Hannes Rapp

# SPIKING NEURAL MODELS & MACHINE LEARNING FOR SYSTEMS NEUROSCIENCE

LEARNING, COGNITION AND BEHAVIOR

# Spiking neural models & machine learning for systems neuroscience

**Learning, Cognition and Behavior.**

*Inaugural-Dissertation*

*zur*

*Erlangung des Doktorgrades*

*der Mathematisch-Naturwissenschaftlichen Fakultät*

*der Universität zu Köln*

*vorgelegt von*

*Hannes Andreas* RAPP

*aus Ulm*

*Universität zu Köln*

*2020*

*Berichterstatter/in:* *Prof. Dr. Martin Paul Nawrot*
*(Gutachter/in)*

*Prof. Dr. Silvia Daun*

*Prof. Dr. Thomas Nowotny*

*Tag der letzten mündlichen Prüfung:* 26.05.2020

*"The true sign of intelligence is not knowledge but imagination."*

*"It's not that I'm so smart, it's just that I stay with problems longer."*

Albert Einstein

# *Acknowledgements*

First and foremost, I would like to thank my supervisor Dr. Martin Paul NAWROT for being the kind and genuine person he is, for guidance and great support at any stage during my PhD - personally and scientifically. I want to thank you for the freedom to follow my own (crazy) ideas and your belief in myself and in what I'm pursuing will be relevant and successful. I also want to acknowledge the rest of our lab-gang, particularly our female fraction, for fostering the soul of our group.

Additionally, I want to acknowledged the following people as they have provided scientific inspiration, supported me during the PhD in general or contributed in other non-materialistic ways. They are acknowledged here in no particular order:

- My wife Nicole
- My sister Martina
- My mom and my dad, Jutta & Günther
- Daniel F.
- Silvia L.-F.
- My Grandpas and Grandmas
- Uli & Lilly
- My highschool teachers in Physics & English
- Moritz Deger
- Yifat Prut
- Merav Stern
- Kenji Doya
- Erik de Schutter

# Contents

# List of Figures

# List of Tables

*Dedicated to my beloved family, great friends and my never-ending curiosity.*

# Chapter 1

# Introduction

Learning, cognition and the ability to navigate, interact and manipulate the world around us by performing appropriate behavior are hallmarks of artificial as well as biological intelligence. This thesis is devoted to study these three major pillars of intelligence using a computational approach. It promotes the importance of considering all three of them as an integrative system and within context of the underlying (behavioral) problem to be solved, instead of treating them as individual, disconnected entities. The thesis approaches this from two related perspectives, artificial intelligent systems and biological intelligent systems. However, it is mostly concerned with biologically relevant behavioral and cognitive problems using the computational substrate of nervous systems: neurons, synapses and action potentials. As such, the problems of learning, cognition and behavior are being studied and organized in three publications across different levels of detail: At the level of single neurons to solve numerical cognition tasks, functional networks thereof in small-brained animals during foraging and applied machine learning for a statistical understanding of the functional role of the cerebellar-thalamo-cortical system for motor control in primates.

## 1.1 Motivation

Cognition refers to the mental process of knowledge acquisition and understanding through reasoning, experience (memory) and sensation. Sensory systems allow organisms to acquire information about their surrounding environment and thus constitutes the major source of input to an intelligent system. From a computational perspective, this brings up two major challenges: First, how to efficiently process sensory information to capture and retain the most salient features of the environment by eliminating noise and other irrelevant or redundant information. And secondly, how to encode, represent and store the relevant sensory information within the nervous system. The specific computational mechanisms of processing and neural encoding employed within each sensory sub-system can have a big impact on the ability to learning, speed of perception and required neuronal resources. A simple neural code might need only few neuronal resources but renders higher brain areas unable to learn and form memories. As such it is important to consider sensory systems as a unit constituted of processing and neural coding. This makes it possible to come up with the learning rule or behavioral problem solving strategy that best exploits the specific computational mechanisms present in the sensory system under investigation.

While sensory systems constitute the major input source, behavior (e.g. motor control) is the major output component of an intelligent system. In order to execute certain behavior the nervous system must be able to make decisions and generate appropriate motor commands. Apart from internal world models, metabolic states and previous experiences these decisions often are based on sensory input and the context of the specific (behavioral) problem to be solved. Thus, there's a causal relationship between sensory processing and how it gets transformed into appropriate behavior, for example through decision making, experience or innate reflexive behavior. Consequently, studying sensory processing and generation of behavioral output as separate problems, is not an optimal approach.

Finally, learning is a key ingredient to enable efficient sensory information processing and behavioral output generation. It further plays an important role in connecting both of which to transform sensory information into appropriate behavioral output. Learning of sensory cues allows to form memories and associations with entities of the surrounding environment. For behavior, learning is necessary to adapt motor output based on metabolic state or changes in the environment to accumulate new experience. Learning can also be considered to be part of cognition, but for this thesis its role is made more explicit.

In summary, there is a strong interdependence between learning, cognition and behavior. All three of them should be considered and taken into account when studying intelligence or intelligent behavior in both, biological and artificial systems. This allows to obtain integrative understanding of each component, the ensemble system and ultimately what makes *intelligence* in general. Interestingly, there is a strong conceptual relationship of these three components with the field of computer science and algorithms, which is highlighted and further discussed in 4.4. Within this thesis, I will refer to this afore described relationship by the *Bermuda triangle of intelligence* analogy (fig. 1.1): You can easily get lost, in terms of failing to understand, in between the three components when trying to take a short cut by only considering cognition, behavior or learning individually.

One promising way to study all three components collectively is by developing detailed functional models of the neural systems involved (Abbott, DePasquale, and Memmesheimer, 2016). Unfortunately the complexity and size of these systems strongly depend on the organism under investigation. For primates and even rodents the current technological limitations make it nearly impossible to build detailed integrative systems at this scale. Thus, insects offer a great opportunity as model systems to study. They currently provide the best trade-off between system size and complexity while showing remarkable cognitive skills and complex behavior. Additionally the neural systems and computational mechanisms identified in one species commonly turn out to be general and can be found with only slight variations in other insects. Finally, the current state of technology and genetic tools allows to collect the most precise and comprehensive experimental results in insect nervous systems (Ito et al., 2008; Eichler et al., 2017; Demmer and Kloppenburg, 2009; Inada, Tsuchimoto, and Kazama, 2017; Kazama and Wilson, 2009; Aso et al., 2014; Caron et al., 2013; Szyszka et al., 2014; Egea-Weiss et al., 2018; Shiozaki, Ohta, and Kazama, 2020). Particularly, very recently the neuroscience community has successfully obtained the full connectome of the central brain of *Drosophila melanogaster* (Xu et al., 2020).

