bayesian filter under the FEP, temporal inference is usually causal with respect to the temporal past [49, 51]. This means that surprise (in terms of VFE) about new information (e.g. the onset of new note) is a function that depends on the stimulus in the past, as well as top-down expectations. However, since

we restrict inference to be conditioned only on the brain signal and process entire sequences of data, we can disregard such temporally causal processing in the stimulus domain and still recover meaningful predictions.

3.8 SUMMARY

In this chapter, we presented the application of a multi-view VAE model to shared auditory concept learning and musical stimulus reconstruction from EEG signals. We showed that the model can learn representations of simple rhythm and timbre related concepts that are shared in audio and EEG data. Furthermore, we could see first steps towards naturalistic music and imagined stimulus reconstruction. The presented experiments provide insights on the application of free energy optimisation as a canonical computational motif to shared processing in humans and machines, with a focus on perceptual representations. The discussed multi-view framework is designed to be expandable to additional modalities, such as fMRI data, or additional reconstruction targets, such as emotional aspects of music cognition. In combination with the ability to perform introspection on the shared representation of stimuli and electrophysiological responses, the model can be an aid for future EEG based music information retrieval and research in music cognition. The following chapters will focus on more elaborate models under the FEP that address temporal and hierarchical inference in greater depth.

4

PREDNET AND PREDICTIVE CODING

In the domain of deep learning, an influential PC inspired DNN architecture has been proposed by William Lotter and colleagues, that focuses on unsupervised future frame prediction in video frames [124]. Due to state-of-the art performance on several video prediction tasks, PredNet has received an substantial amount of attention since its publication. Several works have introduced changes to the architecture to increase performance, or to adapt the model in domains outside video prediction, e.g. in robotics. With respect to the model architecture, PredNet combines several relevant structural motifs that are present in hierarchical PC networks in neuroscience, such as layer-wise error computation and a top-down pathway. As we will see in this chapter, it is difficult to map the underlying generative model to the established PC process models in neuroscience. Nevertheless, PredNet has been shown to reproduce a set of cognitive phenomena, such as illusory motions, making it an interesting and performant baseline for PC based models in machine learning. In this chapter, we provide a review of PredNet from the perspective of hierarchical PC in neuroscience. In this context, we first compare PredNet to the structure of predictive coding networks. We then quantitatively investigate the role of top-down information in the hierarchical model, by including a label classification module as a modification to the network.

The content and figures in this chapter is based on the following publication:

> [pub:7] R. P. Rane, E. Szügyi, V. Saxena, A. Ofner, and S. Stober. "Prednet and predictive coding: A critical review." In: *Proceedings of the 2020 International Conference on Multimedia Retrieval (ICMR)*. 2020, pp. 233–241

The paper builds upon methods developed within a student project by Roshan Prakash Rane, Edit Szügyi and Vageesh Saxena which has been directed and supervised by thesis author. All three students contributed equally to the implementation and analysis for the described experiments and were involved in writing the paper. This chapter summarizes their work and results, meaning that most credit goes to them.

STRUCTURE OF THE CHAPTER

The rest of this chapter is organized as follows:

Section 4.1 briefly introduces core components of the PredNet architecture and provides a comparison to hierarchical and dynamical predictive coding in neuroscience. Section 4.2 covers related work based on deep predictive coding. Section 4.3 introduces the employed datasets and explains how PredNet can be evaluated with respect to a conditioning on top-down labels. Section 4.5 presents quantitative and qualitative results for PredNet and the proposed variant with top-down conditioning. Finally, section 4.6 discusses the results and summarizes the insights of this chapter.

4.1 INTRODUCTION

PredNet is an influential video prediction architecture that models a hierarchy of recurrent representations in a deterministic DNN architecture [124]. A core aspect of PredNet is a hierarchical, i.e. layer-wise prediction of the prediction error of the lower layer. Each hierarchical layer in PredNet i consists of recurrent representations R_i that generate local predictions $\hat{A}_i$ using a convolutional prediction unit. The prediction in each layer is compared to the output of an input convolution unit A_i, that applies a convolution and pooling operation to the layer's input, thereby down-sampling it spatially. Each layer computes errors E_i as the difference between prediction and observation. Crucially, the input to A_i for the lowest hierarchical layer is the observation o, while the prediction errors E_i are propagated upwards and provide the input A_{i+1} to the next layer. This means that only the lowest hierarchical layer predicts data, while hidden hierarchical layers predict prediction errors. Next to this bottom-up flow of prediction errors, the representation R_i in each layer additionally is provided with a top-down source of information that stems from the recurrent representation R_{i+1} of the next higher layer and involves a spatial upsampling step. The prediction error E_i within a layer is furthermore passed to the representation units R_i of that layer. In PredNet, the recurrent units are implemented as convolutional LSTMs (ConvLSTMs) [230]. In combination with convolutional prediction and input networks, the architecture is particularly suited for spatio-temporal data.

The layer-wise prediction of *prediction errors* is in contrast to predicting the *states* of the respective layer in hierarchical PC models in neuroscience. While this difference might seem subtle at first, it makes it quite difficult to interpret the generative model that underlies PredNet. Figure 15 shows a schematic comparison between the PredNet architecture and the connectivity required for hierarchical predictive coding in neuroscience.

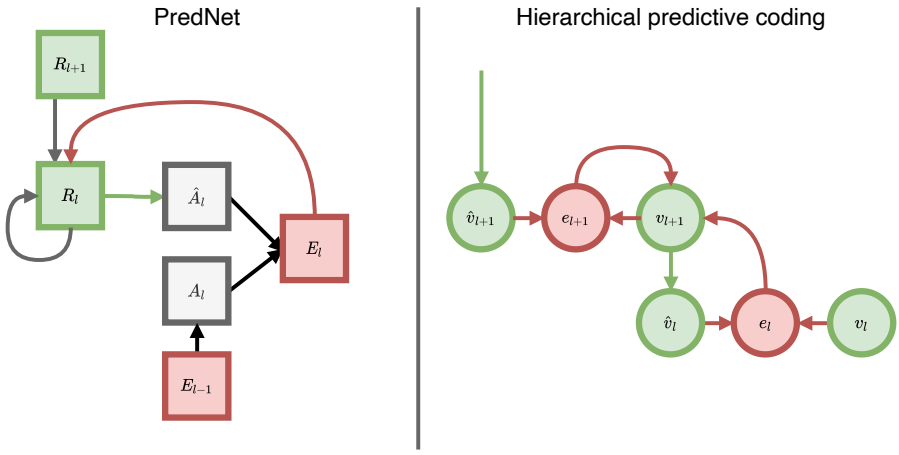


Figure 15: A comparison of PredNet [124] and the structure of hierarchical PC networks in neuroscience [49, 51]. Shown are deterministic nodes (square) and probabilistic nodes (round) as well as the connectivity between two neighboring hierarchical layers. In PredNet, differences between predicted and observed error signals are propagated, while hierarchical PC propagates the error signal between predicted and expected states.

4.2 RELATED WORK

Recent years have seen the development of several DNN models that are inspired by the working principles of predictive coding [71, 125, 212]. Often, this class of models is referred to as "deep predictive coding". These deep predictive coding models include object recognition from static inputs [223] or include recurrence for local recurrent processing of encoded representations [71]. There has been work focusing specifically on contrastive predictive learning, e.g. CPC [212] in the context of an autoregressive model, where the focus lies on learning informative representations in latent space. Next to the PredNet architecture itself, the model evaluated in detail in this chapter, there exist several modifications using PredNet as a starting point. These include AFA-PredNet, which extends PredNet with motor actions that modulate the generative model [234]. The same authors proposed another variant, MTA-PredNet, which models various temporal scales along the hierarchy [235]. Similarly, there have been attempts at making the baseline PredNet architecture more efficient by adding skip-connectivity or to reducing the complexity of the gating process in the recurrent units [43, 181]. PredNet has been evaluated in a variety of contexts, such as weather precipitation, autonomous driving or robotics applications [181, 235]. Interestingly, PredNet's spatio-temporal predictions have been shown to be susceptible to visual illusions, similar to human cognition [221]. Nevertheless, none of these studies specifically reviews and analyses the model from the

perspective of hierarchical predictive coding as a process model in neuroscience.

4.3 METHODS AND DATA

4.3.1 *The something-something dataset*

For the following experiments conducted in their project work, Prakash Rane, Edit Szügyi and Vageesh Saxena resort to the something-something dataset, a large scale video classification dataset covering humans executing actions in the context of everyday objects [66]. The dataset is crowd-sourced and includes a large variety of the noise underlying real world actions, including thousands of different objects, various lighting and background conditions and camera motion. In contrast to related datasets, which usually include relatively coarse-grained labels, the something-something dataset offers fine-grained action labels [73, 97, 112]. Using coarse-grained actions often allows a model to infer the correct label simply using static inputs, e.g. inferring the label 'soccer' from a green field. In contrast, the something-something dataset contains labels, such as 'putting something on a table', 'pretending to put something on a table' or 'putting something on a slanted surface so it slides down', which are substantially more difficult to infer.

4.3.2 *PredNet+ model*

In their project work, Prakash Rane, Edit Szügyi and Vageesh Saxena implemented a modification to the PredNet architecture called PredNet+. The main idea is to extend the baseline model using an additional label classification module that is connected to the hierarchically highest representation layer. Figure 16 schematically shows the signal flow through this additional unit.

This additional module consists of two main parts: An encoder part transforms the outputs of the hierarchically highest representation units into probabilities over class labels using two ConvLSTM layers. These label classes are then projected back to the spatial domain and fed back to the top-down pathway. Throughout their experiments, Prakash Rane, Edit Szügyi and Vageesh Saxena apply this label classification for each frame. For each frame, the weighted sum is passed through a softmax function to get the final class probabilities. To deal with the beginning of the video, where no context for meaningful predictions is available, they use weighing-over-time using an exponential function. PredNet+ is designed such that the latent features at the top-most representation layer are shared between two tasks: Label classification and future frame prediction. The future frame predictions are conditioned on the label predictions made by the label

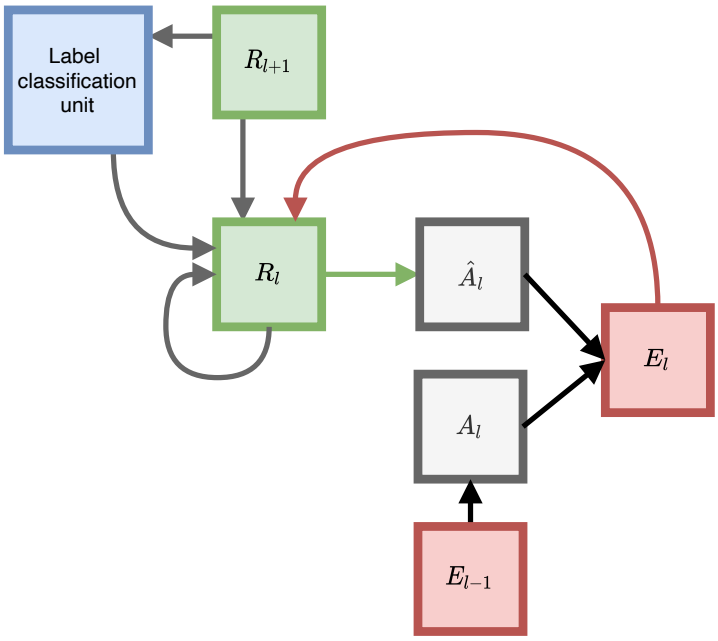


Figure 16: The proposed PredNet+ architecture with an additional classification module. Shown are the representation units (green) of the hierarchically deepest layer and the hierarchical layer below.

classification unit. The highest hierarchical layer is chosen, since it has has the largest spatio-temporal receptive field. We hypothesize that such a setup improves the results on both sub-tasks, as several previous multi-task studies indicate [24, 64].

4.3.3 *Evaluation metrics*

Generally speaking, there is no "best choice" for an evaluation metric for images [29, 132]. Thus, throughout their experiments, Prakash Rane, Edit Szügyi and Vageesh Saxena use three different evaluation metrics in order to judge the quality of model's predictions. For video prediction tasks, they use two commonly used metrics: Peak Signal Noise Ratio (PSNR) [132] and the Structural Similarity Index Measure (SSIM) [219]. Like Mathieu, Couprie, and LeCun [132], they calculate PSNR and SSIM only for the frames which have movement with respect to the previous frame and call them "PSNR movement" and "SSIM movement" respectively. This choice is important, since action videos are often short in duration, i.e. cover only few frames. This increases the risk of rewarding the model for simply predicting a still image. Additionally, they use a third metric called "conditioned SSIM":

$$\mathrm{SSIM}_{\mathrm{cond}} = (\mathrm{SSIM}_{\mathrm{max}} - \mathrm{SSIM}(a_{t-1}, \mathrm{pred}_t)) * \mathrm{SSIM}(a_t, \mathrm{pred}_t) \tag{39}$$

where a_t denotes the actual observation at timestep t. This conditioned SSIM metric measures how different the predictions are from the previous frame. This can be interpreted as measuring the "risk" in the model's prediction in comparison to simply copying the last observed frame.

4.4 EXPERIMENT

In their analysis, Prakash Rane, Edit Szügyi and Vageesh Saxena focus on two different aspects: The first experiment evaluates the performance of the baseline PredNet architecture on the something-something dataset and includes a visualization of the network states in the context of unsupervised prediction. The second and third experiment focus on the additional classification module, where supervised label classification is done simultaneously with video prediction.

4.4.1 *Unsupervised prediction on the something-something dataset*

In the first experiment, the baseline PredNet architecture is trained on the something-something dataset using 10 different hyperparameters settings with a different number of layers, channels per layer, input image size and frames-per-second (FPS) settings. Furthermore, two different loss variants are tested: L_0 and L_{all}. Models trained with L_0 use only the lowest hierarchical layer to compute the parameter gradients, while the L_{all} uses all layer's errors. Table 4 lists the evaluated settings. After running the models, Prakash Rane, Edit Szügyi and Vageesh Saxena visualize the encoded states in each layer of the model using the average activation of all channels in a layer. This approach is inspired by related work and can deliver insightful information, despite a reduction of complexity [71]. They additionally plot the mean of the error signals E_i and representations R_i in every layer throughout the duration of a video. A visualization example is shown in Figure 17.

Prakash Rane, Edit Szügyi and Vageesh Saxena also evaluated the possibility to generate long-term predictions with the PredNet model by feeding back the prediction at time t as the input at the next time step. Multi-step predictions can then be generated by repeating the procedure n times until the end of the video is reached. Using this approach they compared a model trained by using $(t+n)$ prediction to the baseline $(t+1)$ model.

4.4.2 *Classification with PredNet+*

The second and third experiment use the proposed PredNet+ model and evaluate its performance on multi-task learning, where action la-

Model	Frame rate (FPS)	Layers	Image size (pixel)	Number of param.	Loss
0	**3**	4	48 X 56	6.9	L_0
1	**6**	4	48 X 56	6.9	L_0
2	**12**	4	48 X 56	6.9	L_0
3	12	**4**	32 X 48	6.9	L_0
4	12	**5**	48 X 80	5.3	L_0
5	12	**6**	64 X 96	5.8	L_0
6	12	**7**	128 X 192	6.2	L_0
7	12	6	**96 X 160**	**7.2**	L_0
8	12	5	48 X 80	5.3	L_{all}
9	12	6	64 X 96	5.8	L_{all}

Table 4: Evaluated model configurations. Similar models are grouped with horizontal lines and the column that varies is marked in bold.

bel classification is done simultaneously with video prediction. The model is trained on the something-something dataset and compared to state-of-the art results in terms of classification accuracy. In this context, Prakash Rane, Edit Szügyi and Vageesh Saxena evaluate only the best models with 4,5 and 6 layers based on the best results in the first experiment. In order to evaluate various aspects of the architecture, they test three additional variations:

- Removal of the recurrent memory in the label classification unit by replacing the ConvLSTM with convolution layers
- Extension of the label classification loss function such that the model is rewarded for predicting at least the correct verb in the label
- Modification of the loss weightings to control the relative importance of the classification and prediction task

4.4.3 *PredNet+ on synthetic data*

The third and final experiment evaluates the influence of the additional top-down conditioning in the proposed PredNet+ model in a simplified and controllable, synthetic dataset. In this context, Prakash Rane, Edit Szügyi and Vageesh Saxena use a modified version of the moving MNIST dataset designed specifically for this purpose [198].

This dataset features a static background consisting of randomly generated overlapping geometric shapes, and a single hand-written digit moving in one of eight directions. Labels for each frame indicate the direction of the digits movement. An example from this dataset is shown in Figure 24. In order to evaluate the influence of the additional top-down information on the prediction accuracy, they keep track of spatio-temporal prediction performance, while using the movement label classification as an auxiliary task.

4.5 RESULTS

The following sections summarize the quantitative and qualitative results obtained by Prakash Rane, Edit Szügyi and Vageesh Saxena in the previously described experiments with PredNet and the proposed PredNet+ variant.

4.5.1 *Unsupervised prediction on the something-something dataset*

Upon inspection of the generated visualizations and the quantitative results, several observations about PredNet can be made:

- Relevant features are learned better on videos with continuous motion
- The model is sensitive to the temporal sampling frequency
- Model errors do not behave as hierarchical predictive coding would predict
- The network design favors short-term over long-term predictions

Generally speaking, the model appears to resort to previous-frame-copy if there are no cues for motion in the previous frames. If there is a cue for motion and if the direction of the motion is continuous and the motion is smooth, it interpolates the object in the direction of the motion. Otherwise, it blurs the region containing the object of motion to reduce the L2 loss. Figure 17 shows a typical example for the visualization of model states aligned to each predicted frame during next frame prediction with the baseline PredNet model. When the sampling frequency, or FPS setting, is reduced, the results are more pronounced. An example for such low FPS setting is shown in Figure 18. In the context of the something-something dataset, this effect is clearly noticeable, since it contains large amounts of still frames. It should be noted, that this problem is less severe on datasets with continuous motion, such as the KITTI or moving faces datasets, which have been previously used to evaluate PredNet [61, 125].

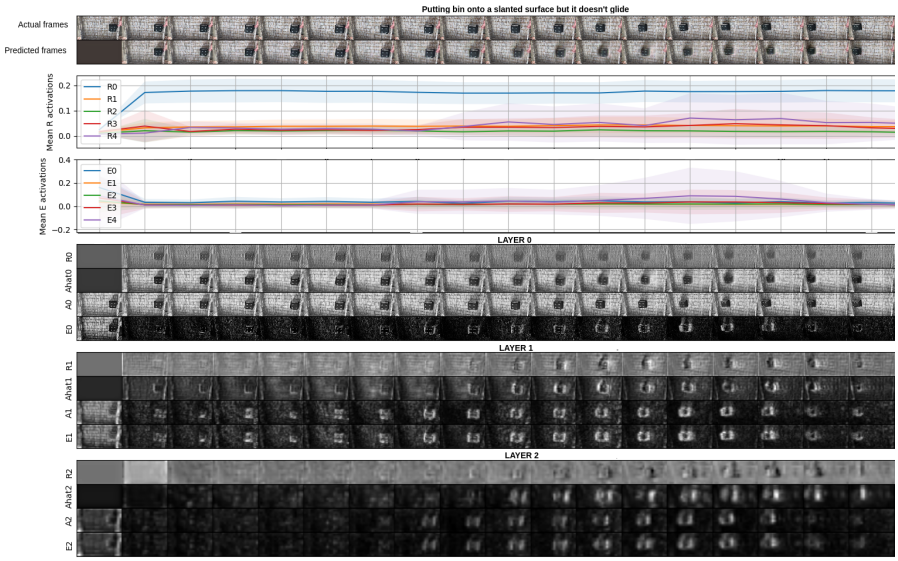


Figure 17: Visualization of model states during next frame prediction. Each column corresponds to a single time step, while rows resemble the computed states in each layer.

Figure 18: Example of a low FPS video and the predictions made by PredNet.

A comparison of model performance after training on videos with FPS rates 3, 6 and 12 is displayed in Figure 19. Shown is the improvement over a model that simply performs last-frame copying for each metric. The resulting scores vary significantly between different temporal resolutions and indicate that high resolution is necessary to see significant improvements over the last-frame copying baseline. This implies that the temporal resolution is a crucial hyper-parameter for the model.

As explained in Section 4.1 PredNet's layer-wise prediction of *prediction errors* is in contrast to predicting the *states* of the respective layer, as it is the case in hierarchical predictive coding models. This makes it difficult to interpret the underlying generative model and makes an empirical analysis necessary.

As visible in Figure 17, the average bottom-up error tends to increase towards higher layers. Naively, one would expect the opposite to happen, i.e. higher layers with more context to have smaller errors. Interestingly, the original PredNet model generally performs better with L_0 loss, i.e. using only the error of the lowest hierarchical layer to compute the gradients, while models trained with L_{all} perform

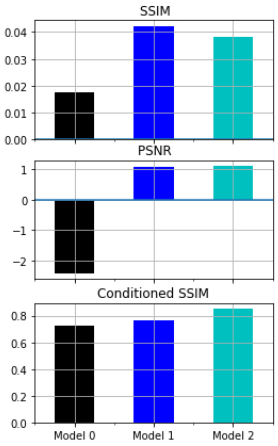


Figure 19: Comparison of model performance with respect to the employed frames-per-second rate (FPS). Shown is the model's improvement on a last-frame-copy baseline.

worse [125]. As displayed in Figure 20, similar results are observable for the something-something dataset used here. From the perspective of hierarchical predictive coding, this is surprising, since one would expect models to benefit from a minimization of all layer's losses.

Another insight in the workings of PredNet is that the states in the lowest layer differ significantly from those in the higher layers. Figure 17 shows an example for this difference. Similarly, the average activation of the representation units is higher and follows a different trajectory in comparison to the remaining layers. This indicates that the first layer does the "actual" prediction, while the remaining layers regress the prediction errors.

As expected, the models trained with multi-step prediction slightly outperformed the single step baseline. Three extrapolations of the best performing model (7) are shown in Figure 21. Most strikingly, PredNet resorts to last-frame-copying after two time steps and creates increasingly blurry predictions throughout the extrapolation. This means that the generated predictions do not suffice to continue the motion when being fed back to the model. Figure 22 shows model performance in comparison to the last-frame-copy baseline dependent on the starting position (in frames) of the extrapolation. The model generally performs better when the extrapolation is started later in the video and more temporal context is available.

These results indicate that PredNet is designed to excel at short-term interpolation tasks in the context of videos with smooth motion and high sampling rates. Generating long-term predictions, in contrast, is difficult to achieve.

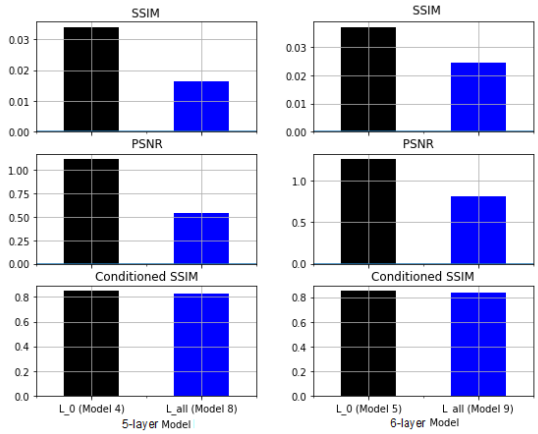


Figure 20: Influence of L_0 and L_{all} loss on model performance. Scores show the model's improvement on a last-frame-copy baseline.

Figure 21: Extrapolating the best performing model. The red mark indicates the extrapolation start.

4.5.2 *Classification with PredNet+*

Table 5 shows the best achieved classification accuracy on the something-something dataset. For comparison, state-of-the-art results by Mahdisoltani et al. [126] and the baseline model described by Goyal et al. alongside the something-something dataset [66] are given. The results clearly show that PredNet+ is far from the state-of-the-art. Surprisingly, the classification results did not change at all ($\pm 0.6\%$) for any of the tested model variations. This indicates that the features from the top-most representation units do not carry information that is central to the model's performance.

Model	Top-1
Baseline [66]	11.5
Ours	28.2
Mahdisoltani et al. [126]	**51.38**

Table 5: Classification accuracy on the something-something dataset.

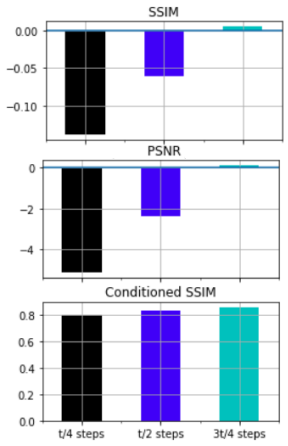


Figure 22: Performance with different starting points for extrapolation. t denotes the total number of frames in the video. The scores indicate the improvement on a last-frame-copy baseline.

Interestingly, as shown in Figure 23, the accuracy of future frame prediction in PredNet+ actually degrades in comparison to the unmodified PredNet models. This effect gets even more severe, when the classification task is given more importance by adjusting the relative weighting of classification over video prediction. These results indicate, again, that the information learned in the highest hierarchical layer, although being meaningful for the classification task, does not improve the video prediction.

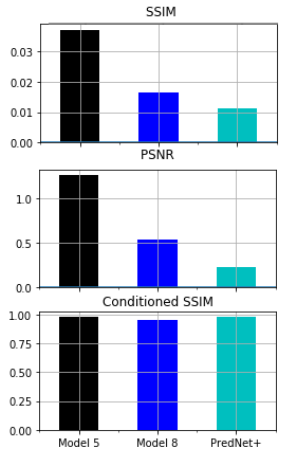
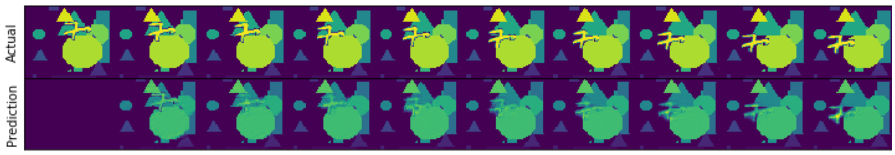


Figure 23: Comparison of the best performing PredNet+ model with unmodified PredNet models. The scores show the improvement over a last-frame-copy baseline.

4.5.3 *PredNet+ on synthetic data*

Both PredNet and Prednet+ generated meaningful predictions, with comparable accuracy on the modified moving MNIST dataset. As visible in Figure 24, the predictions generally were more blurry in moving parts of the frame, in contrast to the static background. The mean absolute errors from the last-frame-copy baseline model, PredNet and PredNet+ are shown in Table 6.

(a) PredNet



(b) PredNet+

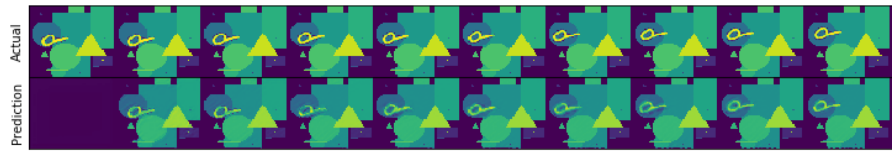


Figure 24: Model predictions on the modified moving MNIST dataset.

Including the additional classification task during video prediction slightly improves the MAE score. Upon qualitative inspection, the predictions of non-stationary parts looked sharper in PredNet+. This indicates the potential of including semantic top-down information to improve model performance, at least when the additional information can directly be related to the observed frames.

Model	MAE score
Previous-frame-copy	8e-050
PredNet	7.6e-05
PredNet+	7.3e-05

Table 6: Comparison of Prednet and PredNet+ on the modified moving MNIST dataset.

4.6 DISCUSSION

In this chapter, we have reviewed PredNet, a popular predictive coding inspired DNN model on a challenging dataset focusing on action

classification. We found that PredNet's architecture is different from hierarchical predictive coding models. This is primarily because PredNet predicts the error signal of lower layers instead of their states. We found that PredNet appears to be tailored specifically towards short-term predictions and often resorts to copying the last frame, especially when the temporal sampling frequency is too low or too high. When the sampling frequency is low, the abrupt motion is difficult to predict and leads to a simple last-frame copying strategy. Interestingly, upon qualitative inspection, the model also resorts to last-frame copying, when the sampling frequency is very high, i.e. minimal difference is between subsequent frames. We found that, in many cases, PredNet outputs blurry predictions, possibly due to the lack of probabilistic temporal predictions, which could cover multiple different outcomes given the temporal past. As we used a challenging dataset, there are many instances with multiple possible future frames. We further tested the possibility to use PredNet in the context of action label classification. To do so, we proposed a modification to the PredNet architecture, that predicts action labels using the hierarchically highest representation units. In the proposed model, the classification module provides additional semantic top-down information to to lower hierarchical layers. We found that PredNet+ performs far from the state-of-the-art architectures and that, surprisingly, adding top-down information worsens the video prediction performance. When evaluated on a simple, synthetic dataset, the same modified architecture outperforms PredNet by a slight margin. These results suggest that PredNet is not a viable candidate for hierarchical predictive coding in the context of deep neural networks. Although not specifically mentioned by the authors, PredNet could also interpreted as a strictly *dynamical* model. In this context, predicting "errors of errors" makes more sense, e.g. when modelling multiple orders of predicted motion. Such investigations into the dynamical aspects of PredNet specifically, however, are left to future work.

5

VARIATIONAL PREDICTIVE CODING FOR AUDIO AND EEG

A major challenge in modelling free energy optimisation in deep neural networks is the inclusion of inference over temporal sequences. In the previous section, the aspect of observations arriving sequentially in time has been simplified, by simply treating sequences of fixed length as static observations. This however, is not in line with process models under the Free Energy Principle, which usually formulate temporal inference as a process of Bayesian filtering. In Bayesian filtering schemes, such as dynamical predictive coding, inference is made *while* new observations are encountered, with respect to information extracted from past observations and hierarchical ("top-down") priors [49, 51]. This chapter focuses on designing a deep recurrent architecture that models temporal information sequentially, one discrete observation at a time. In contrast to existing deep RNN models that are inspired by predictive coding, such as PredNet [124], the model explicitly models a hierarchical Bayesian latent variable model with Gaussian latent states. In line with PredNet [124], amortized inference from data to latent states makes use of deep neural networks that explicitly process prediction errors. Similarly, hidden hierarchical layers are updated with respect to explicitly propagated prediction errors between their predictions and the latent state of the respective lower layer.

We evaluate the model in the context of unsupervised representation learning on audio data, by comparing the prediction error response of the network with evoked responses in the human brain. In particular, we investigate, whether the model prediction error response can be used to derive predictions about *temporal* locations of human responses. In contrast to the previous section, this involves processing only sensory information with the model, followed by an evaluation of the human response at the temporal locations derived from the model response. After evaluating temporal processing of sensory data, we also investigate whether the model can be used directly on brain signal recorded in EEG. In particular, we design a variant of the model that processes EEG data temporally aligned to fixation related potentials (FRPs) during a free reading task and evaluate the possible to actively infer the temporal onsets of FRPs guided by the expected EEG signal at these locations.

The content in this section is based on the following publications:

> [pub:5] A. Ofner and S. Stober. "Modeling perception with hierarchical prediction: Auditory segmentation with deep predictive coding locates candidate evoked potentials in EEG." In: *21st International Society for Music Information Retrieval Conference (ISMIR)*. 2020, pp. 566–573.
>
> [pub:4] A. Ofner and S. Stober. "Balancing Active Inference and Active Learning with Deep Variational Predictive Coding for EEG." In: *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE. 2020, pp. 3839–3844.

STRUCTURE OF THE CHAPTER

The rest of this chapter is organized as follows: Section 5.1 relates dynamical PC to evoked responses in the human brain and motivates the retrieval of brain-related information from a free energy minimizing model applied to audio. Section 5.2 reviews related models and introduces the proposed deep variational predictive coding network. Section 5.3 explains how the model can be applied to derive candidate ERP locations from audio data, followed by a presentation of the results in Section 5.4. Sections 5.5 and 5.6 cover the application of a variant of the model to EEG signal prediction. Finally, Section 5.7 concludes the chapter with a discussion of the results.

5.1 PREDICTIVE CODING AND HUMAN AUDITORY PROCESSING

Studying the brain's response to auditory stimuli is still limited by the lack of resources that map complex musical stimuli to neural processes. Studies in cognitive neuroscience and brain computer interfacing on auditory evoked brain states require labor intensive manual preparation and often focus on isolating particular brain responses using sparse stimuli presented individually [149, 161]. While datasets on brain states evoked by natural music exist, they often lack fine-grained annotations of the event structure and corresponding neural activity [108, 123, 203]. This entails a demand for efficient and unsupervised mapping techniques between natural music and evoked brain states. Furthermore, there is a need for biologically plausible and multi-modal models for such mapping, as induced brain states are a mixture of stimulus-derived and subjective, cognitive or contextual factors.

As a comprehensive explanation for human perception, PC offers a detailed description of how humans parse and predict sounds and map auditory stimuli to musically meaningful and hierarchically organized units [214]. In predictive coding, the neural response to music

is shaped by hierarchically organized expectations [172]. This hierarchy of expectations connects predictions about low-level auditory features to more global context, such as the listener's musical expertise or levels of entrainment during listening [214]. The underlying dependencies between expectancy and uncertainty in PC are particularly interesting in the context of music perception, as it can be described as continuously resolving uncertainty and forming new expectations [32, 74, 107]. This is in line with evidence on the predictive nature of human music perception, especially within studies on unexpected stimulus deviations and the influence of the listener's expectancy on attention and perceptual precision [32, 107]. Under predictive coding, long-term expectations from temporally stable aspects of music, such as genre or tempo form top-down predictions about the activity of layers closer to the actual auditory information [214].

Studying the human perception of music has also received increased interest in other fields of research, such as Music Information Retrieval (MIR). As humans solve tasks such as beat tracking, genre identification or musical prediction with ease, many MIR methods rely on computational models inspired by human perception. Within the field of MIR, the capacity of PC algorithms to compress and represent auditory information on the sensory level has been exploited for various tasks such as speech re-synthesis or audio compression since many years [7, 186]. The human brain, however, augments low-level sensory representations with a hierarchy of more abstract, semantic predictions from other brain areas [172]. As mentioned previously, this aspect of hierarchical predictive learning has found traction in the domain of DNNs, but so far has been applied mostly to images and video processing [70, 124]. Furthermore, most popular implementations of deep PC often only rely on non-linear transformation of the sensory error and not yet abstract away from pure sensory prediction. Autoregressive modeling of audio has seen tremendous progress in recent years, with a plethora of models performing tasks such as sample level audio prediction or speech synthesis, often with impressive results [21, 154, 155]. However, such autoregressive models are computationally expensive and sample-level prediction models still tend to struggle with incorporating more abstract and long-term musical features.

5.1.1 *Auditory evoked potentials and musical structure*

Recent years have shown a variety of approaches to studying the human brain's response to auditory stimuli, especially with functional magnetic resonance imaging and electroencephalography. EEG is especially adequate in the context of music due to its higher temporal resolution. A multitude of auditory features, such as loudness, frequency, tempo and rhythm have been traced in EEG recordings of

brain activity during music perception [22, 152, 153, 207]. Next to these stimulus-derived aspects, recorded brain activity has further been analysed with respect to more contextual aspects of music perception, such as the listener's attention, which is modulated by aspects such as expertise or engagement [6]. Two extensively researched aspects of the neural response underlying perception potentials are ERPs and steady-state evoked potentials (SSEP) [145, 167]. ERPs and SSEPs differ mainly in their temporal scope: While ERPs are aligned to a single loation (typically the onset of a particular event), SSEPs show frequency alignment to stimulus periodicity over longer time frames [165]. For ERPs, the brain response aligned to the event type of interest is analysed after averaging large amounts of trials [94]. Auditory event-related potentials are modulated by aspects such as rhythm, pitch, timbre or the duration of musical events, all of which play an important role in human audio segmentation [136, 147, 166, 167, 185, 190]. Many of these evoked potentials have been explained in the context of PC as a mixture of bottom-up and top-down mechanisms that are modulated both contextual expectations and the auditory stimulus itself [10, 227]. Similar to ERPs, SSEPs are inspected after averaging over many trials, but do not require temporal alignment to event onsets. Instead, SSEPs characterize periodic mappings between auditory features and brain response, such as phase locking to perceived frequencies or loudness envelopes. SSEPs are particularly interesting in the context of natural music, as they allow to inspect more coarse-grained aspects of music perception, such as temporal or rhythmic entrainment, the "groove". Both ERPs and SSEPs can be related to predictive processes aiming at structuring the incoming sensory signal into meaningful events in a hierarchical fashion [151, 227].

These insights motivate us to connect deep PC with variational inference as a model of canonical computation for unsupervised stimulus representation in order to segment natural music into units that are musically meaningful. Following the assumption that hierarchical PC of music explains a substantial amount of evoked brain states, we analyse the retrieved musical structure in terms of the induced neural activity in electroencephalographic signal (EEG).

5.2 DEEP VARIATIONAL PREDICTIVE CODING

From a high level perspective, modelling a deep PC network under the constrains of the FEP leads to several constrains on the model structure. One constraint addresses the underlying hierarchical structure of the probabilistic generative model [49, 51]. Each hierarchical layer abstracts further away from the sensory, summarizing spatial and temporal details of the activity in lower layers. Next to a hierarchical separation into (Gaussian) latent states, this requires each hierarchical layer to compute *temporal* predictions about expected states

in the next lower layer. As we're interested in applying the model to complex sensory inputs, deep neural networks are a straightforward choice for the parameterization model parameters. In this context we resort to the backpropagation of error algorithm, i.e. do not focus on biologically plausible credit assignment between hierarchical layers or timesteps. We treat more strict interpretations of local inference and learning in chapter 7 of this thesis. The following sections describes the proposed model architecture that combines several aspects in the context of PC in deep neural networks:

1. Explicit propagation of error signals for amortized inference
2. A hierarchy of stochastic latent states where posterior states depend on top-down and bottom-up prediction errors
3. Multi-step latent dynamics with a stochastic and a deterministic component in each hierarchical layer
4. Separation of time-scales through top-down prediction of latent state sequences

In isolation, these modelling aspects are not new and have found extensive use in their respective area of research. For example, using explicit error signals to drive amortized inference is a core component of PredNet [124]. Hierarchical Bayesian modelling over different timescales has a long tradition in the context of Bayesian filters in neuroscience [49, 51]. The combination of stochastic and deterministic components has been employed previously to design efficient deep recurrent models, such as Variational Recurrent Neural Networks (VRNNs) [20]. Similarly, the idea of optimizing variational free energy with respect to entire sequences of latent states has found use in the context of deep reinforcement learning models [69]. The next section provides an overview of related work on deep PC networks for temporal prediction and variational inference in recurrent state-space models, followed by a detailed description of the suggested variational PC model.

5.2.1 *State-space models and deep predictive coding*

The design of efficient models for temporal prediction over sequences has a long tradition. Within the class of deep neural networks, a conventional, yet effective way to model long-term dependencies are recurrent neural networks (RNNs) that model transitions of deterministic latent states h_t at discrete time-steps t [177]. This is in contrast to probabilistic state-space models (SSMs), which model distributions over states and their stochastic transitions [33, 69]. In deep learning, the parameterization of such probabilistic state-space models usually relies on amortized variational inference of the probabilistic state s

from the corresponding observation o, e.g. using a variant of the variational autoencoder [104]. More recently, recurrent state space models (RSSM) have been proposed that combine the possibility to learn long-range dependencies using deterministic recurrent connections as in RNNs jointly with the expressiveness of stochastic states of state space models. An influential example of RSSMs are VRNNs [20]. VRNNs are trained in a relatively straightforward way using variational inference, by feeding the observation at every timestep to the model and optimizing accuracy and complexity of the model using a evidence lower bound for each observation [20]. VRNNs have inspired a variety of more elaborate RSSM models, e.g. in the domains of video prediction, time-series forecasting or reinforcement learning [17, 69, 96]. Figure 25 shows an overview of the underlying generative model for RNNs, SSMs and RSSMs with respect to two discrete timesteps. Generally speaking, RSSM based approaches show improved performance over simpler models, while still allowing the interpret the underlying state-space model in terms of a Bayesian filter [69]. Additionally, the RSSM structure allows to separate the overall generative model explicitly into a dynamical part, i.e. the hidden state dynamics and associated memorization, and static part, for representing inferred states. Since we're interested in hierarchical prediction, this allows to focus on predicting the latent states themselves, in contrast to predicting potentially more complex representations that represent aspects of memory and dynamics. Conceptually, this separation can also be motivated using PC models in neuroscience, which often explicitly separate between cause and hidden states, representing dynamics and inferred states respectively. [49, 51]. These reasons motivate use to resort to a RSSM based generative model.

Within the area of deep reinforcement learning, recent work has shown the possibility to generate predictions entirely in latent space, instead of iteratively encoding observations on step at a time. [69]. Making predictions over multiple timesteps allows to express predictions over entire sequences of (hidden factors underlying) data. This idea has also seen use in PC inspired DNN models, such as CPC [156]. Contrastive predictive coding, as the name suggests, is rooted in noise-contrastive estimation. The core idea behind noise-contrastive estimation refers to the chosen loss function, that maximizes the mutual information between compressed representations of a future target x and the current context c via their mutual information [156]:

$$I(x; c) = \sum_{x,c} p(x, c) \log \frac{p(x \mid c)}{p(x)} \tag{40}$$

Since predictions are made with respect to many time-steps, this approach allows to learn slowly moving features that maximize the mutual information over time [156]. Representation learning using a probabilistic noise contrastive loss is a powerful concept. That said,

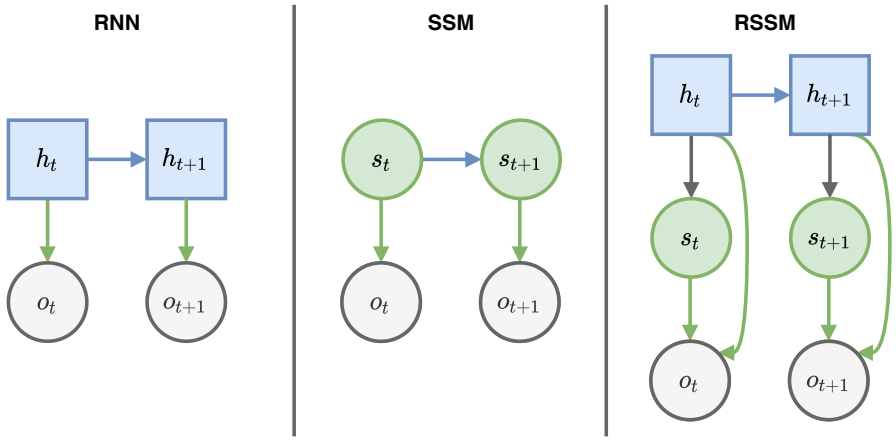


Figure 25: Recurrent dynamics in recurrent neural networks (left), state space models (middle) and recurrent state space models (right). Recurrent neural networks model deterministic transitions (between blue nodes) between inferred hidden states. In contrast, transitions in state space models are stochastic (between green nodes). Recurrent state space models, such as VRNNs [20] use both deterministic and stochastic components, allowing to express probability distributions while maintaining the benefits of deterministic memory.

the generative model in CPC requires negative samples to be chosen explicitly, since these are needed to compute the contrastive loss. This specific choice of the underlying loss function also prevents an interpretation of CPC as a general interpretation of PC that optimises variational free energy, such as we intend to model. The CPC architecture essentially describes an autoregressive model that predicts latent states within a single hierarchical layer. This means that empirical priors, i.e. a top-down signal, from more abstract representations of the data are not present. Noise contrastive prediction inspired by PC has also been applied to video prediction [72]. Next to CPC, a deep PC network for unsupervised representation learning from videos have been suggested within the deep learning area [124]. Differently to CPC, PredNet models a hierarchy of representations and is entirely deterministic. We have reviewed the PredNet architecture in more detail in the previous Chapter 4. To briefly recapitulate: PredNet is inspired by PC models in neuroscience and explicitly propagates prediction errors internally, upwards in the hierarchy. PredNet hierarchically predicts the prediction error of the lower layer, instead of predicting the states of the respective layer (as would be required for hierarchical predictive coding), making it difficult to interpret the generative model. Figure 15 shows a schematic comparison between the PredNet architecture and the connectivity required for hierarchical predictive coding. Interestingly, PredNet has been shown to produce a variety of cognitive phenomena that are often associated with PC theory, such as illusory motion [220]. As a simple, yet effective

deep predictive architecture, PredNet has produced several variants, often times focusing on reducing the complexity of the underlying recurrent model or the integration of actions [44, 233]. Other lines of research have addressed the lack of a feature hierarchy in deterministic dynamical models that, such as PredRNN++ or the Hierarchical Prediction Network (HPNet) [168, 217]. The central idea in these models is a hierarchical organization of powerful RNN models, typically variants of LSTM [80]. While these deterministic models lack a thorough connection to Bayesian interpretations of brain function (or to hierarchical PC more specifically), such as discussed here, they have been shown to reproduce several to neurophysiological phenomena [168]. Another core conceptual aspect of PredNet is the explicit propagation of prediction error signals, in terms of a negative and positive error signal, with encoder networks that drive inference of the model's representations [124]. In the model presented in this chapter, we resort to such explicit error propagation and hierarchical stacking of recurrent neural networks, however in the context of a hierarchical probabilistic latent variable model and variational inference. The striking difference between connectivity in PredNet and hierarchical PC has been noticed and a model with "corrected" hierarchical connectivity have since been implemented [204]. The resulting model is still entirely deterministic, i.e. does not allow a straightforward probabilistic interpretation of the inferred states. Nevertheless, the overall structure more closely resembles PC models in neuroscience and the performance for next-frame prediction outperforms PredNet [84, 172, 204]. Other dynamical DNN models, that are more loosely inspired by predictive coding, have found use in video anomaly prediction or video representation learning [72, 231].

The notion of PC in neuroscience has also influenced a variety of deep neural networks that apply aspects of hierarchical PC to unsupervised representation learning of static inputs, such as images [35, 75, 222]. These models typically lack explicit temporal processing, but often include aspects of iterative inference on static inputs, e.g. via recurrent processing of static inputs [70].

Next to deep PC networks, the notion of PC has found attention in the domain of (neuro-)robotics [3, 19, 87, 88, 93, 206]. In this area of research a variety of variational RNN models has been proposed that learn by reducing prediction errors implicitly using backpropagation of error through time [4, 19]. This is in contrast to models that propagate prediction errors explicitly, such as PredNet or the model investigated in this chapter [124]. In robotics, a particular focus has been devoted to comparing the mutual dependence between deterministic and stochastic processing [3] and employing PC in the context of action imitation and social interaction between humans and robots [88, 206]. The underlying RNN models often capture multiple (continuous) time-scales and have been applied to "online" recognition, i.e.

iteratively inferring observed states using backpropagation of error [4, 19]. Finally, PC inspired RNN using implicit error backpropagation in robotics models have been applied to goal-directed behavior and planning [93].

5.2.2 *Hierarchical predictive coding model*

In order to enable hierarchical predictions across multiple time-scales, we stack multiple recurrent layers and train the network to predict a sequence of probabilistic states s that are inferred in each respective lower layer. In line with Bayesian views on brain function, we express the prior distribution of states s_l in each layer l as a Gaussian parameterized by mean μ_l and variance σ_l parameters [51, 69]. While the lowest hierarchical layer predicts sensory observations o, the network's hidden hierarchical layers predict latent states s_{l-1} of the respective lower layer. More specifically, we sample the prior distribution $p(s)$ of each layer and transform the resulting activation with a decoder network. The decoder network of the lowest layer parameterizes the prediction $\hat{o}_t$ of the expected observation. The decoders in hidden layers output predictions about the mean $\hat{\mu}_t$ and variance $\hat{\sigma}_t$ parameters of the next lower layer.

The model is trained by optimising the variational free energy, or evidence lower bound with respect to a complexity term, that depends on previous observations and states within each hierarchical layer. The lowest hierarchical layer optimises the complexity with respect to predictions about observations o at discrete timesteps t:

$$\text{Complexity} = \mathbb{E}_{q(s_{t-1}|o_{\leqslant t-1})}[\mathrm{KL}[q(s_t|o_{\leqslant t})\|p(s_t|s_{t-1})]] \tag{41}$$

Simultaneously, the accuracy of predicted observations maximized:

$$\text{Accuracy} = \mathbb{E}_{q(s_t|o_{\leqslant t})}[\ln p(o_t|s_t)] \tag{42}$$

In analogy, the complexity for a hidden hierarchical layer s_l is computed with respect to observed states s_{l-1} of the next lower hierarchical layer.

The resulting evidence lower bound for a sequence of observations with length T is:

$$\begin{aligned}\mathrm{ELBO}(q) = \sum_{t=1}^{T} \Big(& o_{q(s_t|o_{\leqslant t})}[\ln p(o_t|s_t)] \\ & - o_{q(s_{t-1}|o_{\leqslant t-1})}[\mathrm{KL}[q(s_t|o_{\leqslant t})\|p(s_t|s_{t-1})]] \Big)\end{aligned} \tag{43}$$

5.2.3 *Propagating explicit prediction errors*

In contrast to the related classes of variational autoencoders and variational RNNs, we do not employ an encoder network that directly transfers observations to a posterior distribution [20, 104]. Instead, the encoder network processes only the error e_t between predicted $\hat{o}_t$ and observed values $\hat{o}_t$ at discrete timesteps t:

$$e_t = o_t - \hat{o}_t \tag{44}$$

Similarly, the encoders between hierarchical layer process the error signal between predicted mean $\hat{\mu}_l$ and variance $\hat{\sigma}_l$ and the observed mean μ_l and variance σ_l of the next lower hierarchical layer:

$$\begin{aligned} e_{\mu_l} &= \mu_l - \hat{\mu}_l \\ e_\sigma &= \sigma_l - \hat{\sigma}_l \end{aligned} \tag{45}$$

Each layer of the model thus computes approximate state posteriors $q(s_{1:T}|e_{1:T}) = \prod_{t=1}^{T} q(s_t|e_{t-1})$ by filtering previously observed prediction errors e_t at time steps t with an encoder network. In this basic model configuration, the model is regularized in each hierarchical layer with respect to the divergence from the prior in 44. By selecting a pair of adjacent layers and minimizing the accuracy and complexity term between them, this structure allows to form predictions that are consistent between layers, i.e. show small or no top-down prediction error.

The state priors $p(s_t|s_{t-1})$ in all layers are modelled as a Gaussian and are parameterized with a feed-forward neural network with respect to mean and variance. After generating the state prior, predictions about lower activities can be formed. The predictions in the input layer are made with a deconvolutional neural network to parameterize the mean of expected inputs. The prediction models in hidden layers parameterizes variance and mean of the same size as the respective lower state posteriors. Without any deterministic or recurrent aspects to the model, the state prior $p(s_t|s_{t-1})$ would not be particularly expressive. Therefore, the model has access to deterministic recurrent memory states that we refer to as belief states b_t. They can be seen as the explicit belief the model has at a particular point in time [17]. Belief updating is done with a LSTM network conditioned on previous beliefs b_{t-1}, the top-down and bottom-up prediction errors e_{td}, e_{bu} and the previous state s_{t-1}. Importantly, any information regarding the previous error for outgoing predictions passes a stochastic node before integration into the belief. Figure 26 shows an overview of the transitions in a single layer and the connection to the top-down predictive pathway.

While the overall network structure was identical for the experiments on audio and EEG processing, they differed in some aspects,

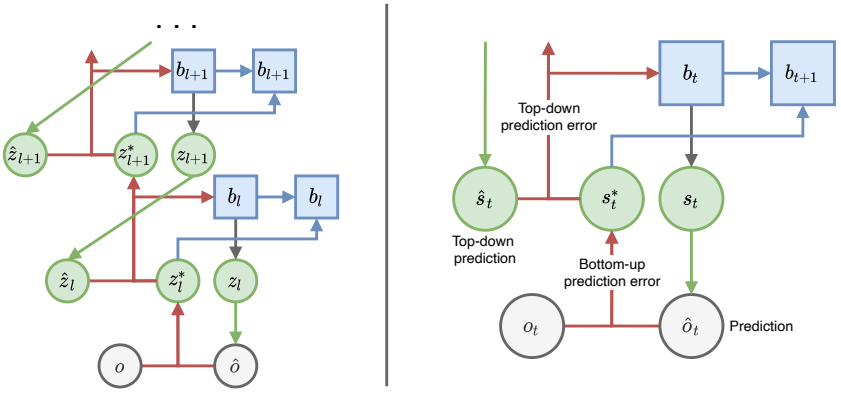


Figure 26: *Left*: Hierarchically organized PC network with two layers. Each hierarchical layer predicts posterior predictions of the respective lower layer. *Right*: A single hierarchical layer in detail. Green arrows indicate descending predictions of lower layer states, or the sensory data. Red errors indicate ascending prediction errors that inform the posterior z^* of probabilistic latent states z in each hierarchical layer. Each hierarchical layer aggregates deterministic information over time using recurrent connections between hidden states h.

such as the type and the size of the chosen LSTM model. The next sections cover these implementational details.

5.2.4 *Audio model*

The audio experiments used a model with three hierarchical layers. We model $q(s_t|e_{t-1})$ as diagonal Gaussian for all layers with mean and variance parameterized by a convolutional neural network with two layers of 64 and 128 units each. The convolutional layers were followed by a dense network of 1024, 512 and 256 units respectively. The network uses a convolutional decoder networks and parameterizes expected mel spectrogram inputs by treating time and frequency axis as spatial domains. We used ReLU activations for all CNNs and hyperbolic tangent activations for the decoder's output layer [148]. Transitions were generated using a LSTM network [80]. In each layer, the prediction error was computed with respect to positive and negative prediction error. The error in each layer was then ReLU activated before propagation to the encoder networks. For all presented experiments, we trained the model to convergence of the input layer reconstruction loss. For this, we used the Adam optimizer with a learning rate of 10^{-3} [103]. The KL divergence terms for each layer were scaled proportionally to the prediction errors. Furthermore, we weighted the reconstruction losses by 2:1:1 for the employed three layer model.

5.2.5 *EEG model*

In the EEG model $q(s_t|e_{t-1})$ in all layers is a diagonal Gaussian with mean and variance parameterized by a convolutional neural network with two layers of 64 and 128 units each. For a three layer model, the convolutional layers are followed by a dense network of 1024, 512 and 256 units respectively. Guided by the idea to assist the model by explicitly capturing spatial information we model all distributions spatially. For the three layer model we used 256, 64 and 16 latent units per layer. Transitions in the model are computed using a convLSTM network [192]. All hidden layers of the encoder, decoder and transition networks use ReLU activations and the final decoder layer uses hyperbolic tangent activations. If not stated differently, we trained all models to convergence of the input layer reconstruction loss with the Adam optimizer. We use a learning rate of 10^{-4} for the first 10 and 10^{-3} for the remaining epochs. We found that initialising with a lower learning rate made learning more stable. We scaled the KL divergence terms with respect to the reconstruction terms at each layer and weighted the reconstruction losses of the layer model 2:1:1.

5.3 LOCATING AUDITORY ERPS IN EEG

Transforming audio features to high-level representations is a complex task, which is often solved with the non-linear processing found in DNNs. Instead of predicting individual frames, we process mel spectrogram representations of audio. The reduced temporal resolution of spectrograms helps reducing the computational complexity while still capturing fine-grained auditory information. The resulting spectrograms spatially extend into two dimensions, time and frequency and can efficiently processed the convolutional decoder of the suggested PC network.

We used the Naturalistic Music EEG Dataset—Tempo (NMED-T) for the evaluations in all presented experiments [123]. As mentioned before in chapter 3, NMED-T features EEG recordings from 10 commercially available music pieces, with durations between 270 and 300 seconds, spanning 55 to 150 BPM in various genres. 20 participants were allowed to freely and passively listen to the music, without any additional cognitive load. We used the provided preprocessed version of the EEG data at a sampling rate of 125 Hz. For all presented ERP experiments, we re-referenced the EEG data to the average of all 125 EEG channels and filtered out background noise using a Savitzky-Golay filter before averaging the evoked responses. For network training, we resorted to the "small" partition of the Free Music Archive (FMA) dataset, featuring 8000 songs with 30 seconds duration [30]. We computed magnitude spectrograms for all ten provided audio files of the NMED-T dataset and the FMA audio files before mapping to the

mel scale, resulting in mel spectrograms at 125 Hz, equal to the EEG sampling rate. All audio processing steps were done with the librosa library [135]. We tested different mel spectrogram lengths as inputs to the lowest network layer and found lengths between 50 and 150 ms to be the sweet spot with low computation time without quick overfitting.

For all audio-based experiments in this chapter, network training was done first on the FMA dataset followed by a evaluation phase using the NMED-T stimuli. After training on the FMA audio, we froze the network weights and processed the NMED-T audio to generate predictions and corresponding prediction errors. For each processed NMED-T audio stimulus we extracted both positive (PPE) and negative (NPE) valued prediction errors. In this context, PPEs refer to areas where the model predictions are lower than the observed threshold, while NPEs refer to predictions that are higher than the actual values. Predictions were computed in a single pass over each song, i.e. without repeated inference of the current musical context.

5.4 DERIVING ERP LOCATIONS FROM PREDICTION ERRORS

In order to inspect the effect of PC at the audio level, we first deactivated the recurrent parts of the lowest layer, forcing the model to express next states as a function of previous observation and the top-down prediction. For model evaluation, we extracted positive and negative prediction errors from each layer of the network. In all layers, we applied a magnitude threshold to pick peaks from the error response, followed by a refinement step that ignores repeated error peaks in a sliding window of fixed size. Both magnitude and window size could be learned by the network itself, leaving the room for more complex and self-supervised segmentation techniques. All audio experiments use the mean of positive and negative prediction errors, if not further specified.