## 1.2 Organization of this thesis

The thesis is organized as follows. The next section 1.3 will present and briefly introduce the research objectives pursued in this thesis. In chapter 2 the research papers addressing the specific objectives are included. Chapter 3 contains additional ongoing research work with significant progress that has been conducted as part of this

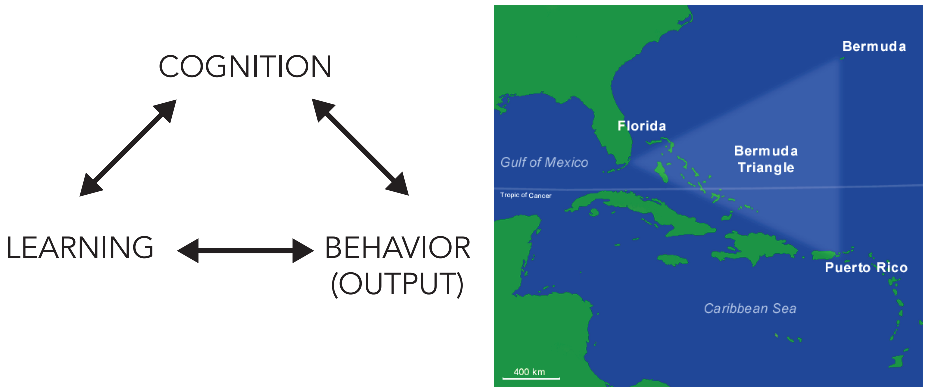


FIGURE 1.1: **Left:** The *Bermuda triangle of intelligence* is used as an analogy within this thesis and refers to the interdependence between cognition, learning and behavioral output. All three components are required for an intelligent system in both, the biological or artificial domain. It illustrates the approach taken by this thesis and the suggestion, that all three should be considered and studied simultaneously to obtain an integrative understanding of each component and what makes *intelligence* in general. **Right:** The physical bermuda triangle [a] of this analogy refers to a geographical region that is associated with the myth of airplanes to disappear. In the context of this thesis this means: You can easily get lost in between (studying intelligence) when trying to take a short cut by only considering cognition, behavior or learning individually.

[a] image source: `https://en.wikipedia.org/wiki/Bermuda_Triangle`

thesis but has not yet matured into a standalone publication. The thesis concludes with chapter 4 where an overall discussion of the higher level concepts that served as inspiration of this work is given and how these concepts contribute towards understanding learning, cognition and behavior in biological and artificial systems. It is further discussed how the research conducted as part of this thesis is embedded within the framework of these broader conceptual ideas and provides an outlook for potential future work along these lines.

## 1.3 Objectives

Based on the motivation introduced above, priority in this thesis is given to the study of sensory processing, learning, cognition and behavioral control in insect nervous systems. This is approached by developing detailed, functional models of sensory processing and learning for cognition and behavioral problem solving as presented in Rapp, Nawrot, and Stern, 2020 and Rapp and Nawrot, 2020. The last publication (Nashef et al., 2017) arose from a research collaboration and provides an outlook on how bigger and more complex systems in primates can be studied by applied machine learning. This approach allows to obtain a statistical understanding of brain function. In particular, the collaborative work in Nashef et al., 2017 is concerned with the functional role of the cerebellar-thalamo-cortical system for motor preparation and execution. Additional research presented in chapter 3 follows the same line of research and proposes a machine learning method based on artificial neural networks for anomaly detection of seizure-like events in EEG time-series data.

The research conducted in this thesis pursues the following three main objectives:

1. Computation with single neurons, synapses & action potentials for cognition and as an alternative to statistical learning.

2. How can learning shape dynamical processing of sensory information and motor control to solve complex behavioral problems in small-brained animals.

3. Applied machine learning as a tool to obtain statistical understanding of large scale brain systems.

In what follows, each objective and how it is addressed by and conceptually related to the included publications in this thesis is briefly summarized. For details and specific results the reader is kindly referred to the specific publication listed in chapter 2.

### 1.3.1 Cognition and machine learning with single spiking neuron computations.

The first objective is addressed and studied in Rapp, Nawrot, and Stern, 2020. This work introduces two different types of numerical cognition tasks. The first task is motivated from a machine learning perspective, where the objective is to count the number of instances of a specific visual concept that are present in a random set of related concepts. Specifically, the data set of handwritten MNIST (LeCun and Cortes, 2010) digits is used to generate $50x50$ pixel images. Each image is divided into a $3x3$ grid and each grid location contains a single random (with replacement) MNIST digit between 0 and 9 (see fig. 1.2). The learning objective is to count the number of instances of the digit "1" that appear within such a single image with occurrences ranging from 0 to 5. The entire image is presented as input to a spiking neuron model after being transformed into a parallel spike train by a non-plastic model of an early visual system. An improved implementation of the Multispike Tempotron (Gütig, 2016) learning rule is introduced and used to adjust the spiking neuron's synaptic weights, such that the number of generated action potentials are equal to the count of digits "1" in the input image. In Rapp, Nawrot, and Stern, 2020 it is shown, that the computational power of a single spiking neuron is strong enough to learn to successfully solve this task with decent accuracy. To compare the performance, a conventional Deep Learning model is trained on the same task, a convolutional neural network (ConvNet). The results show, that the single neuron model achieves satisfactory performance, while using much less resources in terms of number of training samples and training epochs. Furthermore, the work shows that the single neuron model is able to generalize to out-of-distribution (OOD) samples well above chance level while the ConvNet fails to do so. This can be achieved because the spiking neuron implicitly performs representation learning. The term OOD refers to samples that have 0 probability under the training data. A pitfall that is shared by all current machine learning methods that are based on statistical learning theory and maximum likelihood estimation (MLE). To this end, additional images are generated that contain 6 instances of digit "1" while the training set only contained images with 0 to 5 instances. The differences between statistical learning and representation learning are further discussed in 4.5.

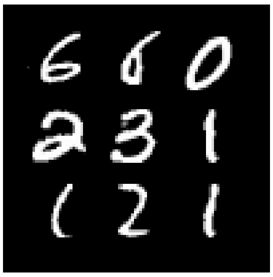


FIGURE 1.2: Each input image shows 9 random MNIST digits positioned on a 3*x*3 grid. MINIST digits are drawn randomly with replacement, such that multiple instances of the same digits can be present within a single image. The learning objective is to count the number of instances of the digit "1". In this case, a single spiking neuron is trained to represent each occurring instance of "1" by a single action potential. In the above example the neuron should elicit 3 action potentials to correctly classify this image when presented as input to the neuron.

For the second task a biologically motivated numerical cognition problem is introduced and solved by using the same single neuron model and learning rule. However, differently to the machine learning problem above, biological organisms usually cannot consume the entire input (here a single image) at once due to limitations of their sensory systems, e.g. limited field-of-view. The introduced task follows the study of Howard et al., 2018, where honeybees have been trained to perform a *greater-than* dual choice task. Honeybees are presented with 2 images. Each image shows between 1 to 6 items (geometric shapes, circles, squares, diamonds). The shapes within an image are consistent but vary in size. The honeybees have to choose the image which is greater (showing more items) than the other by landing on a reward located in the center of the image which contains the greater number of items. Due to their limited field-of-view (FOV), honeybees employ a sequential inspection strategy to scan the entire image by hovering over the image at a distance of $\sim$ 2cm. This translates the overall cognition problem into a temporal cognition and memory problem, where the honeybees have to use working memory to keep track of number of recognized shapes over time (see fig. 1.3). This behavior can be related to a family of algorithms in computer science, known as *divide-and-conquer*. The duality of behavioral strategies and algorithmic problem solving in computer science is further discussed in 4.4. The overall problem is divided into smaller, easier to solve sub-problems (here: brightness change in a single FOV image) and the results are recombined to solve the original problem (here: detecting the number of items).

To solve this task with the single spiking neuron model, the same approach is taken as in the counting MNIST task. Instead of a single static image, now a single input consists of a sequence of multiple FOV images sampled sequentially from the original stimulus images (see Fig. 1.3). Each FOV image is transformed into a parallel spike train using a model of an early visual system. All these individual parallel spike trains are then concatenated in time into a single long parallel spike train. This spike train is fed into the spiking neuron model which is then trained to match its number of output spikes to the number of geometric items present in the original image. This means, that the neuron is trained on the problem of precise counting instead of the "greater than" problem, on which the honeybees have directly been

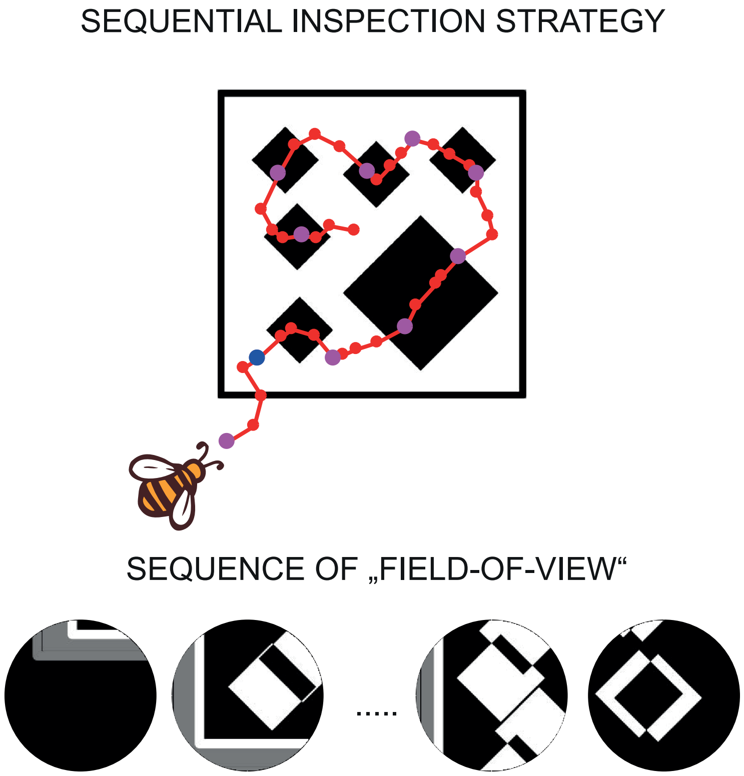


FIGURE 1.3: Honeybees employ a sequential inspection strategy, similar to a divide-and-conquer approach, to perceive an image that is larger than their field-of-view. Such an inspection trajectory (red) results in sampling of the original image in form of a time series of spot-like field-of-view (FOV) images. More precisely, a sequence of derivative FOV images between two subsequent time points $t-1$ and $t$. Using a working memory to keep track of brightness changes over the sequence of FOV images allows honeybees to keep track of the number of items in the overall image.

trained on in Howard et al., 2018. To perform the "greater than" decision, the number of output spikes of the neuron model are compared in response to 2 stimulus images. A decision is correct, if the number of output spikes in response to the stimulus image that shows the larger number of items is greater than the number of spikes generated in response to the second image. The results in Rapp, Nawrot, and Stern, 2020 show, that the single spiking neuron model can successfully solve this complex numerical cognition task and achieves comparable performance as honeybees in the experimental study of Howard et al., 2018. In summary, the study in Rapp, Nawrot, and Stern, 2020 demonstrates that the computational power of a single neuron is strong enough to solve complex cognitive and machine learning tasks. In fact, a very recent experimental study by Gidon et al., 2020 has shown, that dendritic computations of a single neuron can solve the XOR problem, a computational problem that is not linearly separable and thus not solvable by a single rate-based neuron. As a consequence, both studies challenge the question whether artificial rate-based neurons sacrifice too much of their computational power in favor of mathematical convenience. The work in Rapp, Nawrot, and Stern, 2020 further shows, that the choice of sensory (pre-)processing in combination with the choice of a particular learning rule has a strong impact on the success of solving the underlying learning problem. As such, sensory processing and learning should not be considered as two independent problems but rather be studied together in the context of the underlying learning or behavioral problem, as motivated in the introduction of this thesis. This is investigated in more detail in the context of foraging and olfactory sensory processing in the second publication (Rapp and Nawrot, 2020) and additionally discussed in 4.1.

### 1.3.2 Learning & dynamical sensory processing for behavioral control during foraging.

The work in Rapp and Nawrot, 2020 addresses the second major objective of this thesis: How can dynamical sensory processing in the olfactory system be learned and used to generate appropriate motor commands during foraging behavior of a flying insect. The work in Rapp and Nawrot, 2020 first introduces a detailed spiking neural network model of the insect olfactory system as shown in fig. 1.4. This comprises receptor neurons located in the antennae, glomeruli formed by projection (PNs) and local interneurons (LNs) located in the antennal lobe, a population of Kenyon cells in the mushroom body and a plastic mushroom body output neuron (MBON). While the specific model in this work uses the numbers identified by the connectome of *Drosophila melanogaster* (Aso et al., 2014; Takemura et al., 2017; Hige et al., 2015) the general blueprint and its computational properties are homologous across species. The model includes network as well as cellular features that have been identified experimentally (Inada, Tsuchimoto, and Kazama, 2017; Aso et al., 2014; Nagel and Wilson, 2011; Demmer and Kloppenburg, 2009; Wilson, 2013). These biophysical and structural features allow the two major computational mechanisms of this system to emerge, namely temporal and spatial sparse coding (Kloppenburg and Nawrot, 2017). Results of this work show that the proposed model accurately reproduces both computational mechanisms found in-vivo. Next, a classical conditioning task is introduced to perform associative learning in the MBON. Two types of odor stimuli are presented, where one of which is paired with reward (e.g. food) and the other is paired with punishment. The MBON is trained using the same approach as in Rapp, Nawrot, and Stern, 2020 to respond with one action potential to rewarded odor stimuli and with zero action potentials in response to odor stimuli paired with punishment. After training, the output of the MBON is interpreted as a binary behavior response signal, similar to PER (proboscis extension reflex) response in other insects. Results over $N = 100$ independently trained models (e.g. 100 individuals) show rapid behavioral learning dynamics in this differential conditioning paradigm. After $\sim 3 - 5$ presented stimuli substantial learning occours where $\sim 70\%$ of the $N = 100$ models have learned to correctly respond to rewarded stimuli. This few-shot style learning is a hallmark of many insects, for example honeybees (Pamir et al., 2014), but still remains difficult for most computational models. The work proceeds by showing that the MBON is able to perform dynamical memory recall of the learned association during presentation of complex sequences of many cues without additional training. Results further show, that the learned representation is very robust and the MBON can reliably distinguish between cues of distractor odors and cues of the rewarded odor, even if they are very similar to each other. This ability allows to segregate behaviorally relevant odors from background odors which is an important feature for many types of behaviors. Finally the problem of foraging is considered as sketched in Fig. 1.4. The objective is to track down a food source by its emitting odor plume, which is a non-trivial behavioral problem. In natural environments odor plumes break up into thin filaments due to turbulent wind conditions.