Figure 27 shows two examples for input and predicted audio as well as the corresponding prediction errors and selected peaks. The examples illustrate that autoregressive PC decorrelates large parts of the processed audio in the first layer, by reducing the redundancies in the signal using non-linear weighted predictions based on the past values. This is in line with the spatial and temporal whitening effects described by Rao et al. in the context of center–surround receptive fields in the retina [172]. For the following experiments, we use these sensory predictions to derive ERP locations. Increasing the weight of the prediction errors in the hidden layer decreased the error magnitudes. This is expected, as the network now learns to include more global temporal context over multiple steps of the lower layer. Ideally, the network learns to predict the rhythmic and timbral structure perfectly and successfully suppresses the prediction error in the first

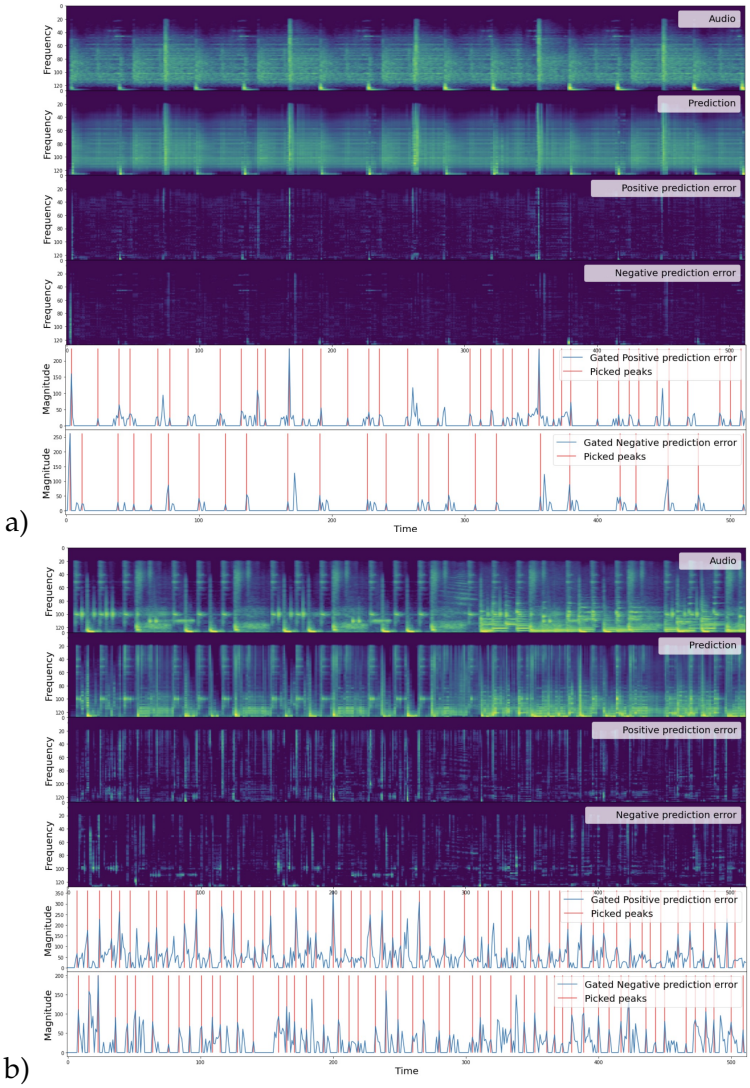


Figure 27: Predicted audio and positive and negative prediction errors in the first PC layer for songs with a) 55 and b) 108 BPM. The visualized model generates local predictions about the next inputs in a sliding window of 50 ms size. This autoregressive and nonlinear process removes temporal redundancy in the residual error response, which can be split into positive and negative parts. The bottom rows show the thresholded prediction error and picked peaks.

and second layer. If the recurrent parts are active in the lowest layer, the long-term temporal dependencies can be memorized in the first layer additionally. In our experiments we found that (with a fixed weighting of the prediction errors between layers) deactivating the recurrence in the lowest layer is essential to learning predictive representations in the hidden layers.

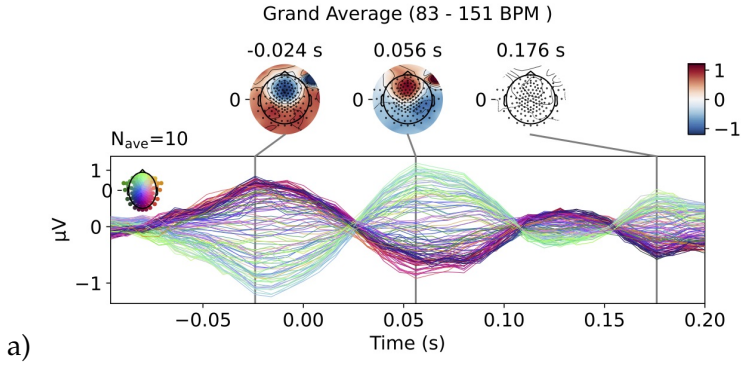
As visible in Figure 27, the tempo of the song, as well as the rhythmic density of the songs have strong influence on the effectiveness of input decorrelation in the lowest layer. While outside of the scope of unsupervised learning, we were able to attenuate this effect by feeding tempo-aligned predictions, which allow the lowest layer to reduce large parts of the temporal surprise. Such (adaptive) temporal alignment could significantly improve network performance with respect to hidden layer predictions in future iterations and is in line with the entrainment of predictions in the human brain [214].

5.4.1 *Grand average ERP*

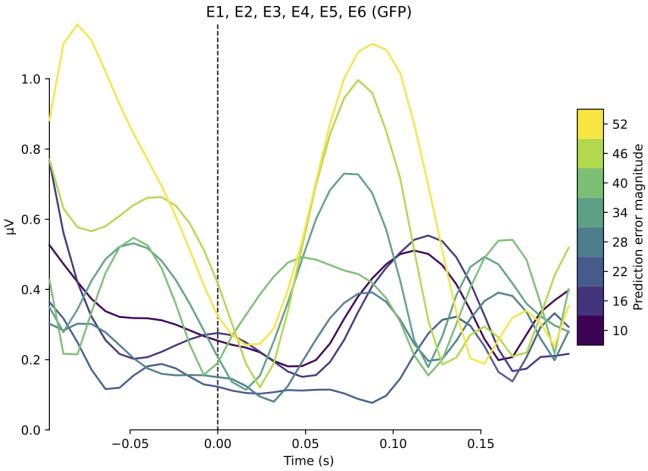
To inspect the possibility to detect ERP events based on the sensory surprise, we extracted the prediction errors from the lowest PC layer and averaged the EEG signal over all trials in all songs and subjects. We were able to derive a total of 242960 trials within 10 songs and 20 subjects using the proposed method. This equates 22140 to 28740 trials per song and between 1108 and 1437 unique event locations per song.

Figure 28 a) shows the grand average ERP for all ten songs in the NMED-T dataset at locations of prediction errors peaks. In comparison to the tempi reported in the orginal NMED-T paper, we sorted the songs between 83 and 151 BPM using beat tracking in the librosa library. The difference between our tempo measures and the ones in the original paper can be explained as being multiples of each other, e.g. 110 BPM being a multiple of 55 BPM. The averaged ERP shows an activity peak for positively correlated channels at around 20 ms previous to the predicted event location, followed by a negative peak around 60 ms after onset. The grand average ERP further shows two smaller peaks around 120 and 170 ms after onset, indicating the presence of surrounding onsets with variable latency. The reduced magnitude of these delayed peaks can be explained by the differences in tempo between songs. Specifically, the difference in peak size between activity close to the predicted onsets and those with greater temporal distance indicates a separation between tempo-independent components (close to the prediction error peak) and attenuated tempo-dependent components.

Figure 28 b) shows the grand average ERP in five positively activated channels, sorted by the prediction error magnitude. The magnitude of the first evoked peak after stimulus onset grows proportional



a)



b)

Figure 28: a) Grand average ERP for all songs in the NMED-T dataset at locations of prediction errors peaks generated by the PC network. b) Grand average ERP in five positively correlated channels for trials sorted after the prediction error magnitude of the PC network.

with the error magnitude for large error values. For smaller prediction error values, the response shows larger latency. Peaks with similar latency of the evoked activity have magnitudes proportional to the prediction error magnitude. This fits with the assumption that the grand average ERP shows temporally variable peaks, induced by differences in tempo.

5.4.2 *Evaluating song-level segmentation*

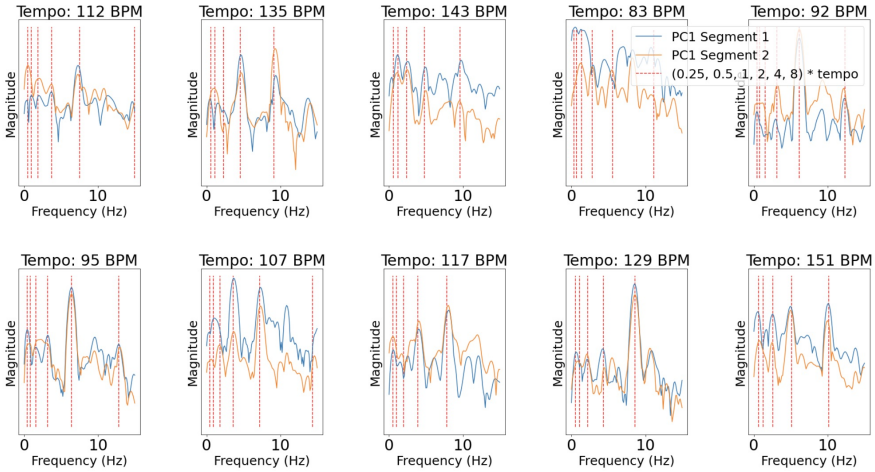


Figure 29: SSEPs in low frequency EEG within the segments derived from gated prediction errors in the first hidden layer of the PC network. Indicated with dashed lines are multiples of the song tempo, ranging from 1 to 16 Hz. Visible differences between the peaks in the power spectrum of both segment indicate different rhythmic processing of the music within the segmentation bounds.

Next to inspecting the predicted ERP responses with the local predictions of the input layer, we want to inspect the possibility to segment stimuli on the song level with the model. For this, we repeat the unsupervised training of the previous experiments, but weight the prediction errors in all layers equally after pretraining for 100000 updates. This approach puts more focus on the temporal consistency of the predictions in the hidden layers. Furthermore, we train with multi-step predictions of length 16, i.e. prediction errors are generated with respect to 16 future states at a time. This follows the assumption that both multi-step predictions and increased weighting of the hidden layer prediction errors increase the network's tendency towards more global predictions. In order to evaluate the ability to retrieve meaningful musical structure with hidden layer predictions, we first segment the whole song with the prediction error of the first hidden layer and subsequently perform SSEP analysis of the low frequency components each song's predicted components separately. Following previous work that illustrates differences in beat processing with

SSEPs, we inspect averages of the low frequency component to detect changes in beat processing or entrainment between the different segments derived from prediction errors in hidden layers of the PC network [123]. Here, we want to inspect whether changes in hidden layer predictions introduced by large peaks in prediction errors show a change that is detectable with SSEPs. To generate binary segmentation, we threshold the prediction error like in the previous steps with a fixed value for each song and switch between segmentation masks when the positive error surpasses the negative error and vice versa. We found that using the mean of the errors between windows of five seconds duration helped to prevent over-segmentation. Intuitively speaking, these changes in hidden prediction error magnitude reflect the "mid-level" surprise of the network, as the pure sensory surprise is largely minimized in the input layer and only the residual errors are further propagated. Future iterations of the model could use learnable error thresholds for improved and self-supervised segmentation. To help visualize the effect of segmentation, we performed Principal Components Analysis (PCA) before averaging the data and analyzed only the first component. Figure 29 shows the induced SSEPs in the power in low-frequency EEG components for selected songs. Visible are peaks in the low frequency EEG components within all segmented parts that are aligned with multiples of the song tempo. In most processed songs, there is noticeable shift in the distribution of induced peaks, indicating rhythmic differences between the annotated segments.

5.5 EEG PREDICTION AND ACTIVE INFERENCE

After evaluating the model's prediction error response to auditory stimuli with respect to temporally aligned evoked responses in the human brain, want now seek to explore the capacity of the model to learn predictive representations from EEG signal. In this context we want to investigate the interplay of observed and expected precision (in terms of an inverse of the inferred variance) of EEG predictions and their applicability to *adaptively* select EEG signal during inference. The following sections motivate a focus on precision as foundation of adaptive processing under the FEP and how adaptive inference relates to the presented PC model.

5.5.1 *Active inference and active learning*

A particularly interesting property of PC networks is that, in their Bayesian interpretation under the Free Energy principle, they provide the necessary perceptual structure that drives actions, adaptive processing and planning in the human brain [55, 57]. Under the Free Energy principle, active components underlying inference such as

(physical) actions or prospective planning to reach rewarding states are often subsumed under the notion of "active inference" [57]. Next to inference, i.e. neural activity on fast timescales, the notion of "active learning" has been formulate in the context of free energy optimisation. Active learning and active inference are based on the idea that intelligent behaviour reduces surprise, either by developing a reliable model of the world or by actively engaging with the environment. Active inference and active learning differ in the type of uncertainty that is reduced: Active learning allows a system to build a predictive model of the world. This focuses on reducing uncertainty about parameters of the model by capturing regularities [187]. Active inference relies on an existing world model and minimises uncertainty about current or next states, i.e. the context [57]. For example, an agent might use active inference to sample colors in the surrounding to reduce uncertainty about whether it is currently inside of a building. Active inference and learning often interact directly, e.g. the ability to map from colors to the context is in turn established by active learning of parameters. [1]

While active inference and learning provide formal descriptions of behavior in their respective time courses, they do not directly cover the question of how these processes are implemented. However, active inference has been cast as a process theory in the context of PC and the human brain [57, 90]. Many computational models of active inference and learning treat planning from the perspective of a decision process on well-defined actions and policies, i.e. sequences over actions [47, 179]. Here, we focus on simpler approach, where notions of active inference and learning are cast in the context of prospective or delayed neural responses [90]. The fundamental idea behind active behavior in the context of PC schemes is that actions influence free-energy optimisation by affecting only the (expected) sensory prediction errors, i.e. by actively sampling observations [53]. Interestingly, the neural mapping from actions to expected sensory observations are often modelled as "simple" reflexes that trigger the activity of stretch receptors in muscles [53]. Here, we focus on investigating the expected precision of prediction errors as they are a fundamental requirement for active inference and learning.

5.5.1.1 *Multi-step predictions and evidence sampling*

Each layer of the proposed network aggregates information about past states with a recurrent memory and can form multi-step predictions. In the previous sections on audio processing, we have taken a straightforward approach and have trained the network using a

1 It should be noted that in the domain of machine learning, the term active learning is used mostly to denote specifically the active query of new annotated data. Here focus on active learning in the broader sense and in the context of unsupervised learning [27].

bottom-up approach, where (multistep) predictions are directly compared to the observations. Assuming appropriate scaling of the KL divergence regularization terms, this results in representations that are driven primarily by the sensory data itself, in terms of a bottom-up pass of information. Here, we take a more elaborate approach and train the network by alternating between a bottom-up pass and a top-down pass. The bottom-up pass refers to first updating the sensory layer and then updating the higher hierarchical layers with respect to the ascending prediction error. The top-down pass operates in the opposite direction and propagates the prediction of the highest hierarchical layer first, followed by an update of the lower layers with respect to the descending prediction.

In each hierarchical layer predictions are made over a sequence of latent states. As a result, there are multiple observations (one for each predicted state) that provide an error signal, or an "evidence" for the correctness of the predictive model. In the context of precision weighted expectations about states, we can now think about an adaptive process that decides between observations that are included and those that are ignored (i.e. have low expected precision). In the extreme case of updating a particular layer without any bottom-up information, the predictions will still be refined with respect to the top-down information. This potentially leads to predictions that are internally consistent but are less and less predictive about the actual data. The following experiments thus focus on investigating the top-down inferred uncertainty in comparison to the observed uncertainty. Next to analysing these precision signals, we will inspect the possibility to use the learned representations to actively select temporal locations in the EEG data that best fit the learned representation.

5.5.1.2 *EEG dataset and preprocessing*

As investigated in chapter 3, learning representations from auditory EEG is a complex task, even if the perceived stimulus is available. To simplify our investigation, we resort to EEG data recorded in the context of Fixation Related Responses (FRPs). FRPs are a variant of ERP that are temporally aligned to eye fixations. This requires additional eye-tracking technology to record the saccadic eye movements. From an experimental point of view, FRPs have the advantage that they provide clear labels, in terms of detected fixations, about *when* the evoked response is observable in the brain [9]. This removes the need to derive possible ERP onsets from the stimulus itself, removing possible sources of ambiguity, such as whether or not the stimulus has been perceived (or attended to) at all. Next to a clear temporal labelling of the evoked response, FRPs have the quality of being actively generated by the human subject [9]. For these reasons, we resort to the Zurich Cognitive Language Processing Corpus (ZuCo), a simultaneous EEG and eye-tracking dataset for natural sentence reading for model train-

ing and evaluation [82]. The dataset was designed specifically with training machine learning systems in mind and provides substantial amounts of recorded EEG signal. We used all available subject data within the second task, where subjects were asked to read English sentences displayed without any additional task. This resulted in a total of 93184 fixations in 12 subjects. We used the preprocessed EEG following the routine described in the paper and resampled the data to 448 Hz [82]. Differently to the authors, we did not disregard fixations based on their duration. The EEG data was aligned to fixation onsets based on the provided eye-tracking data. We split the data on a per-subject basis and used 60% for training. This allows to evaluate the model for unseen inputs for each subject as well as the average performance. We furthermore extracted the fixation duration for the purpose of model evaluation but did not feed any metadata during model training. In the following experiments, the model was trained to predict consecutive EEG inputs of 8 samples duration in the input layer.

5.5.2 *Autoregressive EEG prediction*

We first inspect the model's capacity to reconstruct multi-channel EEG signal directly after each processed time-step in the input layer. At each step, the model predicts the next EEG input by aggregating the previous posterior, the deterministic memory and top-down prediction. We train the model with focus on the input layer and scale the prediction errors of the remaining two layers by 0.1 before error backpropagation. As we are interested in the performance of bottom-up driven prediction, we evaluate the difference between predicted and observed variance, without additional precision weighting.

This can be seen as a non-linear probabilistic version of linear PC (LPC), which is a well established method for audio compression and speech synthesis [158]. In comparison to our method, LPC lacks top down predictions and multi-step latent prediction. We were able to use both single and multi layer networks to reconstruct expected EEG. While these examples were trained on FRP onset aligned inputs, predictions for continuous EEG looked comparable. One common observation in all tested configurations is that learning the oscillatory pattern was improved by gradually increasing the range of posterior means in hidden and the input layer.

We perform an ablation study with three variants of the model. For this, we keep all model parameters constant but allow the models to differ in the way future steps are predicted. The first variant performs inference over a single step after observing prediction error once. The second variant performs multi-step predictions in the first hidden layer and updates the input layer by treating its own predictions as actual observations. Importantly, the prediction feedback is done only

Model	PSNR	MSE
Single step latent prediction	49.9	14.8
Multistep prediction & internal feedback	50.5	13.0
Multistep prediction & autoregressive inputs	52.2	8.6

Table 7: MSE and PSNR (in dB) for multi-channel EEG prediction

at test time. The third and final variant combines latent multi-step prediction and autoregressive processing of actual inputs in the input layer.

As metrics we resort to the mean squared error (MSE):

$$\text{MSE} = \sqrt{(\frac{1}{mn})\sum_{i=1}^{m}\sum_{j=1}^{n}(y_{ij} - x_{ij})^2} \tag{46}$$

between EEG signal x and prediction y, where m denotes the number of channels and n denotes the number of samples in each prediction. We further compute the peak signal-to-noise ratio (PSNR) between input signal and predictions defined as

$$\text{PSNR} = 10\log_{10}(\frac{\text{MAX}^2}{\text{MSE}}) \tag{47}$$

where MAX refers to the maximum EEG value.

5.6 RESULTS ON EEG

As listed in Table 7, the multi-step model outperforms the single step baseline in terms of EEG prediction quality, despite the increased complexity of multi-step predictions. Feeding back predictions to the multi-step model also results in an improvement over the single step baseline, signifying gains both in autoregressive and one-shot prediction.

5.6.1 *Comparing observed and inferred precision*

A key requirement for meaningful evidence sampling is that the model is able to predict and weight the precision with respect to the input data as well as model accuracy. This holds especially for sequential predictions, where information can be aggregated over time for refined prediction. Observed and predicted average precision for a three layer model performing next step prediction in the input layer

is displayed in Figure 30. The distribution of observed precision mirrors the fact that the predictions are largely driven by the reconstruction term of the input layer. While the lowest average precision is observed in the first 5 steps, the highest average precision is observed at the end of the sequence, where most context was aggregated. The highest average predicted precision was reached for the first step and the final two steps. This is compatible with the idea that the variance is easiest to predict with maximal temporal context (in the final steps of the predicted sequence) and at the beginning of prediction, with minimal context available for precise EEG prediction.

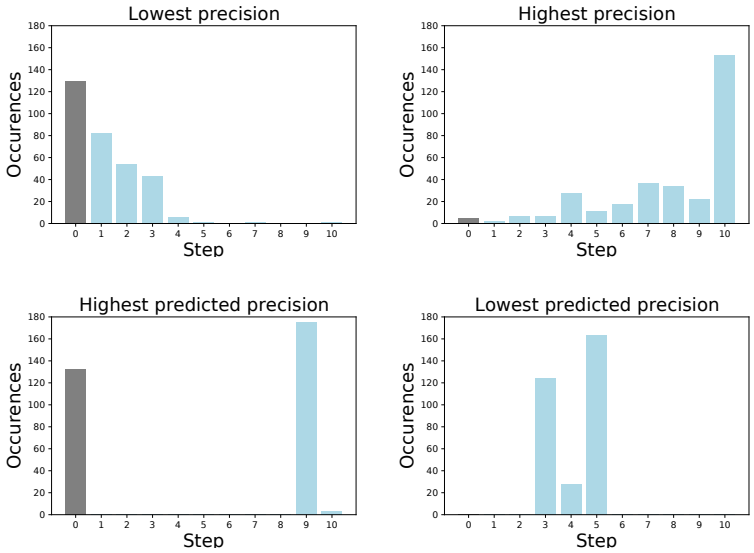


Figure 30: Distribution of observed and predicted averaged posterior precision in the first layer of a three layer model. Listed are number of times where the step had the highest or lowest average posterior precision within the sequence. The bottom row shows the precision predicted in the top-down pathway. All values refer to the average precision of a latent state and are computed as the mean of all spatially distributed units within 10 batches of the test set. The precision in the first step without temporal context for filtering is indicated in grey.

5.6.2 *Fixation Related Potential prediction*

Figure 31 visualizes evoked FRPs from averaged model predictions in across all subjects at test time. Shown are single trial EEG segments that were clustered by the fixation duration before averaging. The averaged predictions show the first positive P1 that is followed by second activation, the P2 component, with onset proportional to the fixation duration. This is in line with previous studies that cluster FRP data based on fixation duration [82].

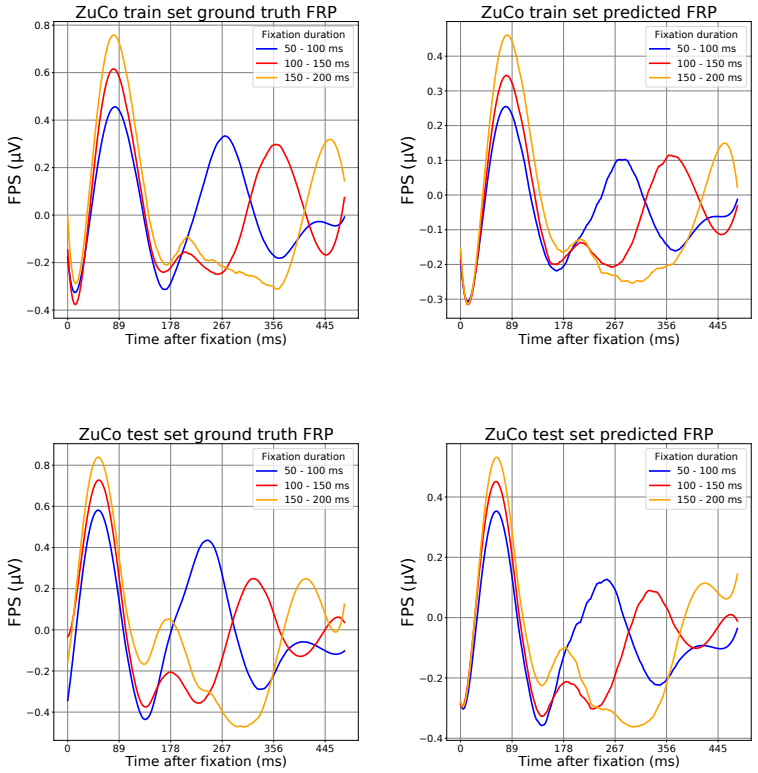


Figure 31: FRPs from averaged input and predicted EEG signal. Predictions were generated using multi-step latent predictions over 15 steps (120 EEG samples) and autoregressive processing of inputs. The EEG trials were sorted after fixation duration before averaging. Input and predicted EEG signal show a positive P1 peak (around 100 ms after the fixation) that is followed by a fixation duration dependent second component P2 (starting around 200 ms after onset).

For onset aligned as well as continuous inputs, averaged model outputs from multi-step latent predictions showed the characteristic activity peaks after fixation onset. Fixation onset aligned predictions looked meaningful even for a single internally propagated time-step prior to sequential prediction onset. In contrast, predictions without autoregression in the input layer on continuous data relied heavily on the prediction errors before sequential predictions and did not lead to meaningful predictions with fewer than 4 input steps. This indicates that onset aligned training leads to a stronger prior preference of the network to predict FRP-like signal. The next section explores the application of these priors for active inference.

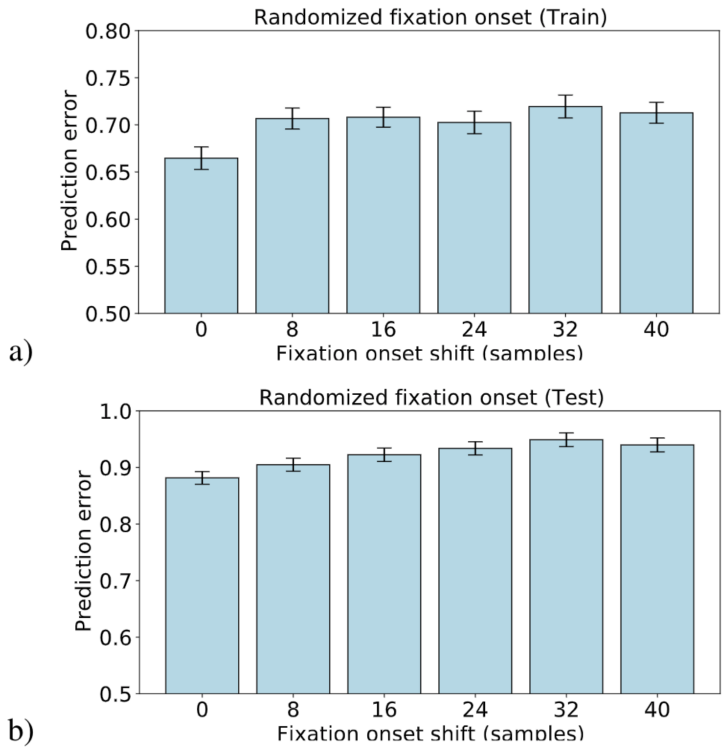


Figure 32: Prediction errors during active inference of the fixation onset position in the a) train and b) test set. Shown are the mean prediction errors and 95 % confidence intervals for multi-step latent predictions over 9 continuous input windows (72 samples). In both subsets, shifting the inputs away from the fixation start point increases prediction error. This prediction error is used for rapid estimation of the context by inferring the position with lowest prediction error without updating model weights.

5.6.3 *Active inference of FRP locations*

Often times, behavior is learned jointly by active inference and active learning, especially when the inferred regularities are stable between trials. When key aspects appear randomly in trials however, active learning is less efficient. In these situations (long-term) learning relies largely on a correct estimation of the context with active inference [187]. Here, we want to look at such a case, where the prediction of FRPs is performed on EEG data with randomized fixation onset locations. We were able to employ the previously introduced multi-step predictions for active inference of the correct fixation onset by comparing the averaged sequential latent prediction in each batch with randomly shifted EEG inputs. For this, we used the trained model with frozen encoder and decoder weights to actively correct the randomized shift. We shifted the inputs through 6 possible offsets and update the inferred state at the position leading to minimal prediction

error between observed and preferred signal. Figure 32 shows that the averaged prediction errors increase towards larger shifts from the FRP onset location. This allows to correct the randomized shift based on prediction errors between the decoded prior (i.e. the preference for FRP-like signal) and the actual EEG signal.

5.7 DISCUSSION

5.7.1 *Locating auditory ERPs with prediction errors*

We demonstrated the application of the deep PC network for unsupervised audio representation learning. We compared the network's prediction errors with evoked potentials in the human brain. For this, we related the hierarchical predictions of the model on ten naturalistic musical pieces to onset-aligned evoked potentials captured in EEG recordings of the songs. We derived temporal locations for individual musical events from the sensory surprise and inspected steady-state evoked potentials that capture rhythmic differences in the segmented songs.

The employed PC model combines deterministic sequential predictions with probabilistic representations. While the deterministic parts allow to learn regularities over time-scales, the probabilistic elements lessens overfitting and helped shortening training duration. While sensory-level predictions can be employed for local event annotations, the predictions and prediction errors in hidden layers target higher levels of temporal abstraction. Here, we qualitatively analysed the possibility to derive song-level segmentation with multi-step predictions generated in hidden layers.

Our results indicate the usefulness of PC models for the retrieval of musically meaningful events across the local and global structure of musical works. The model allows to approach audio segmentation jointly with structuring recorded brain activity, forming a basis for retrieval of information about cognitive processes in music perception. This offers an appealing method for researching auditory evoked potentials, as it eases the mapping between stimulus characteristics and connected evoked potentials across time-scales. Future improvements could enhance the capacity of the model, e.g. by allowing the model to segment inputs based on learned error gating.

5.7.2 *EEG prediction*

The results on direct EEG prediction indicate the possibility to learn meaningful representations from EEG signal with deep variational predictive coding. For this we introduced a hierarchical probabilistic state-space model with multi-step latent predictions in each layer. While the lowest layer expresses predictions about the EEG signal,

higher layers sample evidence by predicting the distributions parameters of lower layers. We demonstrated the possibility to generate multi-step latent predictions in hidden layers and their application to sampling new inputs in active learning and active inference. The model can be used for adaptive input processing by comparing predicted and actual uncertainty and reducing prediction errors based on the learned preferences for future EEG signal. Quantitative and qualitative evaluation of the predicted signals near eye fixations were presented. Future work could scale this up to more elaborate active inference implementations, for example by incorporating explicit policies and physical actions in real-time brain-computer interfaces.

5.7.3 *Potential drawbacks of the model*

While providing promising early results, the particular chosen model also has several drawbacks, which can potentially limit its performance and its role as a model of canonical computation. Firstly, the chosen approach to explicitly predicting sequences of latent states could potentially limit the model due to the lack of a more elaborate separation between represented states and their dynamics. Many established PC models in the neuroscience literature separate between aspects addressing inferred dynamics ("hidden states") and possible causes that induce these dynamics as a sort of control parameter ("cause states") [49, 51]. These models allow to abstract information between hierarchical layers that show abstraction with respect to hierarchical and temporal dynamics, instead of "simply" modelling higher hierarchical layers with slower updates. A second potential drawback of the suggested architecture is the inclusion of an explicit complexity term (i.e. the KL divergence from the prior) within each hierarchical layer. This approach requires a manual tuning of the regularization with respect to the weighting of accuracy, in terms of additional hyperparameters. In this context, more elaborate implementations could, for example, focus on relying only on the top-down signals as empirical priors. From the perspective of biological plausibility, the reliance of the model on the backpropagation of error algorithm towards all parameters also poses problems. Models that rely on strictly local learning rules, i.e. Hebbian learning, in context with complex spatio-temporal observations still need to fully scale up towards the performance achieved with deep neural networks and exact error backpropagation. Nevertheless, they provide the possibility to fully separate inference and learning within each hierarchical layer. Intuitively, this results in separate, and smaller models with less parameters to be learned, potentially allowing to scale up the complexity of the overall model. Such locally informed process models will be covered in chapters 6 and 7.

6

GRADIENT-BASED PREDICTIVE CODING

The previous chapters have covered models of canonical computation under the FEP that rely on credit assignment via exact backpropagation of errors. As they are inspired by hierarchical PC in neuroscience, deep PC models in particular can be connected to a rich theoretical account that explains a variety of cognitive phenomena arising in human processing of music. While these models efficiently learn from complex observations, the biological plausibility of the underlying credit assignment via exact error propagation across the entire model is limited. In particular the required separation of computation into a global "feedforward structure" addressing predictions and a global "feedback structure", addressing credit assignment is difficult to integrate with established neural process theories, such as predictive coding. This chapter will deal with a class of artificial PC networks that learns strictly using locally available information. We still deal with models in the context of gradient computation with automatic differentiation. In this chapter, we will refer to these "gradient-based" models simply as PCNs. While PCNs have been applied to unsupervised representation learning from static information, such as images, their application to complex sequential data, such as audio, is still underexplored. In this chapter, we want to investigate a PCN model that predicts future audio signals and learns to integrate top-down information during dynamical prediction. For this, we discuss and build upon the connections between Infinite Impulse Response filters, Kalman filters, and inference in PC networks. We then evaluate the possibility to use to the network for a beat tracking task and the possibility to perform audio filtering. We find that the model is useful for the presented audio processing tasks, although our results also indicate that the performance is strongly dependent on the chosen dataset.

The content and figures in this chapter is based on the following publication:

> [pub:2] A. Ofner, J. Schleiss, and S. Stober. "Hierarchical Predictive Coding and Interpretable Audio Analysis-Synthesis." In: *Proceedings of the 15th International Symposium on Computer Music Multidisciplinary Research (CMMR)*. 2021, pp. 225–234.

6.1 INTRODUCTION

Research on human auditory processing has demonstrated that humans are efficient at tracking stochastic auditory regularities and can

even disentangle stationary parts, e.g. fundamental frequencies, from dynamic transformations, e.g. resonances, in musical events. From a signal processing perspective, PCNs with a single layer already deliver useful computations, like the source-filter separation in Linear PC (LPC), a widely used Digital Signal Processing (DSP) method. However, in order to live up to their full potential, PCNs need hierarchical structure. The number of existing studies employing PC to process raw audio signal is limited and the available methods are generally difficult to interpret. PCNs treated in neuroscience are generally restricted to simple auditory stimuli or even symbolic inputs [137, 195]. Still, there are striking similarities between the structures of Infinite Impulse Response (IIR) filters and recurrent neural networks (RNN), classes that are already widely used in signal processing applications and those models that address human auditory cognition more specifically, in particular the Kalman filter or PCNs. We will cover these similarities later in this chapter. A major challenge when employing gradient based PCNs for signal processing tasks is that they only deliver approximate results during learning and inference. This poses a major drawback for those tasks where high accuracy is required. Furthermore, it is difficult to design efficiently operating hierarchical PC models, which would have the advantage of naturally scaling to larger signal processing systems while allowing meaningful cognitive interpretations. To solve these challenges, we resort to the structural similarities between PC models and established DSP methods in the next section and then introduce a hierarchical PC model for temporal prediction.

6.1.1 *Predictive coding and error backpropagation*

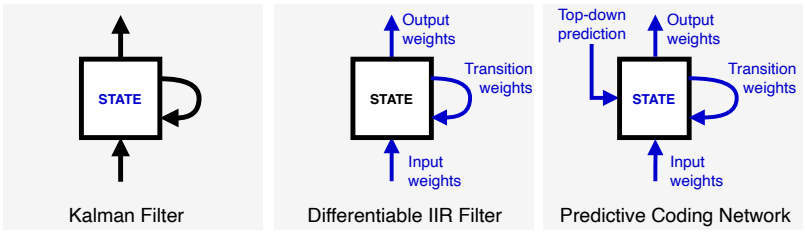
Exact error propagation is an effective method to train large feedforward networks, by computing the gradient of a global loss function with respect to all optimised parameters of the model. In this context, models that contain recurrent connections are unrolled over time steps and treated as a feedforward architecture, allowing to backpropagate errors though time. While efficient, such exact error backpropagation is in contrast to *locally* informed inference and credit assignment described by process models in neuroscience, such as PC models under the FEP [49, 51]. Such models generally have a distributed structure, where errors (in terms of Bayesian surprise) are represented locally. These process models can be mapped onto a structure that is repeated across the human cortex indicating their biological plausibility as a canonical computational motif [11]. Much criticism has focused on the issue that the application of exact error backpropagation involves an algorithm that operates *separately* to the feedforward pass, i.e. the predictions [226]. Furthermore, exact error backpropagation requires a symmetry of forward and backwards weights, since

the back-propagation of information requires an exact copy of the weights used for the forward pass[226]. Examples for such reciprocal and symmetrical weights exist in the brain, but are not as omnipresent as would be required for exact error backpropagation [196, 226]. However, there have been many attempts to mitigate the differences between locally informed learning and exact error backpropagation [120, 182, 197, 225, 226]. Many of these approaches focus on demonstrating that backpropagation of error can approximated by locally informed learning rules, such as PC networks or contrastive Hebbian learning [182, 225].

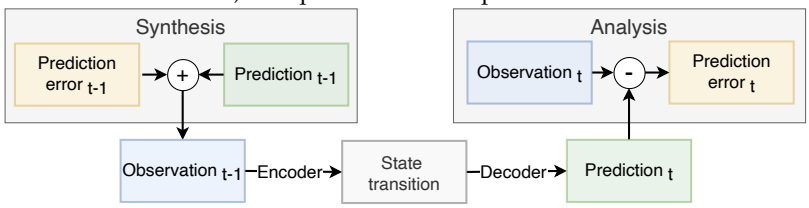
6.2 RELATED WORK

The similarity between IIR filters, Kalman filters, RNNs, and PC networks is particularly apparent when one views these models in their state space model (SSM) form. Figure 33 a) provides an overview of these related classes in state-space form, such as they are used in tasks typical for each class. Aspects of learned model structure, such as filter coefficients, are referred to as weights in the context of artificial networks. Generally speaking, "inference" refers to employing these given coefficients (i.e. weights) to update hidden representations, while "learning" refers to the slower process of optimizing weights.

While the signal flow of the model classes is directly comparable, differences arise in the way inference and learning are addressed in typical tasks. Kalman filters are usually used for dynamic inference given prior assumptions on the data, resulting in mathematically exact updates of their latent state. The deterministic class of IIR filters is typically used to apply a previously designed transfer function to incoming signals, where output signals are a weighted combination of previously processed signals. Some exceptions, such as differentiable IIR filters allow to learn weights during application [113]. Kalman filters and PC networks are typically modeled as probabilistic generative models, keeping track of an inferred latent state with associated variance (or inverse precision). Both have found applications in modeling cognitive and neural processes. In contrast to Kalman filters, optimization in PC networks generally addresses state inference and weights learning simultaneously. Finally, PCNs can include internal predictions of their latent states, i.e. "top-down" expectations about activities in lower PCN layers [2, 51]. This hierarchical structure is similar, but not identical, to the multi-layer architecture of deep neural networks, which typically lack the feedback connections that are inherent to PCNs. More specifically, DNNs can be interpreted as corresponding to pyramidal dendritic connections in the biological counterpart. This means that DNNs, possibly with multiple layers, connect adjacent variables in PCN layers [127]. Here, we explore the audio DSP



a) Comparison in state-space form



b) Signal analysis and synthesis with autoregressive predictive coding

Figure 33: a) Comparison of Kalman filters, differentiable IIR filters, and gradient-based PC networks in state-space form. Blue color indicates variables that are optimized in a typical filtering application for each model. b) Signal analysis and synthesis with autoregressive PC and linear activation functions: In the analysis stage, observations at time-step t are mapped to hidden states using encoder weights. The learned transition dynamics are then applied to the latent state. Outgoing predictions for the next timestep $t+1$ are computed via decoder weights that map from the updated latent state to the expected sensory input. During synthesis, the prediction error is fed to the model jointly with the previous prediction.

capabilities of single-layer and hierarchical PCN models interpreted as biologically plausible Neural Kalman filters. This PCN class has been discussed for single-layer models in [141].

6.2.1 *Audio filtering and state-space models*

Signal analysis with autoregressive filters at discrete time-steps t can be described with respect to a steady state transfer function $H(z)$

$$H(z) = \frac{G}{1 - \sum_{j=1}^{k} a_j z^{-j}} = \frac{G}{A(z)} \tag{48}$$

with input gain G [89, 109]. The parameters a_j with $1 \leqslant j \leqslant k$ and G of this state transfer function can be optimized with respect to the prediction error $e(x)$ between predicted signal $p(t)$ and observed signal $o(t)$, also referred to as excitation or residual signal:

$$e(t) = \frac{1}{G}(o(t) - \sum_{j=1}^{k} a_j o(t-j)) \tag{49}$$

The SSM of this generalized prediction error filter is updated with the following difference equation:

$$\begin{aligned} z[t+1] &= A[t]z[t] \\ o[t] &= C[t]z[t] \end{aligned} \tag{50}$$

where $z[t]$ is the state vector at timestep t and the prediction coefficients a_j are represented by weights A and C. All four discussed model classes, despite originating from the different fields can be interpreted in prediction error minimizing SSM form. Linear PC (LPC), a widely used DSP tool, draws from this possibility for the design of IIR coefficients. LPC is typically used for signal compression, particularly for speech coding, by separating stationary residual signals from imposed resonances [158]. This theoretically allows to analyse and synthesize signals using the same model. However, the efficient algorithms employed in LPC are not directly biologically interpretable and generally do not actually use a SSM to find the coefficients. From this perspective, our work generalises LPC towards the more general class of PCNs, where analysis and synthesis use the same model.

6.2.2 *RNN and differentiable IIR filter*

Recurrent neural networks, in their simplest form, can be expressed by the following difference equations [42, 113]:

$$\begin{aligned} z[t+1] &= \sigma_z\left(W_z z[t] + U_z x[t+1] + b_z\right) \\ y[t+1] &= \sigma_y\left(W_y z[t+1] + b_y\right) \end{aligned} \tag{51}$$

with hidden states z, inputs x and outputs y. W and U are trainable weights and b are biases. Known from previous work is that, in the case where activation functions σ are (non-)linear and the biases are set to zero, this structure directly resembles a (non-)linear all-pole IIR filter

$$\begin{aligned} z[t+1] &= W_z z[t] + U_z x[t+1] \\ y[t+1] &= W_y z[t+1] \end{aligned} \tag{52}$$

which scales to arbitrary order of transfer functions $H(z)$ (also referred to as the filter order) and allows to train differentiable IIR filters using the optimization methodology for RNNs [113]. A useful generalized state space form for such IIR filters is

$$\begin{aligned} z[t+1] &= Az[t] + Bx[t] \\ y[t+1] &= Cz[t+1] + Dx[t+1] \end{aligned} \tag{53}$$

where matrices A, C represent the learnable weights for latent state transition and output transformation and B, D are weights for input transformations [113].

6.2.3 *Kalman Filters*

The Kalman filter gained large popularity in fields such as engineering, statistics, and neuroscience and filters data points with respect to a probabilistic latent state and their expected precision. Typically, dynamics and observation models are linear and the observed noise and the latent states are modeled as Gaussian distributions. Similar to the previously discussed model classes, the Kalman filter can be described in SSM form:

$$\begin{aligned} z[t+1] &= Az[t] + Bu[t] + v \\ y[t+1] &= Cz[t+1] + w[t] \end{aligned} \tag{54}$$

with hidden states h_t at discrete timesteps t. Correspondingly to the deterministic IIR filter, the weights of the transition matrix A describe the linear dynamics. The weights of matrix B and C parameterize the observation model. Weights B transform the control inputs u, i.e. known inputs to the system and C map from inferred state to the sensory prediction. Finally, v and w are white noise Gaussian processes with mean zero. The Gaussian prior $p(z_{t+1})$ and posterior distribution $p\left(z_{t+1} \mid y_{1\ldots t}, x_t\right)$ of the Kalman filter are parameterized by their sufficient statistics, the mean μ and covariance matrix Σ_z [95, 141].

6.2.4 *Gradient-based predictive coding*

Gradient-based PC has been applied to an approximation of the exact inference in the Kalman filter [141]. In the simplest case, without observations or control inputs, we have a state space model of the form

$$\begin{aligned} z[t+1] &= Az[t] \\ y[t+1] &= Hz[t+1] \end{aligned} \tag{55}$$

where A and H are learnable matrices for the state transition dynamics and the observation model respectively.

Following [141], we define the loss function of the PC filter as:

$$\operatorname{argmin}_{\mu_{t+1}} L = \operatorname{argmax}_{\mu_{t+1}} p\left(y_{t+1} \mid z_{t+1}\right) p\left(z_{t+1} \mid z_t\right) \tag{56}$$

In this formulation, weights A and H and the inferred hidden state z (or, more specifically, its mean parameter μ) can be updated using a gradient descend on the precision weighted prediction errors local to the layer [141]:

$$\begin{aligned}\frac{dL}{d\mu_{t+1}} &= -H^T\Sigma_z\epsilon_z + \Sigma_x\epsilon_x \\ \frac{dL}{dA} &= -\Sigma_x\epsilon_x\mu_t^T \\ \frac{dL}{dC} &= -\epsilon_y\mu_{t+1}^T\end{aligned} \tag{57}$$

with sensory prediction errors $\epsilon_y = y - H\mu_{t+1}$ and state prediction errors $\epsilon_z = \mu_{t+1} - A\mu_t$ [141]. This means that each layer optimizes the quality of its signal predictions $p_{y_{t+1}} = H\mu_{t+1}$ and of its state predictions $p_{\mu_{t+1}} = A\mu_t$. Often times, the variance Σ is assumed to be constant during inference and learning, i.e. fixed to a prior hyper-parameter, such as a simple identity. Here, we will also resort to a fixed variance (an identity matrix), although more elaborate implementations could adaptively weight the prediction errors with respect to an inferred variance. In this thesis, we will cover such variance estimation in PC networks in chapter 7, in the context of unsupervised learning on spatial and spatio-temporal data. This optimization process happens locally informed and in parallel for each optimized variable. This implies that many different possible configurations of states and network parameters could be found during minimization of the locally computed prediction errors.

A more general form of the PC SSM includes additional weights for control inputs u and observed inputs x:

$$\begin{aligned}z[t+1] &= Az[t] + Bu[t] \\ y[t+1] &= Hz[t+1] + Dx[t]\end{aligned} \tag{58}$$

In summary, we see that single layer PC models and Kalman filters can be represented using the same SSM as IIRs and RNNs (excluding nonlinearities), but additionally differentiate between control and observed inputs.

6.3 HIERARCHICAL PREDICTIVE CODING OF AUDIO

To create a hierarchy of dynamical layers with local computations, we can augment the PC SSM mentioned in equation 58 with two sets of weights, F and G. These weights modulate the influence of the layer's own latent state z in comparison to a top-down prediction of this state z_{td} provided by a higher layer:

$$\begin{aligned}z[t+1] &= FAz[t] + GAz_{td}[t] + Bu[t] \\ y[t+1] &= Hz[t+1] + Dx[t]\end{aligned} \tag{59}$$

Here, we denote the weighted state prediction from current and next higher layer with $\hat{z} = Fz + Gz_{td}$. In all experiments, we ignore

control inputs u, which could receive known additional (action) signals and feed past observations x_{t-1} to the observation encoder for the filtering task presented in section 6.4.3.

The state prediction error now includes the additional input and weights:

$$\epsilon_z = \mu[t+1] - FA\mu[t] - GA\mu_{td}[t] \tag{60}$$

Figure 34 shows an overview of a single layer PC model and how multiple layers can be connected through locally informed predictions and prediction error signals. More precisely speaking, the lowest PC layer directly predicts audio inputs and receives prediction error e_t at every timestep. In contrast, hidden PC layers predict the hidden states of the lower layer and receive state prediction error e_t^z. Both lowest and hidden PC layers additionally optimize the weights of their transition model that maps from currently inferred state z_t to the next state $z_t + 1$. We can interpret weights F and G as part of the prediction units that produce the optimal state predictions z_{t+1} given the transition model A. Finally, the latent state $z_t + 1$ is optimized in parallel via gradient descent to minimize the prediction error $e_t + e_t^z$ local to the respective layer.

We use an overlap-and-add processing approach which is commonly used in DSP, meaning that the PCN processes audio signals in overlapping sequences. For all experiments, the lowest PCN layer processes these sequences sample-by-sample. Hidden layers have identical update frequencies. We found that sequence sizes between 16 and 2048 frames provide meaningful results. The hop-length was set to half the sequence length.

6.3.1 *Audio analysis and synthesis*

Assuming purely linear prediction and a well-trained model, using the PCN for audio re-synthesis is possible by reverting the process that computes the residual signal at timestep t (i.e. linear prediction error) from the prediction during analysis. Figure 33 b) shows an overview of the steps for synthesis and analysis given at the lowest layer of a hierarchical PC model. While this is not the only possible approach to analyze and synthesize signals with PC networks, it has the advantage of relatively exactly replicating the approach taken in LPC. In LPC the coefficients minimizing the squared error during the linear prediction of the next sample resemble compressed versions of the resonances (typically formants in speech coding) and allow the signal to be transmitted with high compression rates through blockwise filter coefficients and down-sampled residual signals. For linear prediction, this LPC residual signal is equal to the prediction error that arises in (gradient-based) predictive coding. Assuming linear

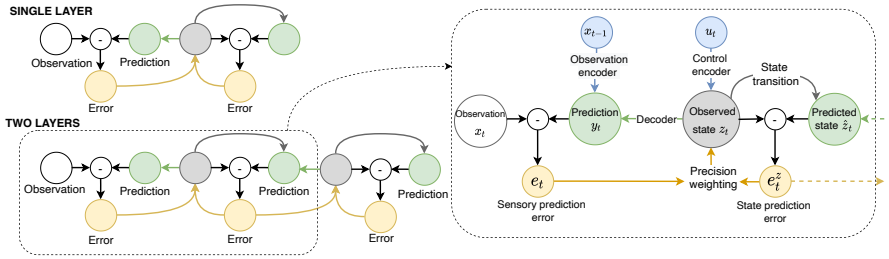


Figure 34: Predictive Coding network for hierarchical Kalman filtering: At each timestep t, predictions y_t are generated from a latent state z_t using decoder weights that are optimized towards the sensory prediction error e_t between observation x and prediction y. Future latent states z_{t+1} are computed with learnable transition weights. The transition weights are optimized towards the state prediction error e_t^z between predicted state $\hat{z}_t$ and the next inferred state z_t. Hidden PC layers minimize the prediction error e_t^z from a "top-down" prediction of the state. The hidden state z is optimized towards sensory and state prediction error e_t and e_t^z and creates a balance between outgoing and incoming predictions. Optional encoders allow to predict with respect to past observations x_{t-1} or control inputs u.

PCN weights and audio with stationary parts, we expect that resonant parts of the audio are gradually removed from the residual.

6.4 RESULTS

6.4.1 *Beat tracking*

In order to quantitatively assess the possibility to extract music information from raw audio using prediction errors, we resort to a beat tracking task using two datasets: The SMC MIREX dataset is commonly used for beat tracking evaluation [83]. Our second evaluation is based on finger tapping recordings in the NMED-T dataset that focuses on EEG recordings during music perception [123]. We choose an approach similar to the predominant local pulse (PLP) method described in Grosche et al. [68] and predict beat timings based on a local enhancement of a novelty function. The novelty function in [68] is based on spectral flux, the spectral difference between subsequent Fourier transformed audio inputs. We feed Fourier transformed audio inputs to the PCN (this being the only place where the PCN inputs are not audio samples) and use the prediction error from a single layer PCN to compute the novelty curve. Wherever possible, we use the same FFT parameters as used in Grosche et al. [68] but do not tune any other hyper parameters. For comparison to other approaches, we report the F-measure and two continuity-based metrics: CMLt, measuring correctly tracked beats at the metrical level, and AMLt, which allows variations such as double, half or offbeat variations [28]. All

SMC MIREX	F-Score	CMLt	AMLt
Ellis, 2007	0.339	0.162	0.315
Grosche, 2010	0.360	0.071	0.221
Böck, 2014	**0.521**	**0.363**	**0.433**
PCN (ours)	0.205	0.108	0.201
NMED-T	F-Score	CMLt	AMLt
Ellis, 2007	0.277	**0.195**	**0.473**
Grosche, 2010	0.305	0.037	0.125
Böck, 2014	0.092	0.105	0.280
PCN (ours)	**0.321**	0.111	0.295

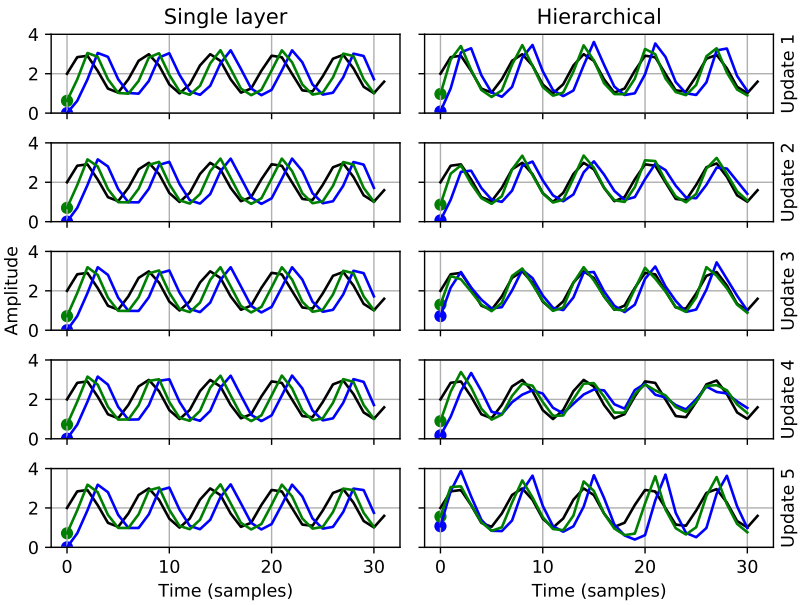
Table 8: Beat tracking evaluation

evaluations are based on the mir_eval package [169]. Next to the PLP model, we compare our approach to established baselines: A dynamic Bayesian network from [13] and the dynamic programming approach from [41]. Table 8 shows resulting scores on both datasets.

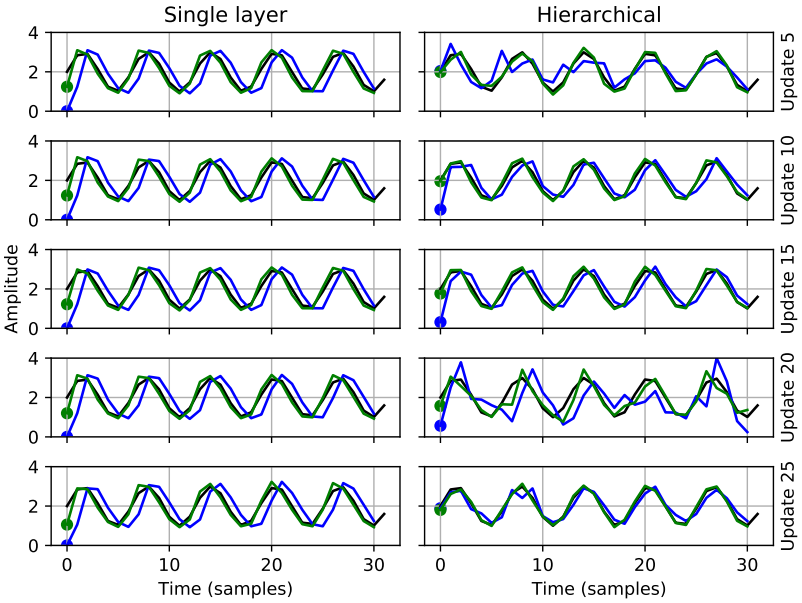
Interestingly, with respect to the F-Measure, our method outperforms the baselines on the NMED-T dataset but delivers the worst performance on the SMC dataset. This indicates a useful performance on genres with salient rhythmical features, as the NMED-T dataset was designed focusing on pop songs with clear rhythms. The SMC dataset features many songs with soft onsets, such as strings, where the novelty function from the prediction error is not sufficient. We hope that these encouraging results motivate future work with improved tracking based on predictive coding.

6.4.2 *Audio filtering with top-down predictions*

Figure 35 shows examples for repeated block-wise prediction of the same audio input with a single layer PCN and a hierarchical PCN with two layers for different gradient steps. In both networks, the inferred state and transition weights of the lowest layer are reset after each sequence prediction. This means that predictions in the single layer PCN are based on local information, i.e. the previously seen samples in the sequence. The hierarchical PCN keeps a top-down prediction of the lower layer's hidden state, providing refined contextual information for each prediction. This learnable state prior noticeably leads to a shifted starting point for the lowest layer in the hierarchical PCN in Fig. 35 a), where the lowest layer has not enough time to converge properly. When initialised with optimised parameters, both variants are able to approximate the target audio to a reasonable degree and the differences in prediction (and associated prediction errors) are



a) Repeated audio prediction with 10 state updates per timestep and 5 updates of the sequence prior.



b) Repeated audio prediction with 15 state updates per timestep and 25 updates of the sequence prior.

Figure 35: a) Repeated prediction of a constant sine wave with single layer (left) and hierarchical PCN with two layers (right). The hierarchical model learns a top-down state prior for the sequence, while the single layer model has only local context. When convergence in the lowest layer is not guaranteed, such as with too few gradient descent steps or with inappropriate initialisation of precision, only the hierarchical model correctly tracks the incoming signal. b) With increased gradient steps for state inference in the lowest layer both single-layer and hierarchical PCN eventually show accurate posterior predictions (green). Predictions from the state prior (blue) improve only for the hierarchical model.

largely restricted to the start of the sequence, as visible in Fig. 35 b). This indicates that minimizing prediction error can be solved through online inference in independent trials as well as through the more gradual process of weights learning when information between trials is carried over. As noticeable in both Fig. 35 a) and b), the learning dynamic of the hierarchical model is significantly more dynamic, since the weighting of the top-down state prior is slightly adapted at each timestep.

The posterior predictions, indicated in Fig. 35 with green lines, show that the lowest PCN layer does not directly adapt to the top-down prior, but needs some time to tune the remaining weights to this additional source of information. When the top-down prior is correctly integrated, however, the hierarchical model quickly improves over the single layer model, especially with parameter initialisation that prevents full convergence of prediction errors in the lowest layer.

6.4.3 *Replicating filter transfer functions*

We tested the possibility to simulate a Butterworth low-pass (LP) filter, which is widely in various DSP applications. Figure 36 shows input and output audio signals to the targeted LP filter and the corresponding in and outputs of a PCN. We test PCNs with single and two layers on a constantly ascending sine wave tone superimposed on constant white noise. Both PCN variants are able to replicate the desired transfer function of the LP filter and show the desired high frequency content removal.

6.5 DISCUSSION

We presented a gradient-based PC model for audio analysis and synthesis. The hierarchical model targets biological plausibility through locally informed updates while still being efficient and accurate enough to replicate classical DSP tasks like filtering and beat tracking. We reviewed the similarities between the autoregressive state-space models underlying predictive coding, IIR filters, recurrent neural networks, and Kalman filtering. From a modelling perspective, the discussed architecture is to trained to integrate top-down predictions by using an additional set of weights in the transition function. As a result, the hidden state dynamics explicitly depend on the top-down prior as a separate source of information. This can be interpreted as a sort of control input (i.e. resembling the role of u). More elaborate versions of the model could improve upon this by treating top-down signals as (slowly-moving) control states, that control the (potentially more complex) dynamics of the respective layer. In PC models in neuroscience this aspect of top-down control of state dynamics is a core aspect that leads to temporal abstraction between layers [49, 51]. How-

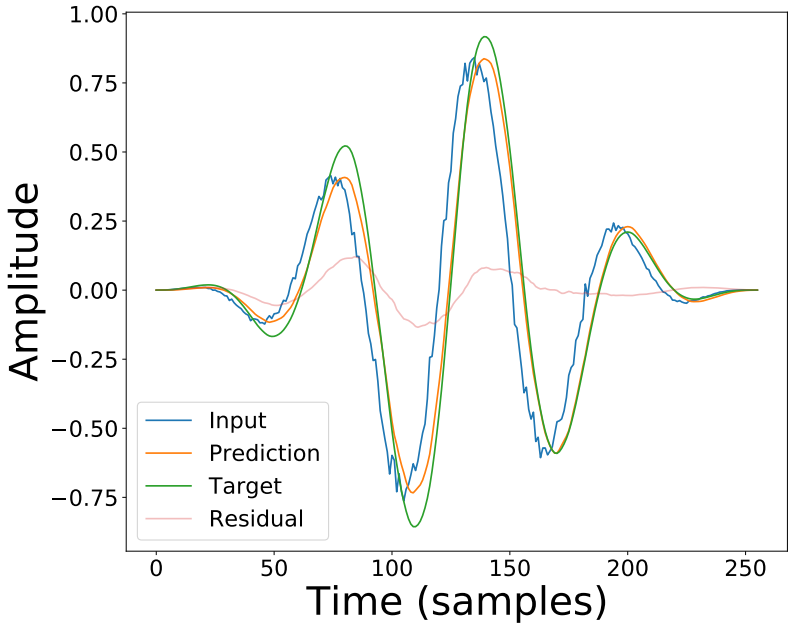


Figure 36: Replicating an order 2 Butterworth LP filter. LP filter and PCN remove high frequency contents and have comparable output magnitudes. As the prediction starts with randomized states and without top-down prior, the prediction error (red) is higher at the sequence start.

ever, this usually implies a separation of states into a set that represents dynamics and into an additional set that controls perturbations of these dynamics. Thus, the complexity of the model is significantly increased. Another promising avenue for more elaborate versions of the investigated model is the inclusion of a variance, i.e. uncertainty, component into the hidden state that is not simply an identity matrix and constant during inference. We will discuss a gradient based PC model that includes cause and hidden state separation as well as variance estimation in the next chapter. It is unclear how estimation of uncertainty would affect performance on signal processing tasks on audio, where high accuracy generally is of high importance, leaving ample room for future exploration. Depending on the chosen focus, the model investigated here provides a basis for future work that could approach more complex audio signal processing applications or aspects of subjectivity in artificial music perception.