As a results the plume forms a complex spatial and temporal odor landscape, where local concentration gradients are not informative. Insects employ a stereotypic *cast & surge* strategy driven by two distinct sensori-motor reflexes (Breugel and Dickinson, 2014). This behavior is characterized by performing crosswind casting trajectories followed by U-turns to locate the plume boundaries. During this behavior, the insect encounters odor filaments as short-lived, discrete events in time. After a few casting trajectories the insect surges upwind until it loses track of the plume and starts over again. From an algorithmic perspective, the problem boils down to accumulation of sensory evidence on short time-scales that can be used to generate appropriate motor commands for U-turn and upwind surge. In the paper is shown, that using the trained MBON to detect brief individual odor cues in time by using dynamic memory recall can generate a neural signal of accumulated sensory evidence. This signal contains all relevant information to generate temporally precise motor commands to execute a cast & surge behavior. The results of this work suggest, that innate foraging behavior can emerge without ongoing plasticity, e.g. by genetically predetermined connectivity or during development. The results further reveal distinct roles of population and temporal sparse coding schemes. While population sparseness is required for successful and rapid learning, temporal sparseness allows for dynamic memory recall to generate precise motor output relevant for behavior. Combining both properties makes it possible to reason about complex composite objects without explicitly learning them. This is achieved by forming memories on simple, atomic sensory cues and evidence accumulation through dynamic memory recall to derive composite signals for higher order processing and reasoning. This is a form of compositionality which is discussed in more detail in 4.7.

### 1.3.3 Applied machine learning as a tool to obtain statistical understanding of large scale brain systems.

In the research conducted in Nashef et al., 2017 the perspective is shifted towards much bigger and complex nervous systems where currently no detailed functional models can be built. This work originates from a research collaboration with the lab of Prof. Yifat Prut at the Hebrew University, specialized in studying cortical areas of primates. It provides an outlook on how statistical and machine learning techniques can be used to study larger scale systems to obtain a statistical understanding of their function, here for example the cortical sub-systems in primates.

The specific system investigated in Nashef et al., 2017 is the cerebellar-thalamo-cortical (CTC) system, which plays a major role in controlling timing and coordination of voluntary movements. As part of this thesis, machine learning methods as well as statistical methods have been applied to analyse multiunit activity (MUA) and local field potential (LFP) responses in the primary motor cortex, the premotor cortex and the somatosensory cortex of primates. The recorded responses are evoked by an implanted chronic stimulation electrode in the superior cerebellar peduncle (SCP). Experimental design, implants and data collection has been conducted by the collaborators Abdulraheem Nashef and Prof. Yifat Prut.

Statistical analysis with principal component analysis is used to estimate the area-specific response properties based on the MUA response shape and time dependent frequency components of the evoked LFP. A machine learning classifier is used to decode each of these signals to predict the cortical site of the evoked responses. Results show a good classification performance between somatosensory and motor related cortical sites when using MUA and LFP signals individually. Discrimination between primary motor and premotor cortical areas is only possible when using combined features from MUA and LFP. When predictions are projected back onto the ground truth topography of the recording sites, results show that the decoding performed by the classifier can recover the functional organization. However, the site-to-site variation in evoked responses due to SCP stimulation within motor cortical areas is considerably large. Overall, the results indicate a substantial impact of SCP on both, primary motor and premotor cortex. Based on documented involvement of these areas in preparation and execution of movement, the results suggest that CTC contributes to motor execution and motor preparation.

Along this line of using applied machine learning, chapter 3 proposes a machine learning method based on artificial neural networks for forecasting, generation and rare event detection of stochastic time-series. The method is named Conditional Probability Echo-State Network (CPESN). The problem of univariate time-series prediction is formalized as the problem of prediction of conditional probability distributions of the form $f(u|x)$. Where the signal value $u(t)$ at time step $t$ is conditioned on some autoregressive history $x$ of previous time steps. In the context of this work, the autoregressive history is implicitly captured by features generated from a random recurrent neural network (RNN) that is driven by the time series as input. The temporal memory of the RNN expands the dimensions of the univariate time series into a state vector of population activity (also called reservoir). This high dimensional population activity captures the dynamics of the time series on multiple different time scales. A multi-layered artificial neural network (ANN) is trained to estimate the complete conditional probability density function $f(u|x)$. This is done by transformation of the marginal density of the univariate time series $u(t)$, namely $f(u)$, into a uniform density $g(s)$ by using the integral transform from probability theory. The transformed signal values an be used to derive training targets for the ANN to perform logistic regression to estimate the transformed cumulative conditional probability function $G(s|x)$. By fitting a smooth b-spline on the ANN outputs and analytical derivation of it, the transformed conditional probability density function $g(s|x)$ can be estimated. By applying the inverse transform and Bayes rule to $g(s|x)$ the estimated full conditional probability density function $f(u|x)$ in the original domain of the signal $u(t)$ can be obtained.

The particular application of this method is concerned with anomaly detection in EEG time-series to detect seizure-like events. Signals containing healthy EEG data are easy to collect and thus large training sets of healthy EEG can be assembled, which are required to successfully train deep learning models in general like the CPESN. Since the CPESN can estimate the full conditional probability density function instead of only a single point estimate, it is possible to compute the probability of the signal's value at each time step. This probability is close to 1 when the signal value follows the regular dynamics of the signal and is low or close to zero when not. Thus, computing such a probability trace can be used to detect rare events within a time series signal, for example seizure-like events.