7

GENERALIZED PREDICTIVE CODING

The previous chapters have covered the VAE as a simple model of variational free energy optimisation with DNNs, a more elaborate PC inspired DNN architecture and investigated gradient-based PC with local learning rules. Out of these, gradient-based PC networks (simply referred to as PCNs in this chapter) arguably have the largest degree of biological plausibility. PCNs can perform approximate backpropagation of error in supervised learning settings and have been scaled to unsupervised representation learning in deterministic settings [140, 208]. However, it is less clear how PC compares to state-of-the-art architectures, such as VAEs, in unsupervised and probabilistic settings. In this chapter we propose a PCN that is directly inspired by generalized PC (GPC) in neuroscience [49, 51] in the context of automatic differentiation and exact gradient computation. The network parameterizes hierarchical distributions of latent states under the Laplace approximation and maximises model evidence via iterative inference using precision weighted local error signals. Unlike its inspiration it uses multi-layer neural networks with nonlinearities between latent distributions. From a machine learning perspective, this approach offers a model that is consistent with an established process model in neuroscience [51] while allowing the flexibility to include single or multi-layer neural networks within each hierarchical layer. Next to a hierarchical abstraction of static observations, the model covers dynamical predictions with respect to a "dynamical hierarchy" covering the first-order motion of latent states as well as their higher order motion derivatives. In this chapter, we compare a static variant of the proposed model to VAE and Variational Laplace Autoencoder (VLAE) baselines on three different image datasets and find that generalized PC quantitatively shows performance comparable to variational autoencoders trained with exact error backpropagation. We also investigate the possibility of learning spatio-temporal dynamics via static prediction by encoding sequential observations in generalized coordinates of motion, i.e. using a hierarchical-dynamical GPC model.

The content and figures in this chapter is based on the following publication:

> [pub:1] A. Ofner, B. Millidge, and S. Stober. "Generalized Predictive Coding: Bayesian Inference in Static and Dynamic models." In: *4th Shared Visual Representations in Human and Machine Intelligence workshop (SVRHM) at NeurIPS*. 2022.

STRUCTURE OF THE CHAPTER Section 7.1 briefly recapitulates variational inference in PCNs and DNNs and provides an overview on related work. Section 7.2 introduces the proposed generalized PC network in the context of automatic differentiation. The implemented models and datasets chosen for experimentation are discussed in Sections 7.3 and 7.4. Section 7.5 covers the results from the experiments, followed by a discussion in Section 7.6.

7.1 RELATED WORK

7.1.1 *Predictive coding and variational inference*

In the previous chapter, we have focused on a simple PCN that predicts the motion of its hidden states with respect to (discrete) sequential observations and a top-down prediction of the hidden state. PCN models in neuroscience often have a more elaborate structure that is "hierarchical-dynamical". "Static" PCNs are organised hierarchically, where top-down signals from higher layers predict the activity of the layer below and bottom-up signals convey prediction errors. In "hierarchical-dynamical" PC models each layer additionally predicts temporal changes of expected neural activity in the layer below. Given these dynamics, Hebbian updates on weights and activities can be defined that minimize the prediction error at each hierarchical layer of the network.

The weight and activity update dynamics of PCNs can be interpreted as performing variational inference by iteratively refining an inferred distribution over possible causes $p(z|o)$ of observed sensory data o [51, 52, 208]. In variational inference, an approximate distribution $q(z;\lambda)$ is fit to the generally intractable posterior $p_\theta(z \mid o)$ by optimizing the variational free energy

$$F : F_\theta(o;\lambda) = E_{q(z;\lambda)}\left[\ln p_\theta(o,z) - \ln q(z;\lambda)\right] \tag{61}$$

In predictive coding, we define $q(z;\lambda)$ to be a simple diagonal or full-covariance Gaussian distribution with λ as the sufficient parameters, i.e. the mean and covariance. Given the generative model θ (decoder) of a particular hierarchical layer, inference in PC models proceeds by estimating the optimal variational parameters λ^* that maximize model evidence given observed data and current parameterization. Learning of the parameters of the generative model θ can be achieved by performing a gradient descent on $F_\theta(o;\lambda^*)$ with respect to θ which results in Hebbian weight updates. Crucially, learning and inference in PCNs is driven by locally generated predictions and prediction errors. In hierarchical PCNs, the predicted distributions of higher layers foster empirical priors for the next lower layer:

$$p(z, o) = p(o \mid z_1)\, p(z_1 \mid z_2) \ldots p(z_{L-1} \mid z_L) \tag{62}$$

such that a layer's inference model can be interpreted as the next higher layer's generative model.

7.1.2 *Gradient-based predictive coding networks*

Overall, a surprisingly small amount of work has focused on scaling inference and learning of PC networks with local learning rules in the domain of machine learning [139]. This is in contrast to the wealth of computational PC models in neuroscience [14, 49, 51, 172]. Within machine learning applications, first steps have been taken to apply PC models to large datasets, usually focusing on image datasets with relatively low complexity, such as the MNIST dataset [138]. In this context, supervised and unsupervised learning has been addressed [138, 139]. Similar hierarchical PCNs has been implemented in the context of convolutional neural networks [140, 180]. These PCNs are often trained with a methodology similar to training deep neural networks, e.g. by computing weight updates with respect to stochastic mini-batches of data from a larger dataset [157, 225].

Most of aforementioned models perform inference and weight updates assuming that an exact inverse of the forward weights in each hierarchical layer is known, i.e. credit assignment proceeds by computing the *exact* gradient for the activity and weights in each hierarchical layer, using the locally arising prediction error at that particular layer. It has been shown that using approximate (learned) backward weights leads to similar performance [142]. This motivates us to focus on exact gradient computation and resort to automatic differentiation of the (nonlinear) feedforward function. This implies that exact propagation of error gradients, but local to each hierarchical layer of the PCN is compatible with PC. From a deep learning perspective, this means that the predictive distribution between hierarchical layer can be a arbitrarily complex function, e.g. a deep neural network, as long as the prediction errors are computed locally with respect to the parameterized hierarchical latent variable. This fits well with the idea that complex (i.e. "deep" or multi-layer) neural networks could be interpreted in terms of dendritic connectivity in biological neurons [128].

Recently, learning in PCNs have been extended to include amortized inference of inferred latent states [208]. The central idea is that PCNs might have an additional pathway enabling "bottom-up" amortized inference via weights that are learned with respect to the entire dataset. This potentially removes the need for costly iterative inference. In this chapter, we will take a similar view and treat amortized inference of latent states. However, we do not explicitly introduce ad-

ditional "bottom-up" weights for amortization. Instead, we stick to the conventional hierarchical structure of PCNs and exploit the fact that each hierarchical layer already amortizes the iterative inference of the respective lower layer.

Typically, the inference in PCNs is interpreted deterministically, e.g. using a dirac-delta (point mass) distribution [139]. This simplifies inference in PCNs to inferring the MAP of an encoded mean parameter in each layer and allows to ignore computing the estimated variance (or uncertainty) in the model. Here, we focus on inference using the Laplace approximation and explicitly interpret the PCN as performing variational inference on Gaussian states with respect to a directly optimised mean and a covariance parameter that equals the Hessian of the joint probability of the model.

Even less work has been done with respect to dynamical PCNs [138, 140]. Existing studies focusing gradient based PCNs in machine learning focused on simple models that perform single step autoregressive predictions, drawing connections to recurrent neural networks and the Kalman filter [140, 141]. Such simple dynamical PCNs that perform inference with respect to observations at a future, discrete timestep have been treated in the previous chapter of this thesis. A possible explanation for the lack of gradient-based PCN models in machine learning is that elaborate dynamical PC models in neuroscience often modelled in continuous-time, or as a hybrid of continuous and discrete time, e.g. using a Taylor series expansion of discrete model inputs [51]. Here, we will take the latter, hybrid, approach and show how scaled up dynamical PCNs can be trained on conventional, i.e. discrete, sequential datasets in machine learning, assuming that a high sampling rate is available.

7.1.3 *Variational autoencoders with iterative inference*

The VAE is a highly influential class of deep neural networks that performs amortized inference of λ using an inference model ϕ (encoder) [104]. The inference model in VAEs learns to predict the approximate posterior $q_\phi(z \mid o)$ by learning the parameters ϕ of the variational mapping over a dataset. In contrast to the Hebbian updates in PCNs, VAEs are trained using exact backpropagation of error through the entire model [177]. In VAEs, backpropagation of errors through samples from the random latent variable $\tilde{z} \sim q_\phi(z \mid o)$ is solved by resorting to the "reparameterization" trick that involves expressing the sampled distribution as a *differentiable* function $g_\phi(\epsilon, o)$ with respect to an additionally introduced noise variable ϵ. A typical choice for the prior distribution in VAEs is a normal distribution with diagonal covariance

$$z \sim q_\phi(z \mid o) = \mathcal{N}\left(z; \mu, \sigma^2 I\right) \tag{63}$$

such that

$$\tilde{z} = \mu + \sigma \odot \epsilon \quad \text{with} \quad \epsilon \sim \mathcal{N}(0, I) \tag{64}$$

More recently, the notion of iterative inference has been adapted for VAEs to improve the posterior distribution [128, 162]. While VAEs (especially those with iterative updates) and static PCNs have striking similarities in terms of architecture and optimisation scheme, there is still a lack of quantitative comparisons in the literature [128]. Similarly, various deep recurrent models for predicting sequential stimuli have been developed, but lack exhaustive comparison to dynamical PCNs [20, 80, 176].

7.1.4 *Generalized predictive coding*

Generalized PC (GPC) describes an influential class of PCNs that covers static and dynamic models in combination with generalized coordinates of state motion and the Laplace approximation [49, 51]. Static GPC networks infer the conditional mean and covariance of cause states v and hidden states x. Each hierarchical layers predicts the expected activity in the next lower layer using non-linear function g: $y = g(x, v, \theta) + z$. Dynamical GPC networks additionally predict the motion of hidden states $\dot{x} = f(x, v, \theta) + w$ using a non-linear transition function f. When hidden states are ignored, the resulting model is static, i.e. lacks dynamical predictions. z and w denote observation noise and transition noise respectively. While cause states are predicted hierarchically, hidden states are usually not observed by higher hierarchical layers.

Under the assumption of local linearity, GPC uses states in generalized coordinates of motion $\tilde{y} = [y, y', y'', \ldots]^T$, where y' denotes the temporal derivative at y. Similarly, for cause and hidden states:

$$\begin{aligned} y &= g(x, v) + z & x' &= f(x, v) + w \\ y' &= g_x x' + g_v v' + z' & x'' &= f_x x' + f_v v' + w' \end{aligned}$$

Using Gaussian priors $p(z) = \mathcal{N}(z; \bar{\mu}, \Sigma)$, GPC infers posterior distributions of the causes $p(\tilde{x} \mid \tilde{v}) = N\left(D\tilde{x} : \tilde{f}, \tilde{\Sigma}^z\right)$ and the hidden states $p(\tilde{y} \mid \tilde{x}, \tilde{v}) = N\left(\tilde{y} : \tilde{g}, \tilde{\Sigma}^z\right)$. Here, D denotes a derivative operator that replaces each order of state motion with the next higher order: $x \leftarrow x', x' \leftarrow x'', \ldots$.

While conditional mean parameters μ are encoded explicitly, the covariance Σ is encoded implicitly as a function of the mean using the Laplace approximation (LA). Under the LA, the covariance is determined by the local curvature of $-\log p_\theta(y, v, x)$ at the inferred mode of $p_\theta(v, x \mid o)$. Figure 37 shows dynamical and static GPC in comparison to a VAE.

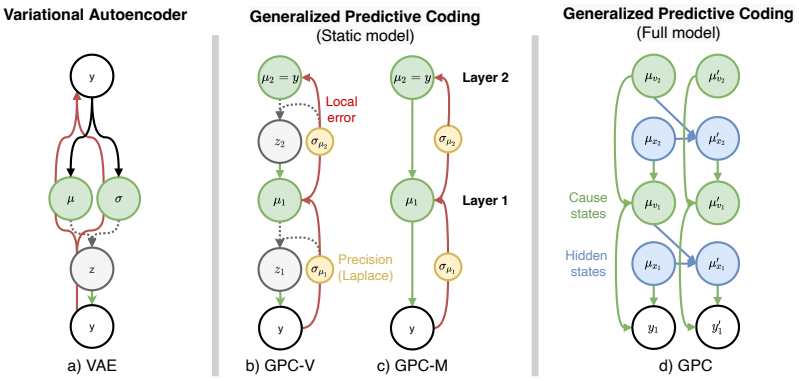


Figure 37: Variational autoencoders (a) encode mean and variance of their latent distribution. Error signals are propagated through the entire network via the backpropagation algorithm. Generalized PC (b-d) propagates local errors and encodes only the mean under the Laplace approximation. The variance is a function of the mean and can be explicitly sampled (b) or appears only as error weighting terms (c).

GPC proceeds by expressing the free energy F of each hierarchical layer l as a function of precision weighted prediction errors $\xi^{(l,o)} = \Sigma^{(l,o)^{-1}} \epsilon^{(l,o)}$ for outgoing predictions and for top-down predictions $\xi^{(l,v)} = \Sigma^{(l,v)^{-1}} \epsilon^{(l,v)}$ from the next higher layer. Here, precision is the inverse of the covariance Σ. Depending on the chosen prior, the distance between prior and posterior distribution is measured by $\xi^{(l,p)} = \Sigma^{(l,p)^{-1}} \epsilon^{(l,p)}$. Here, $\epsilon^p = \mu - \bar{\mu} = \mu - 0$, is the prediction error between posterior and prior and $\epsilon^{(l)v} = \mu^{(l)} - g(\mu^{(l+1)})$ is the hierarchical prediction error between layers (or the sensory prediction error at the lowest layer). For dynamical models, the generalized predictions $\tilde{y}$ result in generalized errors $\tilde{\epsilon} = \tilde{y} - \tilde{o} = [\epsilon, \epsilon', \epsilon'', \ldots]^T$. Inference in each layer is done via gradient descent on $\xi = \Sigma^{-1} \epsilon$ for cause states [51]:

$$\dot{\tilde{\mu}}^{(l)v} = \tilde{\mu}^{(l)v} - \tilde{\varepsilon}_v^{(l)T} \xi^{(l)} - \xi^{(l+1)v}. \tag{65}$$

Within each hierarchical layer, the motion of hidden states is inferred as:

$$\dot{\tilde{\mu}}^{(l)x} = D\tilde{\mu}^{(l)x} - \tilde{\varepsilon}_x^{(l)T} \xi^{(l)}. \tag{66}$$

7.2 GPC WITH AUTOMATIC DIFFERENTIATION

Here we are interested in modelling the GPC model introduced in Section 7.1.4 in the context of multi-layer neural networks and automatic differentiation. We first describe the structure and inference mecha-

nism for a static model. After that, we introduce the full GPC model that additionally models hidden state dynamics over time.

Expanding on related work on hierarchical PCNs, we start with a static PCN that covers L hierarchical layers and computes predictions about the activity of latent (cause) states $v^{(l)}$ on lower layers using a nonlinear function $g(v^{(l+1)})$ that parameterizes its generative network. We allow the nonlinear function to be parameterized by a multilayer NN and employ exact backpropagation of errors to update the parameters of a particular hierarchical layer, with respect to the locally computed prediction error. [1] The locally computed prediction error is composed of three terms, which will cover in the next section.

7.2.1 *Inference and learning with prediction errors*

For a hierarchical layer l, the variational free energy F decomposes into an accuracy term, that measures the quality of the outgoing prediction $g^{(l)}(\mu^{(l)})$, and a complexity term between top down predicted state $g^{(l+1)}(\mu^{(l+1)})$ and inferred state $\mu^{(l)}$. Since we want to compare the GPC model to variational autoencoders, we regularize by the distance between a standard normal prior distribution and the inferred posterior distribution. Under the Laplace approximation, this simplifies to the divergence from zero mean.

$$F(o, q^{(l)}, \hat{q}^{(l)}) = E_q[\log p(o \mid z)] - KL(q^{(l)}(z) \| \hat{q}^{(l)}(z)) - KL(q^{(l)}(z) \| p(z)) \tag{67}$$

Hidden hierarchical layers predict the mean of activity of the layer below, i.e. observations o are replaced by the sufficient parameters μ^{l-1} and Σ^{l-1}. We can also express this lower bound with respect to the optimal inferred posterior distribution $q^{*(l)}(z)$ that is inferred during inference:

$$F(o, q^{(l)}, \hat{q}^{(l)}) = E_q[\log p(o \mid z)] - KL(q^{*(l)}(z) \| \hat{q}^{(l)}(z)) \tag{68}$$

In our GPC model, the optimal inferred posterior distribution $q^{*(l)}(z)$ is inferred using an approximation of the full KL divergences that rests on precision weighted errors that are simple to compute locally for each layer:

$$\begin{aligned} F(o, q^{(l)}, \hat{q}^{(l)}) &= E_q[\log p(o \mid z)] - KL(q^{*(l)}(z) \| \hat{q}^{(l)}(z)) \\ &\approx -\epsilon^{(l,o)} \xi^{(l,o)} - \epsilon^{(l,v)} \xi^{(l,v)} \end{aligned} \tag{69}$$

1 In order to avoid confusion, we will explicitly refer to hierarchical layers of the PCN (i.e. referring to a latent variable and associated generative network) as "hierarchical layers" and refer to layers of the generative network simply as "layers".

Where the corresponding weighted prediction errors ξ are computed as:

$$\begin{aligned}
\xi^{(l,p)} &= \Sigma^{(l,p)^{-1}} \epsilon^{(l,p)}, & \epsilon^{(l,p)} &= (\mu^{(l)} - 0) \\
\xi^{(l,v)} &= \Sigma^{(l,v)^{-1}} \epsilon^{(l,v)}, & \epsilon^{(l,v)} &= (\mu^{(l)} - g^{(l+1)}(\mu^{(l+1)})) \\
\xi^{(l,o)} &= \Sigma^{(l,o)^{-1}} \epsilon^{(l,o)}, & \epsilon^{(l,o)} &= (g^{(l)}(\mu^{(l)}) - o)
\end{aligned} \tag{70}$$

To make use of amortized inference, at the start of iterative inference, the inferred posterior is initialized with its top-down prediction $\mu_0 = \hat{\mu}$. During inference the optimal posterior distribution $q^{*(l)}(z)$ with respect to distance from the prior and decoder accuracy is then computed using a simple gradient descent or using Gauss-Newton updates on $\epsilon^{(l,p)}$ and $\epsilon^{(l,o)}$. After inference, the covariance parameters of the top-down predicted distribution $q(\hat{z})$ and the inferred posterior distribution $q^*(z)$ are inferred following the routine described in Park, Kim, and Kim [162] using $\Sigma^{-1} = -\nabla_z^2 \log p_\theta(o, z)\big|_{z=\mu}$, which can be efficiently computed for ReLU activations. We then compute weights updates using the Adam optimiser by replacing the full KL divergence in Formula 3 with the precision weighted error $\epsilon^{(l,v)}\xi^{(l,v)}$. For model evaluation and comparison to the baselines, we use the full analytical KL divergence. We found that training the model using the full KL terms leads to similar results, although with increased numerical instability when latent vectors have large dimensions or when the amount and size of inference steps is insufficient.

7.2.2 *Laplace approximation with ReLU nonlinearity*

Inspired by the work of Park, Kim, and Kim [162], we employ ReLU non-linearity for the input and hidden layer of each hierarchical layer's decoder network, followed by a linear output layer. For decoder network weights W and Jacobian W_z with respect to latent states $z = (v)$ (or $z = (v, x)$ for a hierarchical-dynamical models, as discussed in the next section) ReLU non-linearity allows to efficiently compute the precision of inferred posterior states

$$\Sigma^{-1} = -\nabla_z^2 \log p_\theta(o, z)\big|_{z=\mu} = {W_z}^T W_z + I\,|_{z=\mu} \tag{71}$$

by computing binary activation masks $\mathrm{ReLU}(Wz) = O(Wz)$ during the decoder's forward pass and recursively multiplying with the decoder's weights [162]. Under a local linearity assumption, we can use the approximation ${W_z}^T W_z$ of the generative network's Hessian matrix for precision weighted state updates

$$\dot{\mu} = \left(W_z^T W_z + I\right)^{-1} W_z^T \varepsilon\,|_{z=\mu} \tag{72}$$

during inference. After a fixed amount of inference steps towards the posterior mode the approximate posterior distribution is

$$q(z \mid o) = \mathcal{N}(\mu, \Sigma) \tag{73}$$

using the Laplace approximation [162]. This distribution is then used to compute gradients for the weights. We perform exact error propagation to the weights strictly locally within each hierarchical layer using automatic differentiation in PyTorch [78, 164]. This is in contrast to a backpropagation pass over all parameters, such as in VAEs, where the decoder's error directly drive updates of encoder parameters.

7.2.3 *Hierarchical-dynamical GPC*

Dynamical GPC models are trained like static GPC via iterative inference. To create a hierarchical-dynamical model GPC, we introduce additional hidden states x, such that each hierarchical layer contains latent states $z = (v, x)$. The resulting model predicts the mean of cause states $z^{(l-1)} = g(z^{(l)})$ of lower layers using the hierarchical generative network g as described in the previous sections. Here, we do not allow skip connections between hierarchical layers, i.e. only the hidden states x are used for outgoing hierarchical predictions.

In the hierarchical-dynamical model, these hierarchical predictions are computed in generalized coordinates: This means that, in addition to decoding the states $y = g(z)$ per se, their explicitly represented state motion $\tilde{y} = g(\tilde{z})$ is also decoded and compared with the data $\tilde{o}$. During decoding the states $y = g(z)$, the Jacobian W_z at the currently inferred mode is computed by masking the decoder network. All other orders of state motion $y' = g_z z', y'' = g_z z'', ...$ are decoded through this masked decoder network W_z.

So far, we have covered hierarchical predictions in generalized coordinates based on a separation between cause and hidden states. To actually make the model dynamical, it needs to be able to predict the motion of hidden states in generalized coordinates $D\tilde{x} = \tilde{f}(\tilde{v}, \tilde{x})$. During the prediction of hidden state motion $x' = f(z)$ the Jacobian of the transition network f_z is computed. This Jacobian is reused for all higher order hidden state motion predictions $z'' = f_z(z')$, ..., $z^N = f_z(z^{N-1})$. We interpret the Taylor series expansion underlying the forward and inverse embedding of sequential data as a convolution operation along the temporal axis, which can efficiently be computed using convolutional kernels. Figure 38 displays the dynamical connectivity in the hierarchical GPC layer.

We now arrived at a fully constructed hierarchical-dynamical model that mirrors the structure of generalized predictive coding in neuroscience [49, 51]. Conceptually, the resulting structure offers a

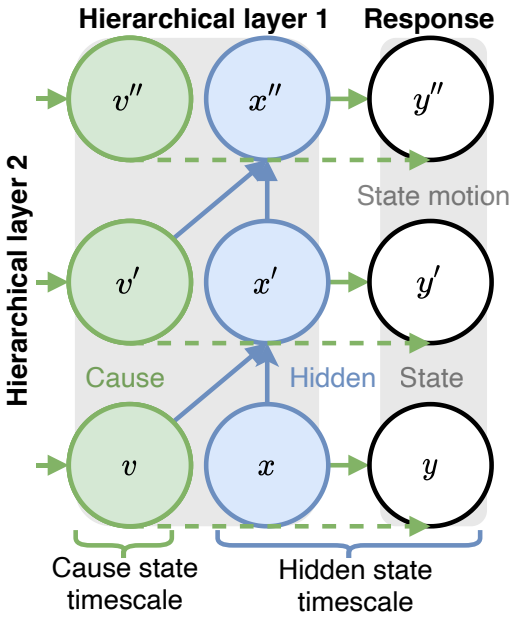


Figure 38: Hierarchical predictions (green) express expectations about causes (or data) in the next lower layer. Dynamical predictions (blue) predict higher orders of state motion. Dotted connections indicate optional skip connections between causes and the layer's hierarchical response (not used here).

large degree of abstraction: At each timestep, each hierarchical layer predicts the cause state and N orders of *instantaneous* cause state motion of the respective lower hierarchical layer. These lower layer's cause states, in turn, perturb the dynamics of the lower hierarchical layer's hidden states. This means that each hierarchical layer abstracts away from the dynamics of the next lower layer, resulting in different time-scales. Expressing the model in generalized coordinates allows to predict all orders of motion, across all hierarchical layers in parallel - rendering dynamical predictions entirely static in time. In the next section, we discuss a proposed method that turns discrete sequences of data in a dataset into a generalized representation, such as required by the model.

7.2.3.1 *Generalized coordinates from discrete sequential data*

We compute temporal embeddings of observations according to a Taylor expansion of form

$$f(x \pm dx) = f(x) \pm dx f'(x) + \frac{dx^2}{2!} f''(x) \pm \frac{dx^3}{3!} f'''(x) + \frac{dx^4}{4!} f''''(x) \pm \dots \tag{74}$$

for points $x \pm dx$ around a point x assuming a fixed step size e.g. $dx = 1$. Since we observe discrete samples $[o_1, ...o_n]$ we approximate

the instantaneous derivatives $f', f'', ...$ up to desired order using a central finite difference operator $\delta_d x^n[f](x)$. We interpret the resulting differencing coefficients as convolutional kernels, which can be applied to any sequential data with sufficiently high sampling rate either online or during preprocessing. Mapping back from the network's states to sequential data can easily be done using the inverse kernel. Figure 39 shows examples for forward and inverse embedding kernels.

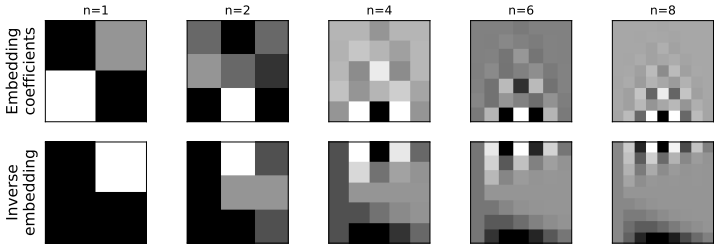


Figure 39: Forward and inverse embedding kernels for five different embedding orders. Embedding coefficients (top row) are applied to the temporal axis of each observed unit and project from discrete samples around an expansion point, the centered sample, to orders of unit motion in generalized coordinates. The reciprocal mapping is achieved with the inverse embedding coefficients (lower row) that map from orders of state motion to discrete samples.

7.2.3.2 *Multiple shooting*

Following related work on multiple shooting (MS) based training of Neural Ordinary Differential Equations (ODEs) we train the model by splitting discrete sequences into multiple segments, which are optimised in parallel [34, 209]. We sample discrete sequences $[o_{t_1}, ...o_{t_n}]$ of length n at m shooting points $[o_{\tau_1}, ...o_{\tau_m}]$ which are then embedded into generalized coordinates. For b sequences sampled at m point, the network is trained with a batch size of $b * m$. In practise, this means that we can omit the term $D\tilde{\mu}^{(1)x}$ for the dynamical predictions, which would be required for sequential filtering.

Figure 40 shows an example with two discrete sequences. Crucially, while hidden states are inferred for each shooting point o_{τ_i} individually, the prediction errors for cause states are averaged over all m samples from a sequence. The network thus learns to represent the instantaneous motion in the generalized observation $\tilde{o}_i$ at each each shooting point τ_i with hidden states $\tilde{x}$ while a sequence-wise cause $\tilde{v}$ controls (or "perturbs") the hidden state dynamics. Multiple shooting provides and efficient way to learn weight parameters and cause states with the model. Another possible ways to address learning in the model is to filter generalized observations sequentially, i.e. one ob-

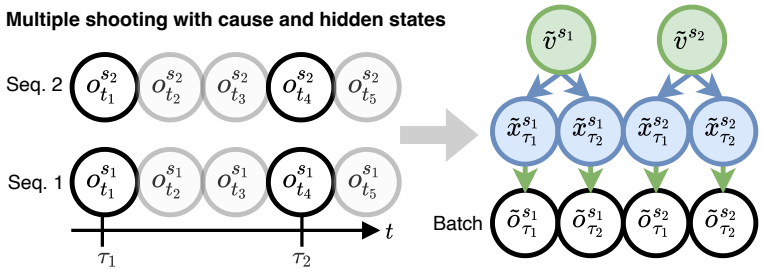


Figure 40: Example for multiple shooting with the proposed PC network. Multiple shooting allows make inference with respect to multiple locations, or shooting points τ in parallel. Discrete samples around each each shooting point τ_i are projected to generalized observations using the embedding kernels shown in 39. These generalized observations $\tilde{o}_i$ are arranged over the batch dimension. Then, hidden states $\tilde{x}$ are inferred for each $\tilde{o}_i$. For each sequence only a single cause state $\tilde{v}$ is inferred.

servation at a time. In such sequential Bayesian filtering, cause states and weights are optimized with respect to the time integral of the variational free-energy, the free action.