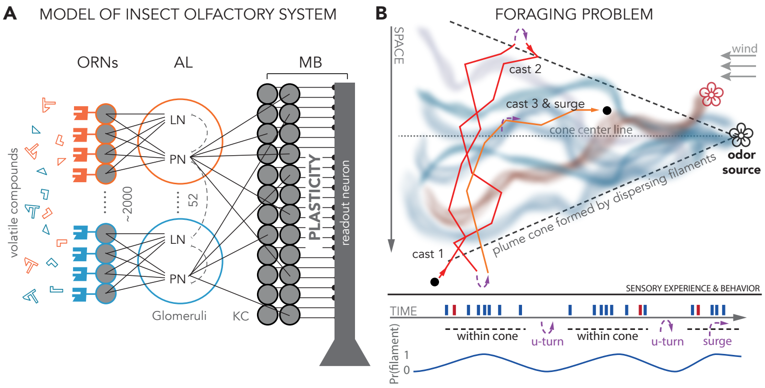


FIGURE 1.4: **A:** Olfactory receptor neurons (ORNs) at the antennae bind and respond to volatile odorant compounds. ORNs expressing the same genetic phenotype project to the same Glumerus in the antennal lobe (AL). Each Glomerulus comprises a projection (PN) and local interneuron (LN). LNs form lateral inhibitory connections among Glomeruli and PNs randomly project to a large population of Kenyon Cells (KC) in the mushroom body where each KC receives input from 6 PNs on average. Sensory processing, learning and memory is performed at the output of the mushroom body (MB) by reading out Kenyon Cells (KC) with a single, plastic mushroom body output neuron (MBON). The overall bauplan of the olfactory system is homologous across species. Here the specific numbers of neurons within each population and connectivity are taken from the connectome of the mushroom body of *Drosophila melanogaster* (Aso et al., 2014; Xu et al., 2020; Inada, Tsuchimoto, and Kazama, 2017). **B:** Sketch of the foraging problem commonly conducted as experiments in a wind tunnel with a pleasant odor source (black flower) and a second distractor source (red flower). Due to turbulence, the odor molecules emitted by the sources form dispersing, intermittent filaments within a cone-like boundary that constitutes the odor plume. A behaving model insect performs stereotypic cast & surge behavior to locate the food source. This is done by scanning crosswind and U-turning after running past the plume cone boundary where no filaments are present. Eventually after several casts (here 3) it surges upwind until it loses track of the plume cone and starts over. Sensory evidence is available through filament encounters during casting trajectories resulting in brief on/off, sequential stimuli of the olfactory system (time axis below). Accumulated sensory evidence can be seen to be proportional to the probability of encountering filaments. This probability is zero outside the plume boundary and increases towards the plume cone's center line (assuming a gaussian distribution of filaments). This signal allows to generate optimal motor commands for U-turn and surge behavior by thresholding and estimating when the derivative turns negative.

# Chapter 2

# Cumulative Publications

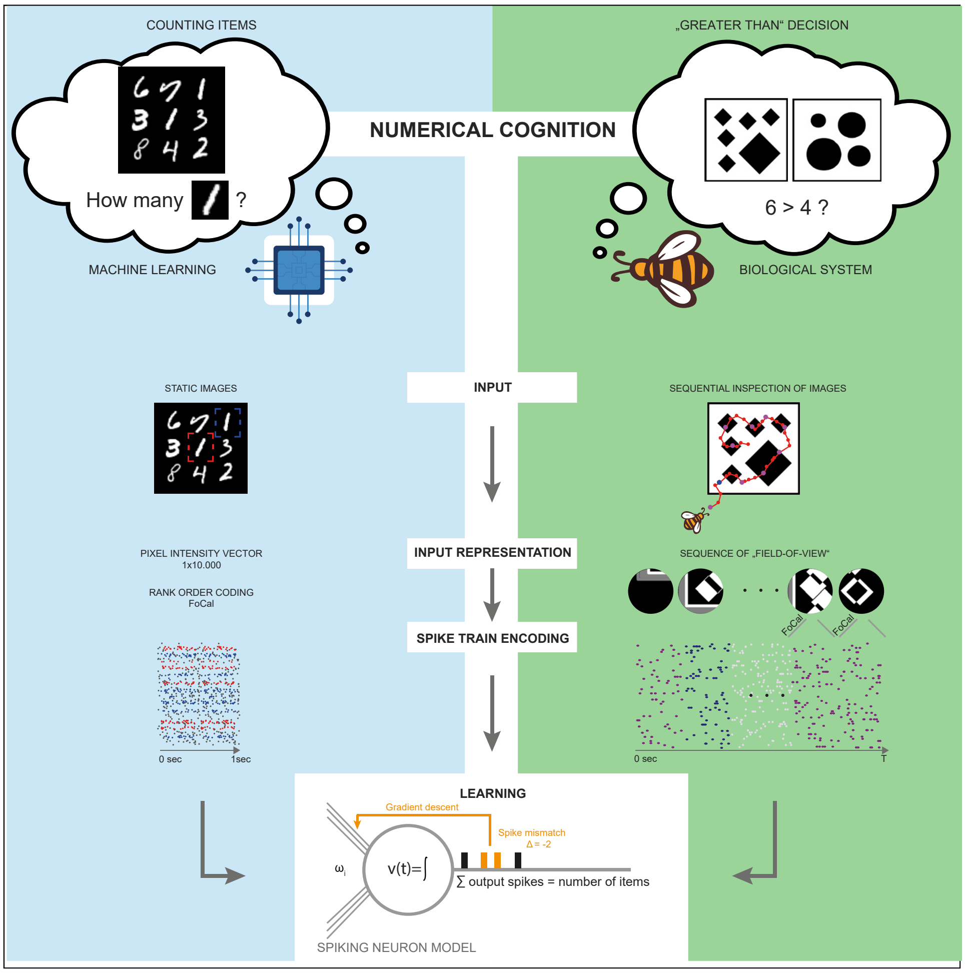
## 2.1 Numerical Cognition Based on Precise Counting with a Single Spiking Neuron.

iScience

CellPress

Article

# Numerical Cognition Based on Precise Counting with a Single Spiking Neuron



Hannes Rapp, Martin Paul Nawrot, Merav Stern

hannes.rapp@smail.uni-koeln.de

## HIGHLIGHTS

A single spiking neuron can successfully learn to solve numerical cognition tasks

The number of action potentials can represent numerosity

Learning to count within few epochs allows generalization to unseen categories

Counting with a single spiking neuron can solve numerical cognition tasks in insects

Rapp et al., iScience *23*, 100852
February 21, 2020 © 2020 The Author(s).
https://doi.org/10.1016/j.isci.2020.100852

iScience

CellPress

Article

# Numerical Cognition Based on Precise Counting with a Single Spiking Neuron

Hannes Rapp,[1,5,*] Martin Paul Nawrot,[1,3,4] and Merav Stern[2,3,4]

## SUMMARY

**Insects are able to solve basic numerical cognition tasks. We show that estimation of numerosity can be realized and learned by a single spiking neuron with an appropriate synaptic plasticity rule. This model can be efficiently trained to detect arbitrary spatiotemporal spike patterns on a noisy and dynamic background with high precision and low variance. When put to test in a task that requires counting of visual concepts in a static image it required considerably less training epochs than a convolutional neural network to achieve equal performance. When mimicking a behavioral task in free-flying bees that requires numerical cognition, the model reaches a similar success rate in making correct decisions. We propose that using action potentials to represent basic numerical concepts with a single spiking neuron is beneficial for organisms with small brains and limited neuronal resources.**