7.3 IMPLEMENTED MODELS AND BASELINES

7.3.1 *VAE and VLAE baseline*

We use the conventional VAE architecture with fully factorized normal distribution, reparameterization of the latent distribution and trained via backpropagation of error [104, 162]. It is trained using a single sample from the latent distribution as input to the decoder and the regularization with a standard normal. The VLAE is a variant of the VAE with iterative mode seeking that defines a full-covariance Gaussian posterior at the mode using the Laplace approximation [162]. The VLAE uses a single sample from the latent distribution at the inferred mode as input to the decoder. Unlike the cited model, we do not use a decaying learning rate for mode seeking. For the VAE and VLAE models, the encoder and decoder consist of two ReLU activated layers with 256 hidden units and parameterize 16 latent units.

7.3.2 *Static GPC model*

We implement a static GPC with two hierarchical layers and fix the mean of the second layer's latents to the data. In this setup, the output of the second hierarchical layer provides empirical priors via amortized inference on the first hierarchical layer's cause states $p(v_1)$

[2]. The resulting architecture resembles that of an autoencoder as it uses the second layer's generative model as the first layer's inference model. Predictions between cause states are parameterized by a dense neural network with three layers. All generative networks have 256 hidden units and 16 latent units (causes). Unlike VAE and VLAE, the PC models do not use biases. In contrast to the VAE baseline, inference and learning in the presented network (GPC-M) does not involve a sampling step. For comparison we also implement a network (GPC-V) that is trained using stochastic updates with a single sample from the posterior distribution using the reparameterization trick, as is standard procedure in VAEs.

7.3.3 *Dynamical GPC model*

We also implement two variants of dynamical GPC models to test amortized inference and dynamical inference of causes respectively: A simplified model with two hierarchical layers without cause states that predicts hidden states top-down. Again, the data serves as fixed input to the second hierarchical layer. The second model consists of a single hierarchical layer that infers cause and hidden states with associated dynamics using multiple shooting. We use 16 units for cause and hidden states in all models. We use multiple shooting with $b = 4, m = 8$ only for the dynamical GPC model that infers causes and use $b = 64, m = 1$ otherwise. We use generalized coordinates of order $N = 2$ for the simplified and $N = 3$ for the full GPC model.

7.4 DATASETS

We employ three popular datasets for unsupervised learning on images: The MNIST dataset (Creative Commons Attribution-Share Alike 3.0 license), Fashion MNIST (MIT license) and OMNIGLOT (MIT License). Evaluation of the dynamical PC model is based on the Disentanglement testing Sprites dataset (Apache License 2.0). MNIST and FashionMNIST contain 60000 train and 10000 test images (with 28×28 pixels) while the OMNIGLOT dataset contains 24345 train and 8070 test images (also with 28×28 pixels). To generate discrete video sequences with high temporal resolution for the dynamical GPC model we use a customized version of the dSprites dataset [133]. We generate 128000 random samples from the original dataset and apply Gaussian blur along both spatial axes with kernel size 3 and Standard deviation of 10 before applying normalization. We restrict x and y positions to six values respectively, such that all sprites appear within a center crop of 32 x 32 pixels. Starting with the noisy version of the sprite, we apply a single direction of rotation (counterclockwise) by

2 Note that the cause state of the second hierarchical layer is fixed, i.e. iterative inference is restricted to the first hierarchical layer.

rotating the sprite by a single degree. All remaining aspects, such as shape, size, horizontal position and vertical position stay constant. The Gaussian noise was applied only to the first frame of each sequence. We then projected the resulting discrete video sequences into generalized coordinates using the embedding kernels discussed in Section 7.2.3.1.

7.5 EXPERIMENTS

7.5.1 *Static predictive coding*

We train and evaluate models with varying amounts of inference steps on MNIST [115], FashionMNIST [229] and OMNIGLOT [114]. Unlike the VLAE baseline, we do not initialise the decoder output variance based on dataset statistics. Instead we add noise from a standard normal distribution and apply a logit transformation for all datasets. Table 9 shows test results on all datasets for 3 and 6 iterative inference steps using the conventional train and test splits. Listed are mean and standard deviation across 10 runs. We trained for $1e+4$ steps with the ADAM optimiser at a learning rate of 1e-2 [103] and inference learning rate of 0.5, the default setting of the VLAE baseline [162]. In almost all configurations, GPC-S and GPC-M slightly outperform the VAE, while the VLAE model consistently outperforms both PCNs. This indicates that PCNs, despite lacking exact error signals for the inference network learn a generative model that is comparable to the VAE. The GPC-M model without explicit sampling consistently outperformed the sample-based GPC-S model, except for one configuration on OMNIGLOT. In terms of divergence from the prior, the GPC models consistently showed posterior complexity that is comparable to, but slightly higher than, VAE complexity. For all tested models, increasing the number of inference steps is beneficial only for low numbers of steps. We found that reducing the inference learning rate or adding a decay term can improve stability, but did not include it in our experiments.

Table 11 in the Appendix shows the posterior complexity of models trained on the static prediction task for MNIST, OMNIGLOT and Fashion MNIST in terms of mean and standard deviation over ten runs. The PC models GPC-S and GPC-M show complexity that is comparable to, but slightly higher than the complexity of the baseline VAE. The VLAE shows complexity values that are smaller than the baseline VAE in four out of the 6 tested configurations. For VLAE and GPC models, increasing the amount of inference steps from 3 to 6 slightly increases the complexity of encoded states.

	MNIST	OMNIGLOT	fMNIST
VAE	901.2±1.4	1019.4±1.2	881.3±0.3
GPC-S (3)	892.9±1.2	1001.1±0.8	882.0±1.5
GPC-M (3)	892.9±1.0	1002.2±0.8	880.0±0.4
VLAE (3)	**881.4±1.2**	**989.3±1.0**	**870.1±0.4**
GPC-S (6)	896.4±1.1	1004.7±0.8	883.0±1.0
GPC-M (6)	894.7±0.8	1003.0±1.4	878.4±0.4
VLAE (6)	**877.4±1.4**	**983.1±3.8**	**869.3±0.2**

Table 9: Negative evidence lower bound (test set)

7.5.2 *Dynamical predictive coding*

To assess the capabilities of dynamical PCNs we train a dynamical GPC model on a variant of the Disentanglement testing sprites dataset [134]. Most conventional video benchmarks have relatively low sampling rates, where the local linearity assumption does not hold. We generate high resolution videos for a single direction of rotation (counterclockwise) and use random, but constant, values for the remaining latent factors. We applied Gaussian blur to all images and cropped the videos to 32x32 pixel resolution, making sure that no sprites appear outside the area.

	GPC-all	GPC-L1
MSE	**0.432±0.124**	0.476±0.204
Layer 1	**0.768±0.257**	0.779±0.422
Layer 2	**0.097±0.013**	0.173±0.031

Table 10: Accuracy of the dynamical model on the rotating dSprites dataset. Variant GPC-all infers prediction error from both dynamical layers. Variant GPC-L1 only infers errors in the lowest dynamical layer. Shown are mean and standard deviation over 10 runs.)

Table 10 shows the MSE over 3e+4 updates using two different variants of the simplified dynamical model: The GPC-all model was trained using the prediction error from both dynamical layers, while GPC-L1 only considers the error in the lowest dynamical layer. Both models smoothly predict the constant rotation across latent factors. GPC-all shows improved MSE in terms of total and per-layer prediction. This indicates that including higher-order dynamical predictions errors propagated through the network's Jacobian indeed improves accuracy. We found that GPC-L1 reacts poorly to increased latent dimensionality and stops predicting any meaningful state motion when

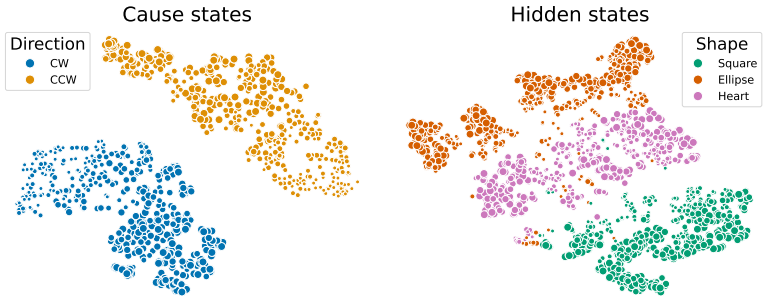


Figure 41: t-SNE projection of cause and hidden states after unsupervised learning. Hidden states encode spatial aspects, such as shape while cause states encode hidden state motion and cluster into rotation directions. Marker sizes indicate the scale of observed sprites.

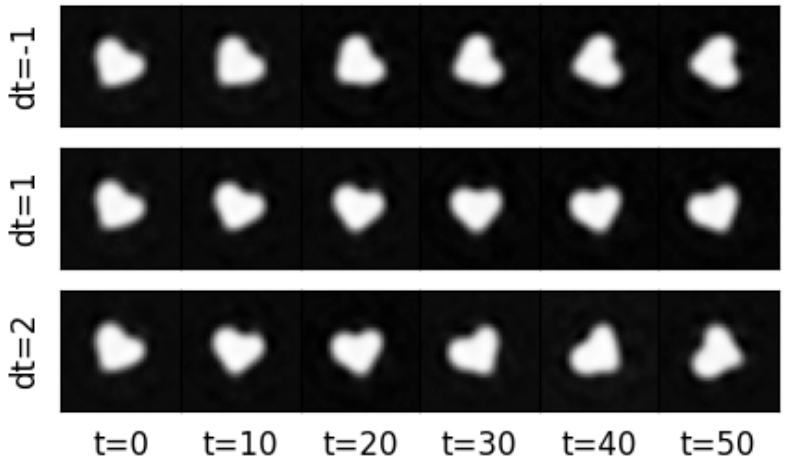


Figure 42: Dynamical prediction with learned causes over three different step sizes dt. Shown is every tenth of 50 steps.

32 or more latent units were used. In contrast, GPC-all showed meaningful transitions for larger embeddings.

7.5.3 *Simultaneously inferring cause and hidden states*

We found that training a dynamical model that infers causes and hidden states simultaneously on the rotating dSprites dataset lead to a clear clustering of causes into the two directions of motion for the inferred cause states, as shown in Figure 41. The hidden states capture spatial aspects, such as sprite shape, which change in dependency of the inferred cause. After training the network and freezing weights, new generalized observations $\tilde{o}$ can be encoded via iterative inference.

The inferred stated can then be used for dynamical prediction of future timesteps, by applying $x' = f(z), x'' = f_z(z'), ...$ Figure 42 shows a typical extrapolation for up to 50 with different step sizes dt scaling the predicted hidden state motion $x' = f(x, v) * dt$. Changing the applied step size allows to increase and decrease the speed of motion forward and backward in time. Figure 43 shows the discrete observations and their temporal embeddings. Additionally shown are the model's generalized prediction $\tilde{y}$ from hidden states as well as the resulting inverse embedding to a discrete sequence.

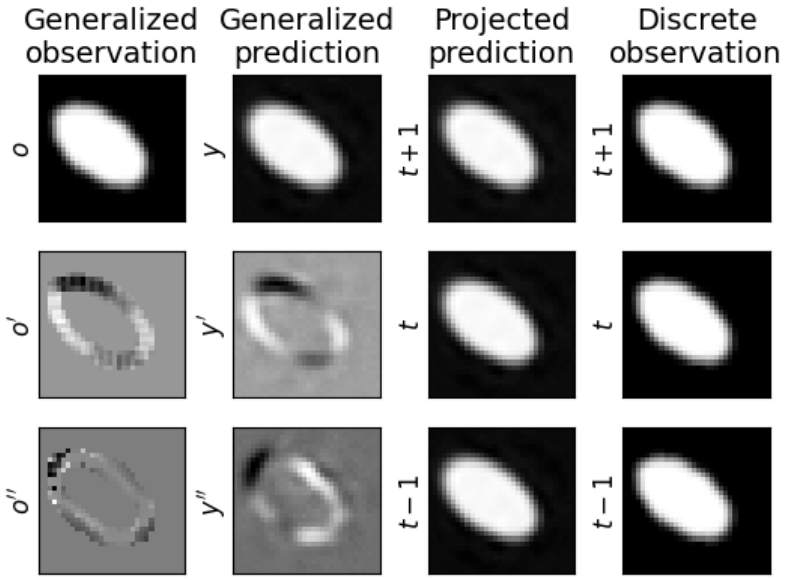


Figure 43: Discrete video frames (right) are fed to the model as generalized observations (left). The generalized sensory prediction of the network can be projected back to discrete sequences (center).

7.5.4 *Gradient descent and Gauss-Newton updates*

We compared Gauss-Newton updates during iterative inference with simple gradient descent steps that do not consider precision weighting. The gradient descent based updates perform well when an adequate learning rate is chosen. Then, in in many cases they outperform VAE baseline in terms of model evidence. As visible in Figure 44, the gradient based updates are much more sensitive to the inference learning rate. We found that values around 0.001 work best, while higher rates lead to degraded performance. In contrast, the Gauss-Newton based updates consider the precision at the currently inferred mode, leading to more stable updates that are adaptively weighted and are less sensitive to the inference learning rate.

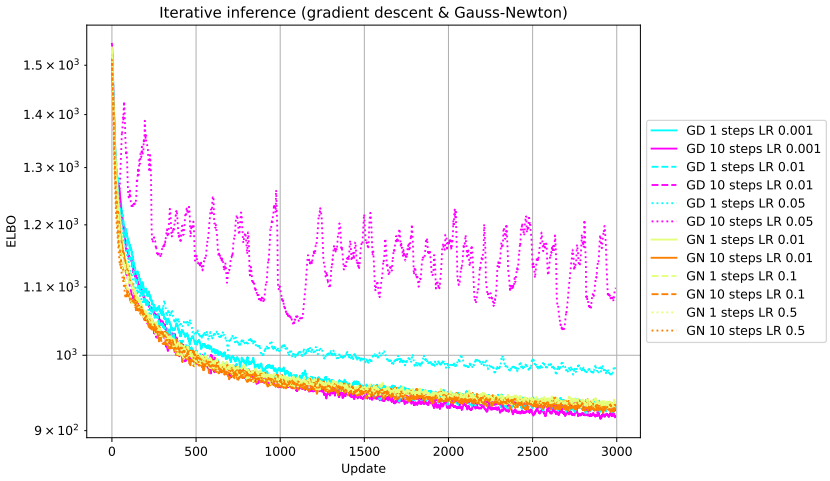


Figure 44: Comparison of static PC with gradient descent and Gauss-Newton updates during iterative inference of the optimal posterior distribution. Shown are the first 3000 weight updates on the MNIST dataset.

7.6 DISCUSSION

We presented a generalized PC network that uses Hebbian updates and the Laplace approximation with nonlinear neural networks to infer posterior distributions. We have shown that the model performs comparably to VAEs trained with exact error backpropagation. We extended the model to cover dynamical predictions of simple video sequences and demonstrated the possibility to learn dynamics using generalized coordinates of motion.

Datasets in the machine learning domain often have relatively low sampling rate and the local linearity assumption does not generally hold. In contrast, physical and biological data such as processed by the brain do not suffer from low sampling rates, but are locally smooth. We found that in many cases, GPC still learns meaningful dynamics on datasets with lower sampling rate. In this chapter, however, we focused on synthetically generated, high resolution data.

Important steps for future work could be to use convolutional neural networks or a comparison to related dynamical models, such as Neural ODEs or RNNs [176, 209]. In this chapter we focused on a model that explicitly represents orders of state motion with a simple inference scheme that allows a direct mapping to canonical microcircuits in the brain [11, 51]. The underlying idea of representing "moving reference frames" to model of high-level motion has recently been connected to autoregressive normalizing flows in the context of deep neural networks trained with exact backpropagation of errors [129]. These insights provide an interesting avenue for the development of more elaborate models, both with respect to deep neural networks

and biologically plausible models of canonical computation in the brain.

In the presented experiments, we have focused on the rectified linear unit activation function, as it allows to efficiently compute the Jacobian. ReLU activations have found widespread use in the deep learning domain. In contrast to other established nonlinearities in DNNs, such as Sigmoid activations, ReLUs are computationally efficient, e.g. due to the lack of exponentials, and show good convergence properties [111]. The Jacobian computation discussed in this chapter exploits the simplicity of the ReLU activation function in terms of derivative computation, which can efficiently be computed as a forward masking operation. This approach is specific to ReLU activations and for other nonlinearities the computation of the Jacobian might be significantly more expensive computationally, possible preventing the possibility to train the model at all.

In this chapter, we have used the widely established Adam optimiser to optimise the model's weights [103]. Adam approximates second order information using the first and second moments, i.e. the mean and variance, of the model parameter first-order gradients. It is a "first order" method, since it requires only the gradient of the parameters in contrast to methods that require higher order gradients. As such, it ignores the covariance of the parameter gradients and with that, does not compute (approximate) information about the optimal *direction* of the gradient descent steps [131]. This is in contrast to other, generally computationally significantly more expensive methods, such as Natural Gradient descent [131] that steer magnitude and direction of gradient descent steps with (approximations of) the Hessian matrix. Similarly, in gradient-based PC models, weights updates are usually expressed with respect to second order information i.e. based on a full-rank (approximation of) the Hessian matrix of the free energy with respect to the model's weights [49, 56]. Depending on the underlying model, the Hessian matrix might be exactly computed (e.g. when the model is entirely linear) or approximated based on the variance of neural activities the precision of the prediction error [49]. We can thus see the Adam optimiser as a simplified (yet in practice efficient and useful) replacement for such full-rank precision estimation with respect to the weights. Future work could provide more elaborate approaches to weights learning with locally represented error signals.

The proposed implementation makes two simplifications that possibly weaken its biological plausibility: Firstly, the gradients of states and weights are computed exactly using automatic differentiation. This means that we have implicitly assumed the existence of exact duplicates of the forward weights as well as the exact computation of the derivatives of employed nonlinearities. It should be noted that this does not address exact copies of forwards weights over *multi-*

ple hierarchical layers, such as would be required for the backpropagation of error algorithm [177] for the entire PCN. Recent work has suggested that this assumption does not have a major impact on PC as an algorithmic scheme, since it can be addressed e.g. by "learning additional sets of parameters with Hebbian update rules without noticeable harm to learning performance" [142]. Learning additional parameters here refers to parameterizing a set of approximate backwards weights, from the locally computed error to the associated states and parameters of a layer in the PC hierarchy. A second simplification addresses the representation of precision on the inferred states. Here, we compute the precision directly, using the presented Jacobian computation based on ReLU activations. The brain, is assumed to estimate precision by representing prediction error units that infer the covariance matrix using lateral connections between error-units [49]. Currently, it is unclear how this or similar mechanisms in the brain could account for computing the state covariance encoded with the Laplace approximation.

8

CONCLUSION

Recent developments in artificial intelligence have led to increasingly powerful and complex generative models. These usually rely on deep neural networks that are trained on large datasets and use exact back-propagation of errors to the parameters of the model. In many cases, the minimization of Bayesian surprise is a central objective for such deep generative models, like the VAE. While several mechanisms in deep neural networks, such as convolutional layers for spatial inputs, are inspired by biology, many aspects, such as the computation of a global error signal across the entire network are still in contrast to evidence about learning in biological brains. At the same time, the Free Energy Principle in neuroscience explains brain function based on the minimization of Bayesian surprise and has led to a variety of biologically plausible process models, such as hierarchical PC. These process models use iterative and locally informed parameter updates to facilitate learning of an internal generative model of the world. Apart from the shared computational objective, the minimisation of Bayesian surprise, it is still unclear how modelling constrains under the FEP relate to deep generative models in terms of performance and comparability to human brain function.

The work covered in this thesis aims at filling some of these gaps by reviewing, designing and evaluating ANN models under the constrains of the FEP. In this context we pay special attention to process models, in particular hierarchical and dynamical PC, that implement canonical computations that are hypothesized to be present across scales in the human brain. In this context we evaluate unsupervised learning, with a focus on auditory and visual stimuli, as well as possibilities to retrieve sensory processing related information in brain activity using FEP based models. The following sections summarize our contributions, followed by a discussion of limitations and possible future research directions.

8.0.1 *Research contributions*

RELATING VAES TO PROCESS MODELS UNDER THE FREE ENERGY PRINCIPLE

In the context of DNN based models, the VAE is a simple, yet effective and widely used architecture that optimises a bound on its Bayesian surprise, the variational Free Energy, or evidence lower bound (ELBO). VAEs consist of an encoder-decoder structure, where the decoder re-

sembles a generative network that maps from a latent representation to expected data. The encoder network, in turn, performs amortized inference over the latent states, by expressing latent state parameters as a function of sensory data. While the objective and underlying generative model is highly reminiscent of a single layer PC model under the FEP, there is still a lack of studies that explicitly relate VAEs and PCNs as a biologically plausible process model. Chapters 3 and 7 in this work address this gap from two different perspectives: Chapter 3 presents a VAE based architecture for shared representation learning from brain signals and auditory data. In this study, we empirically evaluate the possibility to represent simple auditory concepts, like rhythmical patterns or timbral aspects in a DNN model that optimises Bayesian surprise. This is driven by the hypothesis that the free energy minimisation inherent to VAEs, just like in PCNs, can be seen as a canonical model of computation that is also present in the human brain. Learning a shared representation of audio and brain signals, however, is a complex task. Datasets providing time-aligned audio and EEG data are typically small in size and the recorded EEG is very complex, since it records a high-dimensional mixture of brain signals and noise in the hardware. Nevertheless, we were able to reconstruct rhythmically meaningful, time-aligned reconstructions of simple tone patterns, as recorded in the OpenMIIR dataset [203]. To do so, we resorted to a multi-head VAE architecture, that reconstructs audio and EEG from a shared latent representation. We converted the audio signal to mel spectrogram representation and treated time and frequency dimension as spatial axes. Similarly, we treated the time and channel dimension of the EEG as spatial axes. This allows to employ CNNs in order to efficiently process sequential data in a static setup. While two separate decoders were used to reconstruct audio and EEG, a single encoder network was employed, that maps from EEG signal to the latent representation. Using the same setup in the context of more complex, natural music in the NMED-T dataset still led to meaningful reconstructions, although substantial less precise [123]. Next to the model itself, we also proposed and evaluated a technique inspired by research on ERPs in neuroscience, where time-aligned evoked responses are averaged over trials or subjects. By computing the mean over stimuli or subjects at time-aligned data points, we were able to train smaller, yet performant models. Similarly, averaging the reconstructions of models trained on individual inputs helped during qualitative inspection of the results.

Next to this exploratory approach that investigates shared representation learning with VAEs, chapter 7 explicitly relates the architecture of VAEs to that of a hierarchical PCN with two hierarchical layers. In particular, we suggest that the inferred states of the first hierarchical PCN can directly be related to the latent representation of the VAE when the states of the second hierarchical PCN layer are initialized

with the observed data. Then, just like the VAE, the PCN acts as an autoencoder, where the generative network of the second hierarchical layer acts as the inference network. However, in contrast to the VAE, the inference network in PCNs is not informed by the loss of the decoder network that reconstructs the data. Instead, updates in the PCNs are driven strictly by the local prediction errors, the bottom-up and top-down errors in each layer. Using a selection of image benchmark datasets, we compared VAE and PCN model accuracy and latent state complexity. We find that PCNs perform comparable to VAEs in both metrics, although both are outperformed by VLAEs, which combine iterative mode seeking, similar to PCNs, with the exact error propagation used in VAEs. Nevertheless, these results imply that VAEs and PCNs as a process model under the FEP can directly be related with respect to architecture and performance.

DEEP PREDICTIVE CODING MODELS FOR EEG PROCESSING

Several existing deep learning models have been inspired by predictive coding. A particularly influential model is PredNet, which focuses on spatio-temporal predictions and uses multiple autoregressive layers that predict the prediction error of the respective lower layer [124]. Chapter 4 analyses and reviews PredNet in the context of a challenging action classification dataset [66]. After reviewing the architectural differences between PredNet and hierarchical PC models, we empirically evaluate PredNet in an extensive ablation study, covering different network sizes and hyperparameters. We find that PredNet appears to be tailored to excel at short-term temporal extrapolation and is highly sensitive to hyperparameters such as the temporal resolution of videos. Since directly relating the representations of PredNet to a PC model is not straightforward, we empirically investigated the influence of top-down information by including a classification module at the hierarchically highest layer. We find that, while it is possible to classify actions with the modified PredNet architecture, the top-down signal does not improve the prediction process per se. This empirical evidence stresses that the processing in PredNet is substantially different to hierarchical PCNs.

Based on these insights, we contribute a novel deep predictive coding model in Chapter 5. In contrast to PredNet, our model learns by predicting probabilistic sequences of latent states in a hierarchically organized and autoregressive network and matches the connectivity in hierarchical PCNs. We apply the proposed model to two information retrieval tasks on EEG data: A first set of experiments focuses on unsupervised prediction of audio. After training the PCN on audio, we extract the model's prediction error response on the NMED-T dataset [123]. We then threshold this time-aligned prediction error response in order to find temporal locations that are particularly

surprising to the model. Since NMED-T also provides EEG signal, this procedure allows to investigate the human brain's response time-aligned to the PCN. We found that this method works well in practise and detects clearly recognizable human evoked responses.

Next to transferring a PCN's prediction error response to time-aligned human evoked responses, we also investigated the possibility to apply the proposed model directly to EEG signal prediction in Chapter 5. In order to reduce dataset complexity, we resorted to the ZuCo dataset, which contains EEG recordings of human fixation related responses in a free reading task [82]. In this context, we found that the model allows to learn multi-step predictions of EEG signals without complex preprocessing. Furthermore, we were able to actively correct random temporal shifts applied to the EEG signal after training the network on fixation-aligned inputs. While still exploratory, these results hint at the possibilities to use biologically plausible, yet efficient PCNs in the context of information retrieval from brain signals.

SCALING UP PREDICTIVE CODING MODELS WITH LOCAL LEARNING RULES

Next to implementing and evaluating models in the context of DNNs and exact backpropagation of error signals, we also investigated the possibility to scale up PC models that more strictly adhere to the connectivity and learning rules described by hierarchical and dynamical PC under the Free Energy Principle. In particular, we contributed two gradient-based PCNs, that infer states and learn weights using a gradient descent on the prediction error local to each hierarchical layer. The networks are gradient-based in the sense that they do not use explicit backwards weights to propagate the errors within each hierarchical layer, but use automatic differentiation to do this more implicitly. This allows to implement efficient models in the context of established frameworks for automatic differentiation. The models perform strictly locally informed updates without resorting to error backpropagation through multiple hierarchical layers, or through time.

Chapter 6 presents and evaluates a variant of gradient-based PC focusing on audio prediction. The proposed model is deterministic, predicts raw audio signals at discrete timesteps and includes a top-down input to the state update. We found that, when applied to iterative prediction of simple audio inputs, the model successfully integrates the top-down prediction leading to more accurate reconstructions based on prior knowledge. Furthermore, we were able to apply the model to a beat tracking task. We found that the model outperforms baselines on the NMED-T dataset [123], while delivering worse performance on an established benchmark dataset. In this context, we also reviewed the similarities between single layer dy-

namical PCNs and other popular methods in audio processing, such as trainable IIR filters.

The model discussed in Chapter 6 focuses specifically on dynamical prediction at discrete timesteps, without other core aspects that are covered by generalized PC models in neuroscience [51], such as uncertainty estimation or temporal predictions in generalized coordinates of motion. Generalized coordinates capture the instantaneous change in a model's state (as well as the sensory input) with respect to multiple temporal derivatives. Such generalized representations are usually modelled in continuous time, making it difficult to apply to ANN based models straightforwardly. Chapter 7 contributes a generalized PCN model in the context of automatic differentiation. Like generalized PCN models in neuroscience, it uses the Laplace approximation for state uncertainty estimation. We demonstrate the usefulness of the Laplace approximation by comparing a static version of the model with a VAE. The model efficiently projects spatio-temporal data from discrete timesteps to generalized coordinates by interpreting the underlying discrete Taylor expansion as a convolutional kernel over the time axis. The model encodes latent states hierarchically, using "cause" states, while "hidden" states capture the dynamics in each layer. We show that the model is able to represent simple spatio-temporal sequences with respect to the expected change in latent states. By clustering the network's latent states, we show that model learns to separate temporal changes captured in the hidden states from perturbations to these dynamics, encoded by cause states.

8.0.2 *Limitations and future work*

In this work, we have reviewed and implemented ANN based models that are informed by the architectural constrains and the learning rules of process models under the Free Energy Principle. We also evaluated possibilities to apply such models of canonical computation in the context of information retrieval from human EEG data. However, both areas are still underexplored and many limitations are still to be solved. The following paragraphs discuss these limitations and provide an outlook over possible future research.

SCALING UP PREDICTIVE CODING MODELS WITH LOCAL LEARNING RULES

A large portion of this work has been devoted to implementing and analysing predictive coding models in the context of artificial neural networks (ANNs). We reviewed PredNet, a popular DNN based model that is inspired by PC [124]. While our empirical results on a video action classification dataset highlight important deviations of PredNet from hierarchical PC models, the results of the study are restricted to

the particular application - action classification. While PredNet deviates significantly from hierarchical PC architectures in neuroscience [51], its hierarchically error-predicting organisation with respect to being specifically a *dynamical* model have yet to be explored in future work. A particularly interesting avenue could be to relate PredNet's hierarchy of error-predicting modules to the generalized PC models discussed in Chapter 7, since these models cover a hierarchy over hidden state dynamics in each hierarchical layer.