## INTRODUCTION

Insects have been shown to possess cognitive abilities (Chittka and Niven, 2009; Avarguès-Weber et al., 2011,2012; Avarguès-Weber and Giurfa, 2013; Pahl et al., 2013). These include estimating numerosity (Rose, 2018; Skorupski et al., 2018), counting (Chittka and Geiger, 1995; Dacke and Srinivasan, 2008; Menzel et al., 2010), and other basic arithmetical concepts (Howard et al., 2018, 2019). How insects succeed in these cognitive tasks remains elusive. A recent model study by Vasas and Chittka (2019) suggested that a minimal neural circuit with only four rate-based neurons can implement the basic cognitive ability of counting visually presented items. The study implies that their minimal circuits can recognize concepts such as a ''higher'' or ''lower'' item number and ''zero'' (Howard et al., 2018) or ''same'' and ''different'' number of items (Avarguès-Weber et al., 2012) when combined with a sequential inspection strategy that mimics the behavioral strategy of insects during detection (Dacke and Srinivasan, 2008). The neural circuit studied in Vasas and Chittka (2019) was shown to successfully predict whether a particular feature (e.g. yellow) has been presented more or less often than a pre-defined threshold number, despite being presented in a sequence of other features and distractors. This circuit model was hand-tuned in order to successfully estimate numerosity in a numerical ordering task similar to Howard et al. (2018). This poses the question on how an efficient connectivity, which allows the network to estimate numerosity, could be learned by means of synaptic plasticity.

Numerosity estimation tasks that require counting the number of detected instances have also been researched in the field of computer vision, in particular in relation to object recognition tasks. Many resources have been devoted to train artificial neural networks to perform such tasks. Deep learning methods (Schmidhuber, 2015) in particular have been shown to be successful in object detection, and they enable counting by detecting multiple relevant objects within a static scene either explicitly (Ren et al., 2015) or implicitly (Lempitsky and Zisserman, 2010). However, these model classes are costly as they typically need to be trained on a very large number of training samples (in the millions) and often require cloud-computing clusters (Krizhevsky et al., 2012; Simonyan and Zisserman, 2014). Indeed, OpenAI, 2018 recently showed that the amount of computing power consumed by such artificial systems has been growing exponentially since 2012.

Clearly, insects with their limited neuronal resources cannot afford similar costly strategies and hence have to employ fundamentally different algorithms to achieve basic numerical cognition within a realistic number of learning trials. These biological algorithms might prove highly efficient and thus have the potential to inform the development of novel machine learning (ML) approaches.

A number of recent studies managed to train spiking neural networks with gradient-based learning methods. To overcome the discontinuity problem due to the discrete nature of action potentials some

[1]Computational Systems Neuroscience, Institute of Zoology, University of Cologne, Zülpicher Straße 47b, 50923 Cologne, Germany

[2]Department of Applied Mathematics, University of Washington, Lewis Hall 201, Box 353925, Seattle, WA 98195-3925, USA

[3]These authors contributed equally

[4]Senior author

[5]Lead Contact

*Correspondence: hannes.rapp@smail.uni-koeln.de

https://doi.org/10.1016/j.isci.2020.100852

studies evaluated the post-synaptic currents in the receiving neurons for the training procedures (Nicola and Clopath, 2017; Huh and Sejnowski, 2017). Other studies used the timing of spikes as a continuous parameter (Bohte et al., 2000; O'Connor et al., 2017; Zenke and Ganguli, 2018), which led to synaptic learning rules that rely on the exact time interval between spikes emitted by the presynaptic and the postsynaptic neuron. These spike-timing-dependent plasticity (STDP) rules were found experimentally (Bi and mingPoo, 2001) and have gained much attention in experimental and theoretical neuroscience (Caporale and Dan, 2008; Song and Abbott, 2000). Other recent studies approached the problem by either approximating or relaxing the discontinuity problem (Zenke and Ganguli, 2018; Bengio et al., 2013) to enable learning with error backpropagation in spiking neural networks. Training single spiking neurons as classifiers has been proposed by Gütig and Sompolinsky (2006) and Memmesheimer et al. (2014). Closely related, Huerta et al. (2004) trained binary neurons to perform classification in olfactory systems.

Here, we study a biologically realistic spiking neuron model with a synaptic learning rule proposed by Gütig (2016). Our approach to numerical cognition takes advantage of the discrete nature of action potentials generated by a single spiking output neuron. The number of emitted spikes within a short time period represents a plausible biological mechanism for representing numbers. In a virtual experiment we train our neuron model to count the number of instances of digit 1 within a static image of multiple handwritten digits (LeCun and Cortes, 2010). The synaptic weights are learned from the observations, and thus our model overcomes the problem of hand tuning a single-purpose neuronal circuit. We then test the model on the same "greater than" task as in Vasas and Chittka (2019), but we use the model's ability of precise counting to derive the concept of "greater than."

Because in the present work we are interested in estimating numerosity, the teaching signal in our model is a single integer value that is equal to the total number of relevant objects. To achieve successful training we introduce an improvement to the implementation in Gütig (2016) where the membrane potential was considered for gradient-based learning to overcome the spiking discontinuity problem. We show that our improved implementation to this approach allows to train the model with better generalization capabilities and also supports better the reliability of numerosity estimation under inputs with complex distributions, including noise distributions, as naturally present in the brain.

## RESULTS

Our objective is the implementation of a spike-based method that can be trained to solve numerical cognition tasks. We employ the multispike tempotron (MST) (Gütig, 2016), a single leaky integrate-and-fire neuron model with a gradient-based local learning rule. We suggest a modified update rule of the learning algorithm that reduces the variance in training and test error. The model is subjected to three different tasks that progress from a generic spike-pattern detection problem to a biologically inspired dual choice task that mimics behavioral experiments in honeybees.

### Detection of SpatioTemporal Input Spike Patterns

We begin by considering the problem of detecting different events over time. A particular event is represented by a specific spatiotemporal spike pattern across a population of neurons that are presynaptic to the MST. These spike patterns are generic representations of events that could, for instance, represent sensory cues in an animal's environment.

We generated event-specific patterns of fixed duration (1 s) across 500 presynaptic input neurons using a gamma-type renewal process of fixed intensity ($\lambda = 0.89$ spikes per second) independently for each neuron (see Transparent Methods). The MST was presented with an input consisting of a sequence of different patterns on top of a noisy background that was simulated as independent gamma-type renewal processes of either constant or time-varying intensity (see Transparent Methods).

A single input trial is shown in Figure 1A. It accounts for the random occurrence of three different event-specific spatiotemporal spike patterns (in this specific example, each pattern occurring once) as indicated by different spike color and of distractor patterns occurring twice (black spikes). Gray spikes represent the background noise. Generally, for each trial of 10 s duration we randomly drew a number of pattern occurrences and pattern identities from a total of 9 possible patterns (five target patterns and four distractor patterns).