In this work, we contributed a deep autoregressive PCN that, like PredNet, uses BPTT to learn model parameters from sequential data. The proposed model conforms to hierarchical PC more closely than PredNet, since it predicts sequences of latent states top-down, instead of predicting errors. This hierarchy arguably makes the model more biologically plausible under the FEP. However, the model still resorts to BPTT, which is a credit assignment mechanism that is not plausibly implemented in the brain, primarily because it requires exact sequence recall when computing the gradients [119]. For this reason, from the perspective of modelling PC as a canonical computation in the brain, this is a problematic modelling decision. At the moment of writing, BPTT still is substantially more efficient and accurate than other, more biologically plausible temporal credit assignment mechanisms [119]. Next to this deep predictive coding model, we also contribute more directly biologically plausible PCN architectures in this work. These PCNs use updates that are strictly local, both hierarchically and in time. This essentially renders the contributed models a form of Bayesian filters, since they process temporal information online and strictly forward in time. Future work should address the performance gap between these two approaches to temporal credit assignment. In this work, we contributed a generalized PC model, that takes a third approach, by converting discrete data into a continuous-time representations using a Taylor expansion. This approach allows to encode sequential data locally in time, since higher order temporal derivatives are modelled. The results on generalized PC presented in this work, however, are still largely a proof-of-concept. In particular, the conversion from discrete sequences to generalized coordinates requires high sampling rates, which are usually not present in machine learning video datasets. Future work could address this problem, for example by simply supplying more high-resolution data. It should be noted that from the perspective of biological plausibility, this issue is less severe, since physical data, as processed in the brain, generally has high resolution.

INFORMATION RETRIEVAL WITH FEP MODELS

Investigating information retrieval from human brain signals is an exciting, yet challenging area of research. Here, we were driven by

the hypothesis that ANN models based on the FEP have an inherent aspect of biological plausibility, while still offering the possibility to deal with complex data, such as required, e.g. in audio or EEG processing. We addressed information retrieval from human brain signals, as captured in EEG, using two different methodologies.

Firstly, we addressed shared representation learning, where a joint latent representation encodes the mutual information between the model's expectation and the human brain signal. In the context of shared representation learning with auditory EEG, we found that multi-head VAEs allow to reconstruct simple stimuli, especially when the inputs are averaged. Nevertheless, our results are still far from perfect reconstructions, especially when applied to more complex audio stimuli, such as the pop songs in the NMED-T dataset [123]. We found that the model reconstructions are often difficult to interpret qualitatively. Similarly, the squared prediction error, as it is applied to high-dimensional spatial input covering the frequency and temporal domain appears to be a relatively poor evaluation metric regarding musically relevant content. One way to address this, is to use smaller, i.e. temporally shorter, inputs to the model. We have chosen this approach in the context of our proposed autoregressive PCN model. This approach, however, requires sequential predictions, which makes the model substantially more complex than the multi-head VAE. Other approaches could include more elaborate metrics, such as multi-scale spectral losses [45], or even resort to a sonification of the reconstructed audio. Recent years have seen an increase in EEG datasets particularly suited for large-scale machine learning applications, such as the NMED-T and ZuCO dataset used in this work [82, 123]. Despite this progress, another crucial limitation in the context of shared representation learning from EEG data is still the size of the available datasets.

Next to shared representation learning, we also evaluated the possibility to use a deep autoregressive predictive coding network for unsupervised prediction of EEG. While the network is able to make short-term predictions over expected future EEG signal, the contributed model is far from making accurate long-term predictions, leaving ample room for future work. We demonstrated the possibility to actively infer temporal positions in EEG signal that optimally fit the expectancy of the network, in a process inspired by active inference under the FEP [2, 57]. This mechanism itself is useful, e.g. to correct accidental temporal misalignment's during EEG recording. More elaborate active inference models, however, model complex action sequences, called policies, when interacting with sensory data [47, 57, 210]. Investigating such more elaborate models in the context of (online) EEG processing is a promising direction of research, especially with respect to configurations where the generated EEG data can be influenced, such as in Brain-Computer Interfaces [1, 102].

In this work, we did not evaluate a static predictive coding model on EEG data. Given the similarities between (hierarchical) VAEs and hierarchical PCNs, this is another promising area for future research.

We also addressed a more indirect approach to information retrieval, that involves applying the model the sensory data and using the model's error response to retrieve candidate temporal locations for human evoked responses. The approach works well when applied to an entire dataset of aligned audio and EEG recordings. In this work, we focused on a "global average" ERP, which simply includes all trials from all subjects in the test dataset. We found that for the NMED-T dataset, averaging over predicted ERP locations in a single song still leads to meaningful results, since each song covers multiple minutes. Due to the nature of averaging, the approach works less well, when the amount of data gets smaller, e.g. when a single, temporally short trial is used. In the presented study, we thresholded the model's averaged prediction error using a fixed magnitude. This approach does not cover more fine-grained aspects of the prediction error, such as its distribution over frequencies in the audio. Future work could thus employ more elaborate error thresholding mechanisms, possible learned directly by the model, to retrieve more specific information.

A

APPENDIX

A.1 ADDITIONAL PUBLICATIONS

Several additional publications were created during the course of writing, which were not directly used in this thesis. This section briefly summarizes their content.

[A:1] A. Ofner and S. Stober. "PredProp: Bidirectional stochastic optimization with precision weighted predictive coding." In: arXiv preprint arXiv:2111.08792 (2021).

This paper proposes a novel optimisation method for gradient-based predictive coding networks with a focus on the precision of the prediction errors after propagation to the optimised parameters.

[A:2] A. Ofner and S. Stober. "Towards bridging human and artificial cognition: Hybrid variational predictive coding of the physical world, the body and the brain." In: Physics Workshop at NeurIPS 2018 (2018).

This paper discusses an early-stage model that aims at integrating human and artificial generative models using a DNN model that predicts the motion of human subjects in video based on recorded EEG signals.

[A:3] A. Ofner and S. Stober. "Knowledge transfer in coupled predictive coding networks." In: Bernstein Conference 2019 (2019).

This paper employs a version of the model discussed in Chapter 5 in the context of two coupled networks. We show that knowledge about stimuli that are not visible to one model can be extracted from the response of a second network that has access to the stimuli.

[A:4] A. Ofner and S. Stober. "Distributed Planning with Active Inference." In: Bernstein Conference 2021 (2021).

This paper proposes an active inference model, where multiple independent planners are embodied within one agent and are predicted top-down. By actively exploring the preferences of the ensemble of planners, the agent learns to memorize and exploit their behavior.

[A:5] A. Ofner, R. K. Ratul, S. Ghosh, and S. Stober. "Predictive coding, precision and natural gradients." In: arXiv preprint arXiv:2111.06942 (2021).

This paper explores the estimation of prediction error precision in predictive coding networks.

[A:6] A. Ofner and S. Stober. "Hybrid Active Inference." In: arXiv preprint arXiv:1810.02647 (2018).

This paper reviews possibilities to employ active inference as a guiding principle to design neuro-hybrid artificial intelligence.

[A:7] A. Ofner and S. Stober. "Differentiable Generalised Predictive Coding." In: arXiv preprint arXiv:2112.03378 (2021).

This paper reports early-stage progress on the generalized predictive coding model discussed in Chapter 7 of this thesis. It primarily evaluates possibilities to implement dynamical predictions in generalized coordinates in the context of automatic differentiation.

A.2 GENERALIZED PREDICTIVE CODING

	MNIST	OMNIGLOT	fMNIST
VAE	37.7±0.4	32.8±0.3	30.3±0.6
GPC-S (3)	37.5±0.1	34.1±0.1	30.9±0.3
GPC-M (3)	37.8±0.1	33.3±0.1	35.7±0.1
VLAE (3)	36.8±0.1	34.8±0.1	29.8±0.1
GPC-S (6)	39.0±0.1	36.2±0.1	31.2±0.2
GPC-M (6)	38.0±0.1	34.5±0.1	33.4±0.1
VLAE (6)	37.0±0.1	36.0±0.1	30.0±0.1

Table 11: Posterior complexity (test set) of models trained on the static prediction task for MNIST, OMNIGLOT and Fashion MNIST in terms of mean and standard deviation over ten runs.

A.3 SHARED REPRESENTATION OF AUDIO AND EEG

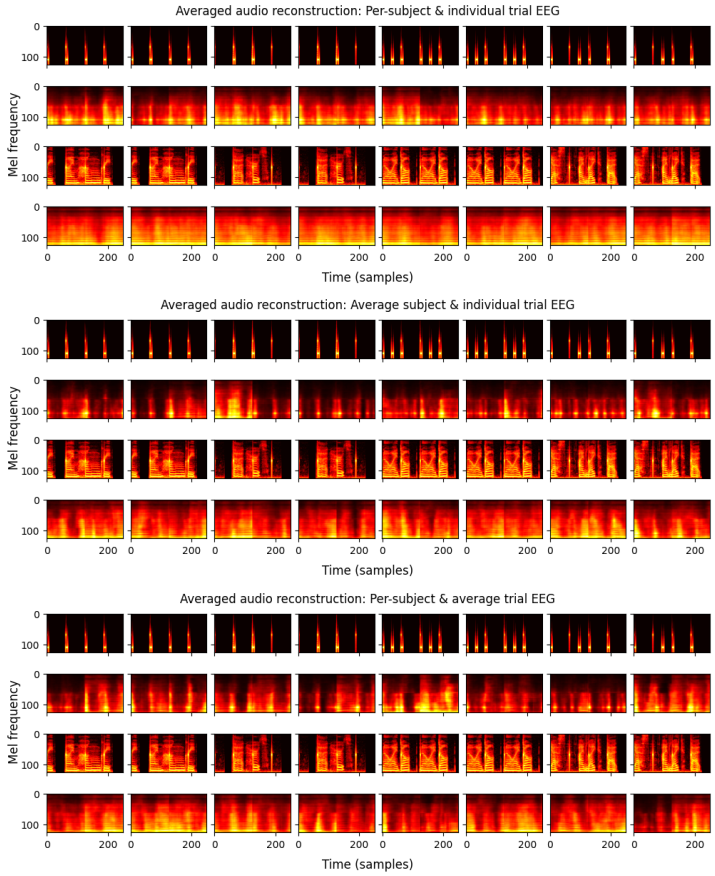


Figure 45: Averaged audio reconstructions from a model trained on the OpenMIIR dataset without averaged EEG data. First, predictions are made from individual EEG inputs or from EEG that has been averaged across subjects (within the same trial) or across trials (within the same subject). After inference, the mean of the audio reconstructions is computed for each trial.

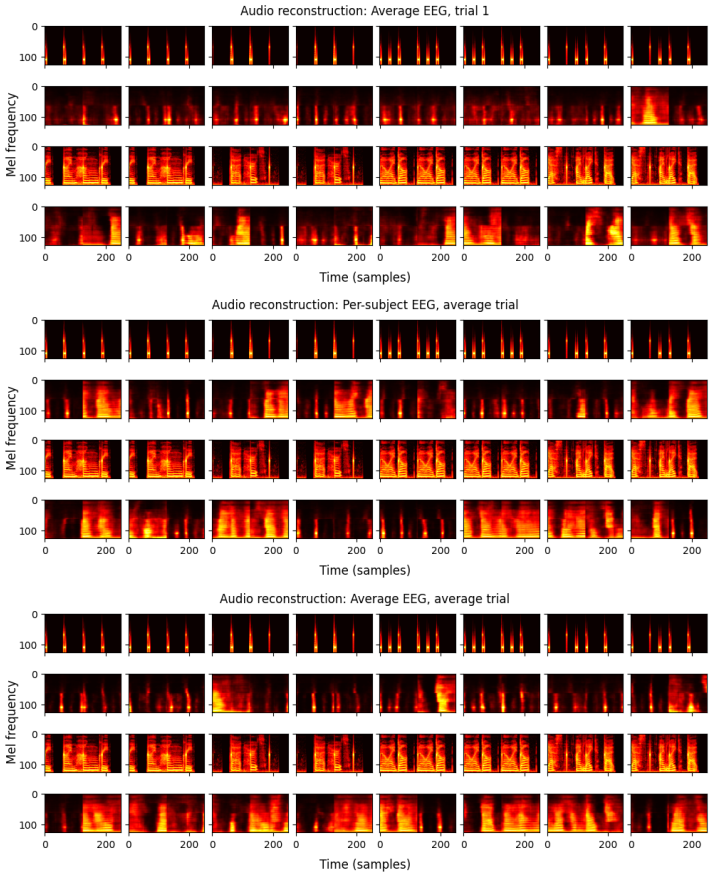


Figure 46: Reconstructing audio stimuli from averaged EEG inputs after training the model the OpenMIIR dataset without averaged EEG data. Shown are all 16 different trial types, i.e. 8 rhythmic (top row) and the corresponding 8 speech trials (bottom row). EEG inputs are either averaged across subjects, over trials or across both dimensions.

BIBLIOGRAPHY

[1] R. Abiri, S. Borhani, E. W. Sellers, Y. Jiang, and X. Zhao. "A comprehensive review of EEG-based brain–computer interface paradigms." In: *Journal of neural engineering* 16.1 (2019), p. 011001.

[2] R. A. Adams, K. J. Friston, and A. M. Bastos. "Active inference, predictive coding and cortical architecture." In: *Recent Advances on the Modular Organization of the Cortex*. Springer, 2015, pp. 97–121.

[3] A. Ahmadi and J. Tani. "Bridging the gap between probabilistic and deterministic models: a simulation study on a variational Bayes predictive coding recurrent neural network model." In: *International conference on neural information processing*. Springer. 2017, pp. 760–769.

[4] A. Ahmadi and J. Tani. "A novel predictive-coding-inspired variational rnn model for online prediction and recognition." In: *Neural computation* 31.11 (2019), pp. 2025–2074.

[5] L. Aitchison and M. Lengyel. "With or without you: predictive coding and Bayesian inference in the brain." In: *Current opinion in neurobiology* 46 (2017), pp. 219–227.

[6] A. Aroudi, B. Mirkovic, M. De Vos, and S. Doclo. "Auditory attention decoding with EEG recordings using noisy acoustic reference signals." In: *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE. 2016, pp. 694–698.

[7] B. S. Atal and M. R. Schroeder. "Adaptive predictive coding of speech signals." In: *Bell System Technical Journal* 49.8 (1970), pp. 1973–1986.

[8] R. Auksztulewicz and K. Friston. "Repetition suppression and its contextual determinants in predictive coding." In: *cortex* 80 (2016), pp. 125–140.

[9] T. Baccino and Y. Manunta. "Eye-fixation-related potentials: Insight into parafoveal processing." In: *Journal of Psychophysiology* 19.3 (2005), p. 204.

[10] T. Baldeweg. "ERP repetition effects and mismatch negativity generation: a predictive coding perspective." In: *Journal of Psychophysiology* 21.3-4 (2007), pp. 204–213.

[11] A. M. Bastos, W. M. Usrey, R. A. Adams, G. R. Mangun, P. Fries, and K. J. Friston. "Canonical microcircuits for predictive coding." In: *Neuron* 76.4 (2012), pp. 695–711.

[12] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. "Variational inference: A review for statisticians." In: *Journal of the American statistical Association* 112.518 (2017), pp. 859–877.

[13] S. Böck, F. Krebs, and G. Widmer. "A Multi-model Approach to Beat Tracking Considering Heterogeneous Music Styles." In: *Proc. of the 15th Int. Society for Music Information Retrieval Conference*. 2014.

[14] R. Bogacz. "A tutorial on the free-energy framework for modelling perception and learning." In: *Journal of mathematical psychology* 76 (2017), pp. 198–211.

[15] D. Bolis and L. Schilbach. "Beyond one Bayesian brain: Modeling intra-and inter-personal processes during social interaction: Commentary on "Mentalizing homeostasis: The social origins of interoceptive inference" by Fotopoulou & Tsakiris." In: *Neuropsychoanalysis* 19.1 (2017), pp. 35–38.

[16] R. M. Brown and C. Palmer. "Auditory and motor imagery modulate learning in music performance." In: *Frontiers in human neuroscience* 7 (2013), p. 320.

[17] L. Buesing, T. Weber, S. Racaniere, S. Eslami, D. Rezende, D. P. Reichert, F. Viola, F. Besse, K. Gregor, D. Hassabis, et al. "Learning and querying fast generative models for reinforcement learning." In: *arXiv preprint arXiv:1802.03006* (2018).

[18] S. Chandar, M. M. Khapra, H. Larochelle, and B. Ravindran. "Correlational neural networks." In: *Neural computation* 28.2 (2016), pp. 257–285.

[19] M. Choi and J. Tani. "Predictive coding for dynamic visual processing: Development of functional hierarchy in a multiple spatiotemporal scales rnn model." In: *Neural computation* 30.1 (2018), pp. 237–270.

[20] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio. "A recurrent latent variable model for sequential data." In: *Advances in neural information processing systems* 28 (2015).

[21] Y.-A. Chung, W.-N. Hsu, H. Tang, and J. Glass. "An unsupervised autoregressive model for speech representation learning." In: *arXiv preprint arXiv:1904.03240* (2019).

[22] L. K. Cirelli, D. Bosnyak, F. C. Manning, C. Spinelli, C. Marie, T. Fujioka, A. Ghahremani, and L. J. Trainor. "Beat-induced fluctuations in auditory cortical beta-band activity: using EEG to measure age-related changes." In: *Frontiers in psychology* 5 (2014), p. 742.

[23] J. D. Cohen, S. M. McClure, and A. J. Yu. "Should I stay or should I go? How the human brain manages the trade-off between exploitation and exploration." In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 362.1481 (2007), pp. 933–942.

[24] R. Collobert and J. Weston. "A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning." In: *Proceedings of the 25th International Conference on Machine Learning*. ICML '08. Helsinki, Finland: ACM, 2008, pp. 160–167. ISBN: 978-1-60558-205-4. DOI: 10.1145/1390156.1390177. URL: http://doi.acm.org/10.1145/1390156.1390177.

[25] F. Crick. "The recent excitement about neural networks." In: *Nature* 337.6203 (1989), pp. 129–132.

[26] I. Daly, N. Nicolaou, D. Williams, F. Hwang, A. Kirke, E. Miranda, and S. J. Nasuto. "Neural and physiological data from participants listening to affective music." In: *Scientific Data* 7.1 (2020), pp. 1–7.

[27] S. Dasgupta and D. Hsu. "Hierarchical sampling for active learning." In: *Proceedings of the 25th international conference on Machine learning*. 2008, pp. 208–215.

[28] M. E. Davies, N. Degara, and M. D. Plumbley. "Evaluation methods for musical audio beat tracking algorithms." In: *Queen Mary University of London, Centre for Digital Music, Tech. Rep. C4DM-TR-09-06* (2009).

[29] K. De and V Masilamani. "Image Sharpness Measure for Blurred Images in Frequency Domain." In: *Procedia Engineering* 64 (Dec. 2013). DOI: 10.1016/j.proeng.2013.09.086.

[30] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson. "Fma: A dataset for music analysis." In: *arXiv preprint arXiv:1612.01840* (2016).

[31] A. P. Dempster, N. M. Laird, and D. B. Rubin. "Maximum likelihood from incomplete data via the EM algorithm." In: *Journal of the royal statistical society: series B (methodological)* 39.1 (1977), pp. 1–22.

[32] S. L. Denham and I. Winkler. "Predictive coding in auditory perception: challenges and unresolved questions." In: *European Journal of Neuroscience* 51.5 (2020), pp. 1151–1160.

[33] A. Doerr, C. Daniel, M. Schiegg, D. Nguyen-Tuong, S. Schaal, M. Toussaint, and S. Trimpe. "Probabilistic recurrent state-space models." In: *arXiv preprint arXiv:1801.10395* (2018).

[34] B van Domselaar and P. W. Hemker. "Nonlinear parameter estimation in initial value problems." In: *Stichting Mathematisch Centrum. Numerieke Wiskunde* NW 18/75 (1975).

[35] S. Dora, C. Pennartz, and S. Bohte. "A deep predictive coding network for inferring hierarchical causes underlying sensory inputs." In: *International Conference on Artificial Neural Networks*. Springer. 2018, pp. 457–467.

[36] R. J. Douglas and K. Martin. "A functional microcircuit for cat visual cortex." In: *The Journal of physiology* 440.1 (1991), pp. 735–769.

[37] R. J. Douglas, K. A. Martin, and D. Whitteridge. "A canonical microcircuit for neocortex." In: *Neural computation* 1.4 (1989), pp. 480–488.

[38] K. Doya, S. Ishii, A. Pouget, and R. P. Rao. *Bayesian brain: Probabilistic approaches to neural coding*. MIT press, 2007.

[39] C. Du, C. Du, and H. He. "Sharing deep generative representation for perceived image reconstruction from human brain activity." In: *Neural Networks (IJCNN), 2017 International Joint Conference on*. IEEE. 2017, pp. 1049–1056.

[40] P. Elias. "Predictive coding–I." In: *IRE transactions on information theory* 1.1 (1955), pp. 16–24.

[41] D. P. Ellis. "Beat tracking by dynamic programming." In: *Journal of New Music Research* 36.1 (2007), pp. 51–60.

[42] J. L. Elman. "Finding structure in time." In: *Cognitive science* 14.2 (1990), pp. 179–211.

[43] N. Elsayed, A. S. Maida, and M. Bayoumi. "Reduced-Gate Convolutional LSTM Architecture for Next-Frame Video Prediction Using Predictive Coding." In: *2019 International Joint Conference on Neural Networks (IJCNN)*. 2019, pp. 1–9. DOI: 10.1109/IJCNN.2019.8852480.

[44] N. Elsayed, A. S. Maida, and M. Bayoumi. "Reduced-gate convolutional long short-term memory using predictive coding for spatiotemporal prediction." In: *Computational Intelligence* 36.3 (2020), pp. 910–939.

[45] J. Engel, L. Hantrakul, C. Gu, and A. Roberts. "DDSP: Differentiable Digital Signal Processing." In: *arXiv preprint arXiv:2001.04643* (2020).

[46] H. Feldman and K. J. Friston. "Attention, uncertainty, and free-energy." In: *Frontiers in human neuroscience* 4 (2010), p. 215.

[47] Z. Fountas, N. Sajid, P. Mediano, and K. Friston. "Deep active inference agents using Monte-Carlo methods." In: *Advances in neural information processing systems* 33 (2020), pp. 11662–11675.

[48] K. Friston. "Learning and inference in the brain." In: *Neural Networks* 16.9 (2003), pp. 1325–1352.

[49] K. Friston. "Hierarchical models in the brain." In: *PLoS computational biology* 4.11 (2008), e1000211.

[50] K. Friston. "The free-energy principle: a unified brain theory?" In: *Nature reviews neuroscience* 11.2 (2010), pp. 127–138.

[51] K. Friston and S. Kiebel. "Predictive coding under the free-energy principle." In: *Philosophical transactions of the Royal Society B: Biological sciences* 364.1521 (2009), pp. 1211–1221.

[52] K. Friston, J. Kilner, and L. Harrison. "A free energy principle for the brain." In: *Journal of physiology-Paris* 100.1-3 (2006), pp. 70–87.

[53] K. Friston, J. Mattout, and J. Kilner. "Action understanding and active inference." In: *Biological cybernetics* 104.1 (2011), pp. 137–160.

[54] K. Friston, J. Mattout, N. Trujillo-Barreto, J. Ashburner, and W. Penny. "Variational free energy and the Laplace approximation." In: *Neuroimage* 34.1 (2007), pp. 220–234.

[55] K. Friston, F. Rigoli, D. Ognibene, C. Mathys, T. Fitzgerald, and G. Pezzulo. "Active inference and epistemic value." In: *Cognitive neuroscience* 6.4 (2015), pp. 187–214.

[56] K. J. Friston. "Variational filtering." In: *NeuroImage* 41.3 (2008), pp. 747–766.

[57] K. J. Friston, M. Lin, C. D. Frith, G. Pezzulo, J. A. Hobson, and S. Ondobaka. "Active inference, curiosity and insight." In: *Neural computation* 29.10 (2017), pp. 2633–2683.

[58] K. J. Friston, T. Parr, and B. de Vries. "The graphical brain: belief propagation and active inference." In: *Network neuroscience* 1.4 (2017), pp. 381–414.

[59] Y. Fujiwara, Y. Miyawaki, and Y. Kamitani. "Modular encoding and decoding models derived from bayesian canonical correlation analysis." In: *Neural computation* 25.4 (2013), pp. 979–1005.

[60] K. Fukushima. "Neocognitron: A hierarchical neural network capable of visual pattern recognition." In: *Neural networks* 1.2 (1988), pp. 119–130.

[61] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. "Vision meets Robotics: The KITTI Dataset." In: *International Journal of Robotics Research (IJRR)* (2013).

[62] S. J. Gershman. "What does the free energy principle tell us about the brain?" In: *arXiv preprint arXiv:1901.07945* (2019).

[63] S. Geuter, M. A. Lindquist, and T. D. Wager. "Fundamentals of functional neuroimaging." In: (2017).

[64] R. B. Girshick. "Fast R-CNN." In: *CoRR* abs/1504.08083 (2015). arXiv: 1504.08083. URL: http://arxiv.org/abs/1504.08083.

[65] M. González, E. Rojas, W. Bolaños, J. P. Segura, L. Murillo, A. Solano, E. González, N. Röhner, and L. Yu. "Auditory imagery classification with a non-invasive Brain Computer Interface." In: *2019 9th International IEEE/EMBS Conference on Neural Engineering (NER)*. IEEE. 2019, pp. 151–154.

[66] R. Goyal et al. "The "something something" video database for learning and evaluating visual common sense." In: *CoRR* abs/1706.04261 (2017). arXiv: 1706.04261. URL: http://arxiv.org/abs/1706.04261.

[67] A. Gramfort, M. Luessi, E. Larson, D. A. Engemann, D. Strohmeier, C. Brodbeck, R. Goj, M. Jas, T. Brooks, L. Parkkonen, et al. "MEG and EEG data analysis with MNE-Python." In: *Frontiers in neuroscience* 7 (2013), p. 267.

[68] P. Grosche and M. Muller. "Extracting predominant local pulse information from music recordings." In: *IEEE Transactions on Audio, Speech, and Language Processing* 19.6 (2010), pp. 1688–1701.

[69] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson. "Learning latent dynamics for planning from pixels." In: *arXiv preprint arXiv:1811.04551* (2018).

[70] K. Han, H. Wen, Y. Zhang, D. Fu, E. Culurciello, and Z. Liu. "Deep predictive coding network with local recurrent processing for object recognition." In: *Advances in Neural Information Processing Systems*. 2018, pp. 9201–9213.

[71] K. Han, H. Wen, Y. Zhang, D. Fu, E. Culurciello, and Z. Liu. "Deep predictive coding network with local recurrent processing for object recognition." In: *Advances in neural information processing systems* 31 (2018).

[72] T. Han, W. Xie, and A. Zisserman. "Video representation learning by dense predictive coding." In: *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*. 2019, pp. 0–0.

[73] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles. "ActivityNet: A large-scale video benchmark for human activity understanding." In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015), pp. 961–970.

[74] M. Heilbron and M. Chait. "Great expectations: is there evidence for predictive coding in auditory cortex?" In: *Neuroscience* 389 (2018), pp. 54–73.

[75] O. Henaff. "Data-efficient image recognition with contrastive predictive coding." In: *International conference on machine learning*. PMLR. 2020, pp. 4182–4192.

[76] S. C. Herholz, A. R. Halpern, and R. J. Zatorre. "Neuronal correlates of perception, imagery, and memory for familiar tunes." In: *Journal of cognitive neuroscience* 24.6 (2012), pp. 1382–1397.

[77] I. Hertrich, S. Dietrich, J. Trouvain, A. Moos, and H. Ackermann. "Magnetic brain activity phase-locked to the envelope, the syllable onsets, and the fundamental frequency of a perceived speech signal." In: *Psychophysiology* 49.3 (2012), pp. 322–334.

[78] G. E. Hinton and D. Van Camp. "Keeping the neural networks simple by minimizing the description length of the weights." In: *Proceedings of the sixth annual conference on Computational learning theory*. 1993, pp. 5–13.

[79] I. Hipólito, M. J. Ramstead, L. Convertino, A. Bhat, K. Friston, and T. Parr. "Markov blankets in the brain." In: *Neuroscience & Biobehavioral Reviews* 125 (2021), pp. 88–97.

[80] S. Hochreiter and J. Schmidhuber. "Long short-term memory." In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[81] K. Hoenig, C. Müller, B. Herrnberger, E.-J. Sim, M. Spitzer, G. Ehret, and M. Kiefer. "Neuroplasticity of semantic representations for musical instruments in professional musicians." In: *NeuroImage* 56.3 (2011), pp. 1714–1725.

[82] N. Hollenstein, J. Rotsztejn, M. Troendle, A. Pedroni, C. Zhang, and N. Langer. "ZuCo, a simultaneous EEG and eye-tracking resource for natural sentence reading." In: *Scientific data* 5.1 (2018), pp. 1–13.

[83] A. Holzapfel, M. E. Davies, J. R. Zapata, J. L. Oliveira, and F. Gouyon. "Selective sampling for beat tracking evaluation." In: *IEEE Transactions on Audio, Speech, and Language Processing* 20.9 (2012), pp. 2539–2548.

[84] M. Hosseini and A. Maida. "Hierarchical predictive coding models in a deep-learning framework." In: *arXiv preprint arXiv:2005.03230* (2020).

[85] H. Hotelling. "Relations between two sets of variates." In: *Biometrika* 28.3/4 (1936), pp. 321–377.