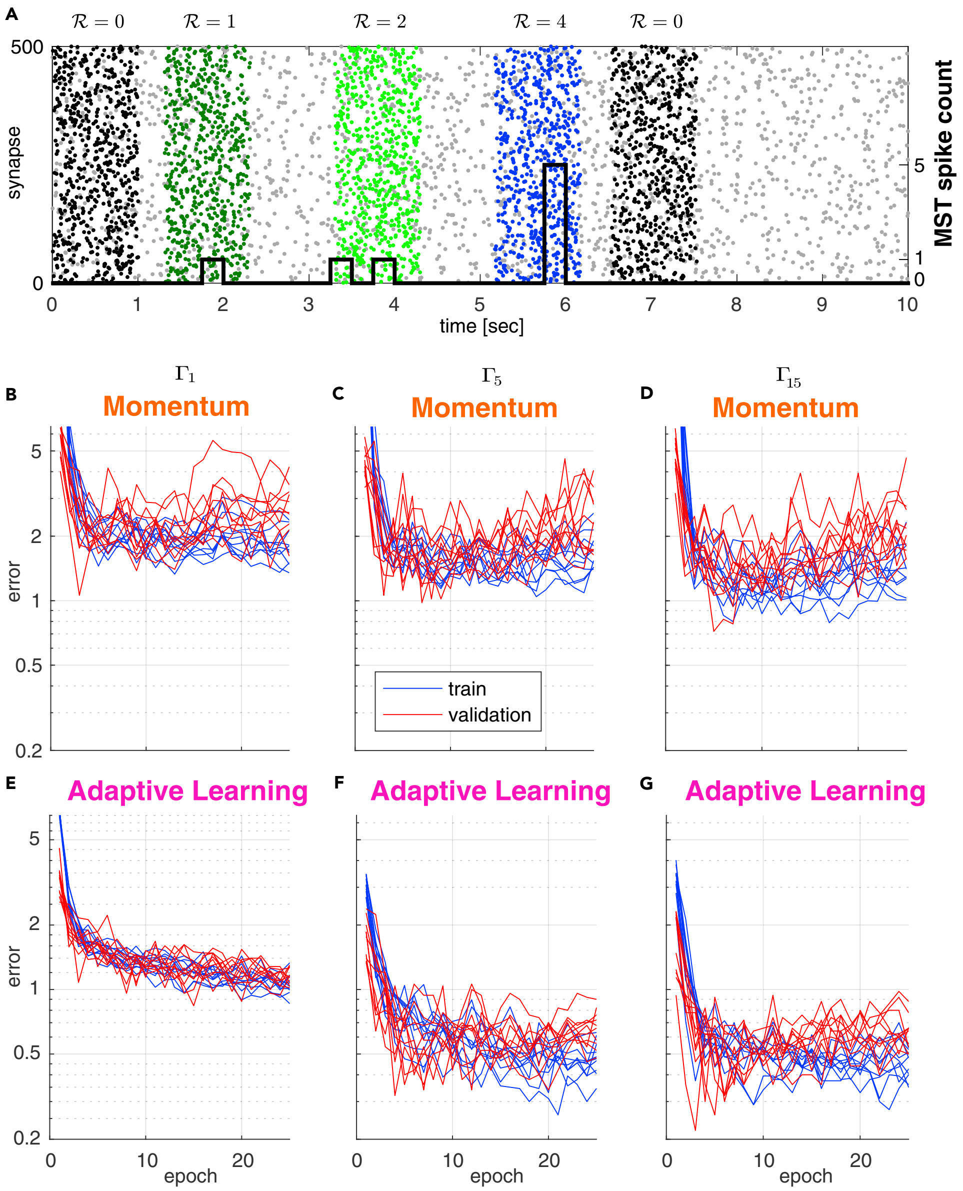


**Figure 1. Comparison of Training Convergence for Momentum and Adaptive Learning under Different Background Noise Conditions**

(A) Sample input sequence: A 10-s-duration spike train input example. The spike train is composed of three patterns, each with a distinct target (dark green, light green, blue), background activity (gray), and two distracting patterns (black). Number of MST output spikes superimposed as black step function. The MST is supposed to fire $\Sigma_i \mathscr{R}_i = 7$ spikes over the whole sequence, $\mathscr{R}=0$ spikes for distractors, and $\mathscr{R}=1, 2$ or 4 for the colored dark green, light green, and blue patterns accordingly. Patterns are simulated with gamma processes of different order (separate datasets): $\Gamma_1$ (Poisson), $\Gamma_5$, and $\Gamma_{15}$. Patterns are superimposed onto 10 s inhomogeneous Poisson background activity.

(B–G) Training curves (blue) and validation curves (red) for 10 independent simulations of the (B and E) $\Gamma_1$ (Poisson), (C and F) $\Gamma_5$, and (D and G) $\Gamma_{15}$ patterns. (B–D) MST with Momentum-based learning implementation (Gütig, 2016). (E–G) MST with adaptive learning implementation. Learning (training) convergence shows larger variance when using Momentum as compared with using adaptive learning. The same is true for the validation (testing) error. This indicates that adaptive learning is capable of finding better optima compared with Momentum.

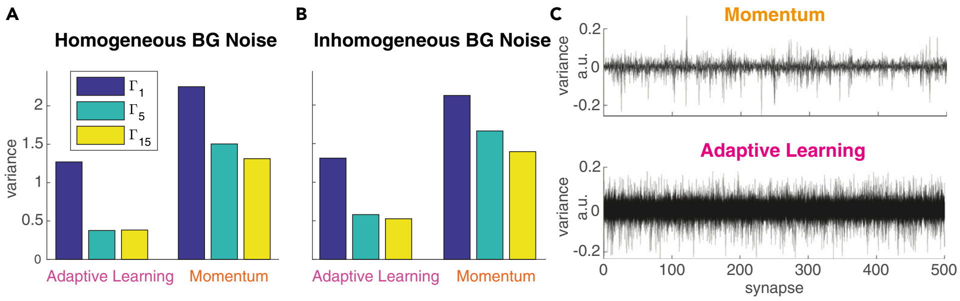


**Figure 2. Training Convergence Properties of Momentum and Adaptive Learning**

(A and B) Variance of validation error measured at epoch 10 for datasets with homogeneous (A) and inhomogeneous (B) background noise.

(C) Empirical analysis of the regularizing effect on the error variance. Weight changes $\Delta\omega_i$ over all training steps (and all epochs) are collected for each synapse $\omega_i$. PCA is performed to reveal which synapses' weight changes show the largest variance over the training process. Large variance in $\Delta\omega_i$ implies strong modification of a synapse. For both Momentum (top) and adaptive learning (bottom) the first 10 principal components are shown where $x$ axes correspond to the synapses $\omega_i$ and $y$ axis shows variance in total weight change per synapse $\omega_i$. The Momentum method tends to tune only a small subset of the available synapses strongly, whereas the adaptive learning method leads to modifications that are more uniformly distributed over all synapses and more broadly distributed in magnitude.