[86] R. J. Huster, S. Debener, T. Eichele, and C. S. Herrmann. "Methods for simultaneous EEG-fMRI: an introductory review." In: *Journal of Neuroscience* 32.18 (2012), pp. 6053–6060.

[87] J. Hwang, J. Kim, A. Ahmadi, M. Choi, and J. Tani. "Predictive coding-based deep dynamic neural network for visuomotor learning." In: *2017 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*. IEEE. 2017, pp. 132–139.

[88] J. Hwang, J. Kim, A. Ahmadi, M. Choi, and J. Tani. "Dealing with large-scale spatio-temporal patterns in imitative interaction between a robot and a human by using the predictive coding framework." In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 50.5 (2018), pp. 1918–1931.

[89] A. Irshad and M. Salman. "State-space approach to linear predictive coding of speech—A comparative assessment." In: *2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA)*. IEEE. 2013, pp. 886–890.

[90] T. Isomura, H. Shimazaki, and K. Friston. "Canonical neural networks perform active inference." In: *bioRxiv* (2020).

[91] L. Itti and P. Baldi. "Bayesian surprise attracts human attention." In: *Vision research* 49.10 (2009), pp. 1295–1306.

[92] E. Jones, T. Oliphant, and P. Peterson. "SciPy: open source scientific tools for Python." In: (2014).

[93] M. Jung, T. Matsumoto, and J. Tani. "Goal-directed behavior under variational predictive coding: Dynamic organization of visual attention and working memory." In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2019, pp. 1040–1047.

[94] T.-P. Jung, S. Makeig, M. Westerfield, J. Townsend, E. Courchesne, and T. J. Sejnowski. "Analyzing and visualizing single-trial event-related potentials." In: *Advances in neural information processing systems*. 1999, pp. 118–124.

[95] R. E. Kalman. "A new approach to linear filtering and prediction problems." In: (1960).

[96] M. Karl, M. Soelch, J. Bayer, and P. Van der Smagt. "Deep variational bayes filters: Unsupervised learning of state space models from raw data." In: *arXiv preprint arXiv:1605.06432* (2016).

[97] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. "Large-Scale Video Classification with Convolutional Neural Networks." In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 1725–1732. DOI: 10.1109/CVPR.2014.223.

[98] I. Kavasidis, S. Palazzo, C. Spampinato, D. Giordano, and M. Shah. "Brain2Image: Converting Brain Signals into Images." In: *Proceedings of the 2017 ACM on Multimedia Conference*. ACM. 2017, pp. 1809–1817.

[99] J. Kiefer and J. Wolfowitz. "Stochastic estimation of the maximum of a regression function." In: *The Annals of Mathematical Statistics* (1952), pp. 462–466.

[100] M. Kiefer and L. W. Barsalou. "15 grounding the human conceptual system in perception, action, and internal states." In: *Action science: Foundations of an emerging discipline* (2013), p. 381.

[101] M. Kiefer, E.-J. Sim, B. Herrnberger, J. Grothe, and K. Hoenig. "The sound of concepts: four markers for a link between auditory and conceptual brain systems." In: *Journal of Neuroscience* 28.47 (2008), pp. 12224–12230.

[102] D.-W. Kim, J.-C. Lee, Y.-M. Park, I.-Y. Kim, and C.-H. Im. "Auditory brain-computer interfaces (BCIs) and their practical applications." In: *Biomedical Engineering Letters* 2 (2012), pp. 13–17.

[103] D. P. Kingma and J. Ba. "Adam: A method for stochastic optimization." In: *arXiv preprint arXiv:1412.6980* (2014).

[104] D. P. Kingma and M. Welling. "Auto-encoding variational bayes." In: *arXiv preprint arXiv:1312.6114* (2013).

[105] M. Kirchhoff, T. Parr, E. Palacios, K. Friston, and J. Kiverstein. "The Markov blankets of life: autonomy, active inference and the free energy principle." In: *Journal of The royal society interface* 15.138 (2018), p. 20170792.

[106] D. C. Knill and A. Pouget. "The Bayesian brain: the role of uncertainty in neural coding and computation." In: *TRENDS in Neurosciences* 27.12 (2004), pp. 712–719.

[107] S. Koelsch, P. Vuust, and K. Friston. "Predictive processes and the peculiar case of music." In: *Trends in Cognitive Sciences* 23.1 (2019), pp. 63–77.

[108] S. Koelstra, C. Muhl, M. Soleymani, J.-S. Lee, A. Yazdani, T. Ebrahimi, T. Pun, A. Nijholt, and I. Patras. "Deap: A database for emotion analysis; using physiological signals." In: *IEEE Transactions on Affective Computing* 3.1 (2012), pp. 18–31.

[109] A. M. Kondoz. *Digital speech: coding for low bit rate communication systems*. John Wiley & Sons, 2005.

[110] A. Krishnan and C. J. Plack. "Neural encoding in the human brainstem relevant to the pitch of complex tones." In: *Hearing research* 275.1-2 (2011), pp. 110–119.

[111] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "Imagenet classification with deep convolutional neural networks." In: *Communications of the ACM* 60.6 (2017), pp. 84–90.

[112] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. "HMDB: A Large Video Database for Human Motion Recognition." In: *Proceedings of the 2011 International Conference on Computer Vision*. ICCV '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 2556–2563. ISBN: 978-1-4577-1101-5. DOI: 10.1109/ICCV.2011.6126543. URL: http://dx.doi.org/10.1109/ICCV.2011.6126543.

[113] B. Kuznetsov, J. D. Parker, and F. Esqueda. "Differentiable IIR filters for machine learning applications." In: *Proc. Int. Conf. Digital Audio Effects (eDAFx-20)*. 2020, pp. 297–303.

[114] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum. "Human-level concept learning through probabilistic program induction." In: *Science* 350.6266 (2015), pp. 1332–1338.

[115] Y. LeCun. "The MNIST database of handwritten digits." In: *http://yann. lecun. com/exdb/mnist/* (1998).

[116] Y. LeCun, Y. Bengio, and G. Hinton. "Deep learning." In: *nature* 521.7553 (2015), pp. 436–444.

[117] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. "Backpropagation applied to handwritten zip code recognition." In: *Neural computation* 1.4 (1989), pp. 541–551.

[118] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition." In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[119] T. P. Lillicrap and A. Santoro. "Backpropagation through time and the brain." In: *Current opinion in neurobiology* 55 (2019), pp. 82–89.

[120] T. P. Lillicrap, A. Santoro, L. Marris, C. J. Akerman, and G. Hinton. "Backpropagation and the brain." In: *Nature Reviews Neuroscience* 21.6 (2020), pp. 335–346.

[121] G. W. Lindsay. "Convolutional neural networks as a model of the visual system: Past, present, and future." In: *Journal of cognitive neuroscience* 33.10 (2021), pp. 2017–2031.

[122] S. Lodato and P. Arlotta. "Generating neuronal diversity in the mammalian cerebral cortex." In: *Annual review of cell and developmental biology* 31 (2015), pp. 699–720.

[123] S. Losorelli, D. T. Nguyen, J. P. Dmochowski, and B. Kaneshiro. "NMED-T: A Tempo-Focused Dataset of Cortical and Behavioral Responses to Naturalistic Music." In: *ISMIR*. Vol. 3. 2017, p. 5.

[124] W. Lotter, G. Kreiman, and D. Cox. "Deep predictive coding networks for video prediction and unsupervised learning." In: *arXiv preprint arXiv:1605.08104* (2016).

[125] W. Lotter, G. Kreiman, and D. D. Cox. "Deep Predictive Coding Networks for Video Prediction and Unsupervised Learning." In: *CoRR* abs/1605.08104 (2016). arXiv: 1605.08104. URL: http://arxiv.org/abs/1605.08104.

[126] F. Mahdisoltani, G. Berger, W. Gharbieh, D. J. Fleet, and R. Memisevic. "Fine-grained Video Classification and Captioning." In: *CoRR* abs/1804.09235 (2018). arXiv: 1804.09235. URL: http://arxiv.org/abs/1804.09235.

[127] J. Marino. "Predictive coding, variational autoencoders, and biological connections." In: *arXiv preprint arXiv:2011.07464* (2020).

[128] J. Marino. "Predictive coding, variational autoencoders, and biological connections." In: *Neural Computation* 34.1 (2022), pp. 1–44.

[129] J. Marino, L. Chen, J. He, and S. Mandt. "Improving sequential latent variable models with autoregressive flows." In: *Symposium on advances in approximate bayesian inference*. PMLR. 2020, pp. 1–16.

[130] D. Marr. *Vision: A computational investigation into the human representation and processing of visual information*. MIT press, 2010.

[131] J. Martens. "New insights and perspectives on the natural gradient method." In: *The Journal of Machine Learning Research* 21.1 (2020), pp. 5776–5851.

[132] M. Mathieu, C. Couprie, and Y. LeCun. "Deep multi-scale video prediction beyond mean square error." In: *CoRR* abs/1511.05440 (2015). arXiv: 1511.05440. URL: http://arxiv.org/abs/1511.05440.

[133] L. Matthey, I. Higgins, D. Hassabis, and A. Lerchner. *dSprites: Disentanglement testing Sprites dataset*. https://github.com/deepmind/dsprites-dataset/. 2017.

[134] L. Matthey, I. Higgins, D. Hassabis, and A. Lerchner. "dSprites: Disentanglement testing sprites dataset, 2017." In: *URL https://github. com/deepmind/dsprites-dataset* (2020).

[135] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto. "librosa: Audio and music signal analysis in python." In: *Proceedings of the 14th python in science conference*. 2015, pp. 18–25.

[136] M. Meyer, S. Baumann, and L. Jancke. "Electrical brain imaging reveals spatio-temporal dynamics of timbre perception in humans." In: *Neuroimage* 32.4 (2006), pp. 1510–1523.

[137] M. A. Miguel, M. Sigman, and D. Fernandez Slezak. "From beat tracking to beat expectation: Cognitive-based beat tracking for capturing pulse clarity through time." In: *PloS one* 15.11 (2020), e0242207.

[138] B. Millidge. "Implementing predictive processing and active inference: Preliminary steps and results." In: *PsyArXiv - preprint* (2019).

[139] B. Millidge, A. Seth, and C. L. Buckley. "Predictive coding: a theoretical and experimental review." In: *arXiv preprint arXiv:2107.12979* (2021).

[140] B. Millidge, A. Tschantz, and C. L. Buckley. "Predictive coding approximates backprop along arbitrary computation graphs." In: *arXiv preprint arXiv:2006.04182* (2020).

[141] B. Millidge, A. Tschantz, A. Seth, and C. Buckley. "Neural Kalman Filtering." In: *arXiv preprint arXiv:2102.10021* (2021).

[142] B. Millidge, A. Tschantz, A. Seth, and C. L. Buckley. "Relaxing the constraints on predictive coding models." In: *arXiv preprint arXiv:2010.01047* (2020).

[143] M.-A. Moinnereau, T. Brienne, S. Brodeur, J. Rouat, K. Whittingstall, and E. Plourde. "Classification of auditory stimuli from EEG signals with a regulated recurrent neural network reservoir." In: *arXiv preprint arXiv:1804.10322* (2018).

[144] N. Moran. "Social implications arise in embodied music cognition research which can counter musicological "individualism"." In: *Frontiers in psychology* 5 (2014), p. 676.

[145] S. Morgan, J. Hansen, and S. Hillyard. "Selective attention to stimulus location modulates the steady-state visual evoked potential." In: *Proceedings of the National Academy of Sciences* 93.10 (1996), pp. 4770–4774.

[146] S. O. Murray, D. Kersten, B. A. Olshausen, P. Schrater, and D. L. Woods. "Shape perception reduces activity in human primary visual cortex." In: *Proceedings of the National Academy of Sciences* 99.23 (2002), pp. 15164–15169.

[147] R. Näätänen and T. Picton. "The N1 wave of the human electric and magnetic response to sound: a review and an analysis of the component structure." In: *Psychophysiology* 24.4 (1987), pp. 375–425.

[148] V. Nair and G. E. Hinton. "Rectified linear units improve restricted boltzmann machines." In: *Icml*. 2010.

[149] I. Nambu, M. Ebisawa, M. Kogure, S. Yano, H. Hokari, and Y. Wada. "Estimating the intended sound direction of the user: toward an auditory brain-computer interface using out-of-head sound localization." In: *PloS one* 8.2 (2013).

[150] K. V. Nourski, R. A. Reale, H. Oya, H. Kawasaki, C. K. Kovach, H. Chen, M. A. Howard, and J. F. Brugge. "Temporal envelope of time-compressed speech represented in the human auditory cortex." In: *Journal of Neuroscience* 29.49 (2009), pp. 15564–15574.

[151] S. Nozaradan, I. Peretz, and P. E. Keller. "Individual differences in rhythmic cortical entrainment correlate with predictive behavior in sensorimotor synchronization." In: *Scientific Reports* 6 (2016), p. 20612.

[152] S. Nozaradan, I. Peretz, M. Missal, and A. Mouraux. "Tagging the neuronal entrainment to beat and meter." In: *Journal of Neuroscience* 31.28 (2011), pp. 10234–10240.

[153] S. Nozaradan, I. Peretz, and A. Mouraux. "Selective neuronal entrainment to the beat and meter embedded in a musical rhythm." In: *Journal of Neuroscience* 32.49 (2012), pp. 17572–17581.

[154] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. "Wavenet: A generative model for raw audio." In: *arXiv preprint arXiv:1609.03499* (2016).

[155] A. v. d. Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. v. d. Driessche, E. Lockhart, L. C. Cobo, F. Stimberg, et al. "Parallel wavenet: Fast high-fidelity speech synthesis." In: *arXiv preprint arXiv:1711.10433* (2017).

[156] A. v. d. Oord, Y. Li, and O. Vinyals. "Representation learning with contrastive predictive coding." In: *arXiv preprint arXiv:1807.03748* (2018).

[157] J. Orchard and W. Sun. "Making predictive coding networks generative." In: *arXiv preprint arXiv:1910.12151* (2019).

[158] D. O'Shaughnessy. "Linear predictive coding." In: *IEEE potentials* 7.1 (1988), pp. 29–32.

[159] K. O'Shea and R. Nash. "An introduction to convolutional neural networks." In: *arXiv preprint arXiv:1511.08458* (2015).

[160] J. A. O'Sullivan, A. J. Power, N. Mesgarani, S. Rajaram, J. J. Foxe, B. G. Shinn-Cunningham, M. Slaney, S. A. Shamma, and E. C. Lalor. "Attentional selection in a cocktail party environment can be decoded from single-trial EEG." In: *Cerebral cortex* 25.7 (2015), pp. 1697–1706.

[161] P. Paavilainen, C. Kaukinen, O. Koskinen, J. Kylmälä, and L. Rehn. "Mismatch negativity (MMN) elicited by abstract regularity violations in two concurrent auditory streams." In: *Heliyon* 4.4 (2018), e00608.

[162] Y. Park, C. Kim, and G. Kim. "Variational laplace autoencoders." In: *International conference on machine learning*. PMLR. 2019, pp. 5032–5041.

[163] T. Parr, D. Markovic, S. J. Kiebel, and K. J. Friston. "Neuronal message passing using Mean-field, Bethe, and Marginal approximations." In: *Scientific reports* 9.1 (2019), pp. 1–18.

[164] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. "Automatic differentiation in pytorch." In: (2017).

[165] P. M. Picciotti, S. Giannantonio, G. Paludetti, and G. Conti. "Steady state auditory evoked potentials in normal hearing subjects: evaluation of threshold and testing time." In: *Orl* 74.6 (2012), pp. 310–314.

[166] T. W. Picton. *Human auditory evoked potentials*. Plural Publishing, 2010.

[167] G Plourde. "Auditory evoked potentials." In: *Best Practice & Research Clinical Anaesthesiology* 20.1 (2006), pp. 129–139.

[168] J. Qiu, G. Huang, and T. S. Lee. "A neurally-inspired hierarchical prediction network for spatiotemporal sequence learning and prediction." In: *arXiv preprint arXiv:1901.09002* (2019).

[169] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis. "mir_eval: A transparent implementation of common MIR metrics." In: *Proc. of the 15th Int. Society for Music Information Retrieval Conference*. 2014.

[170] A. E. Raftery. "Approximate Bayes factors and accounting for model uncertainty in generalised linear models." In: *Biometrika* 83.2 (1996), pp. 251–266.

[171] P. Rakic. "Evolution of the neocortex: a perspective from developmental biology." In: *Nature Reviews Neuroscience* 10.10 (2009), pp. 724–735.

[172] R. P. Rao and D. H. Ballard. "Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects." In: *Nature neuroscience* 2.1 (1999), pp. 79–87.

[173] F. Raposo, D. M. de Matos, R. Ribeiro, S. Tang, and Y. Yu. "Towards Deep Modeling of Music Semantics using EEG Regularizers." In: *arXiv preprint arXiv:1712.05197* (2017).

[174] B. A. Richards and T. P. Lillicrap. "Dendritic solutions to the credit assignment problem." In: *Current opinion in neurobiology* 54 (2019), pp. 28–36.

[175] S. Ruder. "An overview of gradient descent optimization algorithms." In: *arXiv preprint arXiv:1609.04747* (2016).

[176] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. *Learning internal representations by error propagation*. Tech. rep. California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[177] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. "Learning representations by back-propagating errors." In: *nature* 323.6088 (1986), pp. 533–536.

[178] O. Rybkin, K. Daniilidis, and S. Levine. "Simple and Effective VAE Training with Calibrated Decoders." In: *CoRR* abs/2006.13202 (2020). arXiv: 2006.13202. URL: https://arxiv.org/abs/2006.13202.

[179] N. Sajid, P. J. Ball, T. Parr, and K. J. Friston. "Active inference: demystified and compared." In: *Neural computation* 33.3 (2021), pp. 674–712.

[180] T. Salvatori, Y. Song, T. Lukasiewicz, R. Bogacz, and Z. Xu. "Predictive coding can do exact backpropagation on convolutional and recurrent neural networks." In: *arXiv preprint arXiv:2103.03725* (2021).

[181] R. Sato, H. Kashima, and T. Yamamoto. "Short-Term Precipitation Prediction with Skip-Connected PredNet." In: *Artificial Neural Networks and Machine Learning – ICANN 2018*. Ed. by V. Kůrková, Y. Manolopoulos, B. Hammer, L. Iliadis, and I. Maglogiannis. Cham: Springer International Publishing, 2018, pp. 373–382. ISBN: 978-3-030-01424-7.

[182] B. Scellier and Y. Bengio. "Equilibrium propagation: Bridging the gap between energy-based models and backpropagation." In: *Frontiers in computational neuroscience* 11 (2017), p. 24.

[183] R. S. Schaefer. *Images of time: temporal aspects of auditory and movement imagination*. 2014.

[184] R. S. Schaefer, P. Desain, and P. Suppes. "Structural decomposition of EEG signatures of melodic processing." In: *Biological psychology* 82.3 (2009), pp. 253–259.

[185] R. S. Schaefer, P. Desain, and P. Suppes. "Structural decomposition of EEG signatures of melodic processing." In: *Biological psychology* 82.3 (2009), pp. 253–259.

[186] G. Schuller and A. Hännä. "Low delay audio compression using predictive coding." In: *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 2. IEEE. 2002, pp. II–1853.

[187] P. Schwartenbeck, J. Passecker, T. U. Hauser, T. H. FitzGerald, M. Kronbichler, and K. J. Friston. "Computational mechanisms of curiosity and goal-directed exploration." In: *Elife* 8 (2019), e41703.

[188] A. K. Seth and K. J. Friston. "Active interoceptive inference and the emotional brain." In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 371.1708 (2016), p. 20160007.

[189] A. Shahin, L. E. Roberts, C. Pantev, L. J. Trainor, and B. Ross. "Modulation of P2 auditory-evoked responses by the spectral complexity of musical sounds." In: *Neuroreport* 16.16 (2005), pp. 1781–1785.

[190] A. Shahin, L. E. Roberts, C. Pantev, L. J. Trainor, and B. Ross. "Modulation of P2 auditory-evoked responses by the spectral complexity of musical sounds." In: *Neuroreport* 16.16 (2005), pp. 1781–1785.

[191] A. Sherstinsky. "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network." In: *Physica D: Nonlinear Phenomena* 404 (2020), p. 132306.

[192] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo. "Convolutional LSTM network: A machine learning approach for precipitation nowcasting." In: *Advances in neural information processing systems* 28 (2015).

[193] S. Shipp, R. A. Adams, and K. J. Friston. "Reflections on agranular architecture: predictive coding in the motor cortex." In: *Trends in neurosciences* 36.12 (2013), pp. 706–716.

[194] A. Shrestha and A. Mahmood. "Review of deep learning algorithms and architectures." In: *IEEE access* 7 (2019), pp. 53040–53065.

[195] B. Skerritt-Davis and M. Elhilali. "Computational framework for investigating predictive processing in auditory perception." In: *Journal of Neuroscience Methods* (2021), p. 109177.

[196] S. Song, P. J. Sjöström, M. Reigl, S. Nelson, and D. B. Chklovskii. "Highly nonrandom features of synaptic connectivity in local cortical circuits." In: *PLoS biology* 3.3 (2005), e68.

[197] Y. Song, T. Lukasiewicz, Z. Xu, and R. Bogacz. "Can the Brain Do Backpropagation?—Exact Implementation of Backpropagation in Predictive Coding Networks." In: *Advances in neural information processing systems* 33 (2020), pp. 22566–22579.

[198] N. Srivastava, E. Mansimov, and R. Salakhudinov. "Unsupervised learning of video representations using lstms." In: *International conference on machine learning*. PMLR. 2015, pp. 843–852.

[199] A. Sternin, S. Stober, J. Grahn, and A. Owen. "Tempo estimation from the EEG signal during perception and imagination of music." In: *International Workshop on Brain-Computer Music Interfacing/International Symposium on Computer Music Multidisciplinary Research (BCMI/CMMR)*. 2015.

[200] M. Steyvers, M. D. Lee, and E.-J. Wagenmakers. "A Bayesian analysis of human decision-making on bandit problems." In: *Journal of mathematical psychology* 53.3 (2009), pp. 168–179.

[201] S. Stober. "Toward studying music cognition with information retrieval techniques: Lessons learned from the OpenMIIR initiative." In: *Frontiers in psychology* 8 (2017), p. 1255.

[202] S. Stober, D. J. Cameron, and J. A. Grahn. "Using Convolutional Neural Networks to Recognize Rhythm Stimuli from Electroencephalography Recordings." In: *Advances in neural information processing systems*. 2014, pp. 1449–1457.

[203] S. Stober, A. Sternin, A. M. Owen, and J. A. Grahn. "Towards Music Imagery Information Retrieval: Introducing the OpenMIIR Dataset of EEG Recordings from Music Perception and Imagination." In: *ISMIR*. 2015, pp. 763–769.

[204] Z. Straka, T. Svoboda, and M. Hoffmann. "PreCNet: next frame video prediction based on predictive coding." In: *arXiv preprint arXiv:2004.14878* (2020).

[205] D. Stuss, A Toga, J Hutchison, and T. Picton. "Feedback evoked potentials during an auditory concept formation task." In: *Progress in brain research*. Vol. 54. Elsevier, 1980, pp. 403–409.

[206] J. Tani et al. "Accounting for the minimal self and the narrative self: Robotics experiments using predictive coding." In: *CEUR workshop proceedings*. Vol. 2287. 2019.

[207] M. S. Treder, H. Purwins, D. Miklody, I. Sturm, and B. Blankertz. "Decoding auditory attention to instruments in polyphonic music using single-trial EEG classification." In: *Journal of neural engineering* 11.2 (2014), p. 026009.

[208] A. Tschantz, B. Millidge, A. K. Seth, and C. L. Buckley. "Hybrid Predictive Coding: Inferring, Fast and Slow." In: *arXiv preprint arXiv:2204.02169* (2022).

[209] E. M. Turan and J. Jäschke. "Multiple shooting for training neural differential equations on time series." In: *IEEE Control Systems Letters* 6 (2021), pp. 1897–1902.

[210] K. Ueltzhöffer. "Deep Active Inference." In: *arXiv preprint arXiv:1709.02341* (2017).

[211] J. Ukita, T. Yoshida, and K. Ohki. "Characterisation of nonlinear receptive fields of visual neurons by convolutional neural network." In: *Scientific reports* 9.1 (2019), pp. 1–17.

[212] A. van den Oord, Y. Li, and O. Vinyals. "Representation Learning with Contrastive Predictive Coding." In: *arXiv e-prints*, arXiv:1807.03748 (2018), arXiv:1807.03748. arXiv: 1807.03748 [cs.LG].

[213] R. Vigo, M. Barcus, Y. Zhang, and C. Doan. "On the learnability of auditory concepts." In: *The Journal of the Acoustical Society of America* 134.5 (2013), pp. 4064–4064.

[214] P. Vuust and M. A. Witek. "Rhythmic complexity and predictive coding: a novel approach to modeling rhythm and meter perception in music." In: *Frontiers in psychology* 5 (2014), p. 1111.

[215] K. S. Walsh, D. P. McGovern, A. Clark, and R. G. O'Connell. "Evaluating the neurophysiological evidence for predictive processing as a model of perception." In: *Annals of the new York Academy of Sciences* 1464.1 (2020), pp. 242–268.

[216] W. Wang, X. Yan, H. Lee, and K. Livescu. "Deep variational canonical correlation analysis." In: *arXiv preprint arXiv:1610.03454* (2016).

[217] Y. Wang, Z. Gao, M. Long, J. Wang, and S. Y. Philip. "Predrnn++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning." In: *International Conference on Machine Learning*. PMLR. 2018, pp. 5123–5132.

[218] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. "Image quality assessment: From error visibility to structural similarity." In: *IEEE Transactions on Image Processing* 13.4 (2004), 600–612. DOI: 10.1109/tip.2003.819861.

[219] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. "Image Quality Assessment: From Error Visibility to Structural Similarity." In: *IEEE TRANSACTIONS ON IMAGE PROCESSING* 13.4 (2004), pp. 600–612.

[220] E. Watanabe, A. Kitaoka, K. Sakamoto, M. Yasugi, and K. Tanaka. "Illusory motion reproduced by deep neural networks trained for prediction." In: *Frontiers in psychology* (2018), p. 345.

[221] E. Watanabe, A. Kitaoka, K. Sakamoto, M. Yasugi, and K. Tanaka. "Illusory Motion Reproduced by Deep Neural Networks Trained for Prediction." In: *Frontiers in Psychology* 9 (2018), p. 345. ISSN: 1664-1078. DOI: 10.3389/fpsyg.2018.00345. URL: https://www.frontiersin.org/article/10.3389/fpsyg.2018.00345.

[222] H. Wen, K. Han, J. Shi, Y. Zhang, E. Culurciello, and Z. Liu. "Deep predictive coding network for object recognition." In: *International Conference on Machine Learning*. PMLR. 2018, pp. 5266–5275.

[223] H. Wen, K. Han, J. Shi, Y. Zhang, E. Culurciello, and Z. Liu. "Deep Predictive Coding Network for Object Recognition." In: *CoRR* abs/1802.04762 (2018). arXiv: 1802.04762. URL: http://arxiv.org/abs/1802.04762.

[224] P. J. Werbos. "Backpropagation through time: what it does and how to do it." In: *Proceedings of the IEEE* 78.10 (1990), pp. 1550–1560.

[225] J. C. Whittington and R. Bogacz. "An approximation of the error backpropagation algorithm in a predictive coding network with local hebbian synaptic plasticity." In: *Neural computation* 29.5 (2017), pp. 1229–1262.

[226] J. C. Whittington and R. Bogacz. "Theories of error backpropagation in the brain." In: *Trends in cognitive sciences* 23.3 (2019), pp. 235–250.

[227] I. Winkler and I. Czigler. "Evidence from auditory and visual event-related potential (ERP) studies of deviance detection (MMN and vMMN) linking predictive coding theories and perceptual object representations." In: *International Journal of Psychophysiology* 83.2 (2012), pp. 132–143.

[228] G. F. Woodman. "A brief introduction to the use of event-related potentials in studies of perception and attention." In: *Attention, Perception, & Psychophysics* 72.8 (2010), pp. 2031–2046.

[229] H. Xiao, K. Rasul, and R. Vollgraf. "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms." In: *arXiv preprint arXiv:1708.07747* (2017).

[230] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo. "Convolutional LSTM network: A machine learning approach for precipitation nowcasting." In: *Advances in neural information processing systems*. 2015, pp. 802–810.

[231] M. Ye, X. Peng, W. Gan, W. Wu, and Y. Qiao. "Anopcn: Video anomaly detection via deep predictive coding network." In: *Proceedings of the 27th ACM International Conference on Multimedia*. 2019, pp. 1805–1813.

[232] C. Zhang, J. Bütepage, H. Kjellström, and S. Mandt. "Advances in variational inference." In: *IEEE transactions on pattern analysis and machine intelligence* 41.8 (2018), pp. 2008–2026.

[233] J. Zhong, A. Cangelosi, X. Zhang, and T. Ogata. "AFA-PredNet: The action modulation within predictive coding." In: *arXiv preprint arXiv:1804.03826* (2018).

[234] J. Zhong, A. Cangelosi, X. Zhang, and T. Ogata. "AFA-PredNet: The action modulation within predictive coding." In: *CoRR* abs/1804.03826 (2018). arXiv: 1804.03826. URL: http://arxiv.org/abs/1804.03826.

[235] J. Zhong, T. Ogata, and A. Cangelosi. "Encoding Longer-term Contextual Multi-modal Information in a Predictive Coding Model." In: *CoRR* abs/1804.06774 (2018). arXiv: 1804.06774. URL: http://arxiv.org/abs/1804.06774.

DECLARATION

Declaration

Magdeburg, December 2022

André Ofner