We first trained the original MST of Gütig (2016) to detect pattern occurrence. To each of the five event-specific patterns we assigned a specific target number of MST output spikes $\mathscr{R}$ (from 1 to 5). We did not assign a target to any of the distractor patterns (i.e. the MST was expected to produce zero output spikes in response to a distractor pattern). At the end of each training trial (one sequence of multiple patterns and distractors) the sum of actual output spikes was evaluated and compared with the desired number of output spikes determined by the trial-specific random realization of the input pattern sequence. The absolute difference between the desired and the actual spike count determined the training error in the range of $0 - N \in \mathbb{N}_+$. If the actual number of spikes was larger than the sum of the desired target spikes by some $\Delta_k$, a training step of the MST was performed toward decreasing its output spikes by the difference $\Delta_k$. Similarly, if the actual number was smaller than the sum of desired target spikes, a training step was performed to increase the MST's number of output spikes by $\Delta_k$. No training step was performed for correctly classified samples.

To analyze model performance we computed the training error and validation error for up to 25 training epochs (see Figures 1B–1D). Each training epoch consisted of a fixed, randomized set of 200 trials, and the validation set consisted of 50 trials. Both training error (blue) and validation error (red) dropped sharply with increasing number of training epochs and reached a plateau at about two spikes after ~10 epochs, independent of the type of the gamma-order used for pattern generation (Figures 1B–1D).

## Local Synaptic Update Method Improves Performance and Robustness

Training and test errors exhibited a high variance across repeated models (Figures 1B–1D), indicating limited robustness of model performance. We therefore replaced the Momentum method for gradient descent implemented in the original work of Gütig (2016) by a synaptic specific adaptive update approach similar to RMSprop as proposed by Tieleman and Hinton (2012) (see Transparent Methods).

Although speed of convergence is similar when using the adaptive learning method compared with Momentum, we find that using adaptive learning results in less variant training error (Figures 1E–1G). This also holds for the variance of the test error on an independent validation set indicating better generalization capabilities to previously unseen inputs (Figures 1E–1G, 2A, and 2B). The adaptive, per synapse learning rate combined with exponential smoothing over past gradients has a regularizing effect and prevents the model from overfitting to the training data. We further conclude that the modified algorithm is potentially able to find better and wider optima of the error surface as compared with learning with Momentum. More importantly, this behavior is consistent and independent of the spike-generating process and noise level (Figures 2A and 2B).

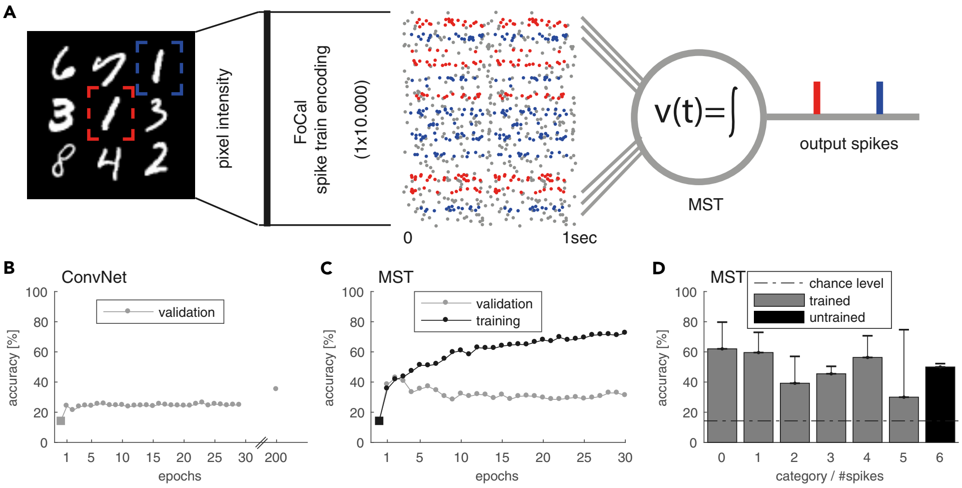


**Figure 3. Counting of Visual Concepts with Spikes**

(A) Sketch of counting task. The goal of this task is to count the number of occurrences of digit 1 in an image of random MNIST digits. Example image (50×50 px) with multiple random digits from the counting MNIST dataset positioned within a 3×3 grid. The image is encoded into parallel spike trains by applying FoCal encoding, resembling a 4-layer early visual system with rank-order coding. The multivariate spike train converges onto the MST via 10.000 synapses. The MST is trained to elicit exactly *k* output spikes where *k* is equal to the number of digit 1 occurrences in the original image (here 2).

(B) For reference we trained a ConvNet on the same raw images. Shown is the performance in terms of mean accuracy (five-fold cross-validation). After 200 training epochs the ConvNet reached ~40% accuracy.

(C) Performance of the MST in terms of mean accuracy (five-fold cross-validation). The MST shows rapid learning reaching a similar level of accuracy as the ConvNet after 200 training epochs within only two to four training epochs.

(D) Mean accuracy +std for the possible numbers of digit 1 present within a single image (categories). The MST is trained on samples of categories 0–5 to generate 0–5 output spikes respectively. The MST is then tested on the untrained category 6 and is able to generalize reasonably while the ConvNet, by design, cannot make predictions for this category.

At this point we cannot provide a theoretically grounded explanation for the regularizing effect we see when using adaptive learning instead of Momentum. Development of theoretically grounded explanations of the effects of different gradient-decent optimizers is a very recent and active research field in the Deep Learning community. To provide insights for the regularizing effect we therefore conducted an empirical analysis of the weight updates, as shown in Figure 2C. Specifically, we performed PCA on the weight changes $\Delta\omega_i$ applied to all synapses over all training steps. The intuition here is that large variance in $\Delta\omega_i$ implies strong modification of a synapse over the training process. Results of our analysis (Figure 2C) show that for the adaptive learning method the weight changes are more uniformly distributed over all synapses and more broadly distributed in magnitude. In contrast, with the Momentum method only a small subset of synapses is strongly modified. We conclude that distributing the updates uniformly over all synapses leads to a more deterministic convergence behavior toward good minima in the error surface, independently from the initial, random initialization of $\omega_i$. The results shown are obtained from a specific choice of meta-parameters ($\alpha = \gamma = 0.99$, $\lambda = 0.01$), but we verified that it remains true over a broad range of possible values and combinations.

Moreover, we find that adaptive learning improves absolute performance converging to a smaller error independent of the actual gamma process when using the same values for the free meta-parameters for both methods. Although choosing different values for the meta-parameters results in different (and in some cases even lower) train and validation errors, our main result regarding the variance still holds. For subsequent tasks we used the MST with adaptive learning.

## Counting Handwritten Digits

We apply the MST model to the problem of counting the number of instances of digit 1 within an image showing several random handwritten MNIST digits (LeCun and Cortes, 2010). The digits are randomly positioned within a fixed 3×3 grid (Figure 3A). Each image can contain between zero and six instances of the digit 1 at one of the nine possible grid locations. To solve this problem with the MST we take the 50x50px

| Counting MNIST Results: Counting Ones | | | |
|---|---|---|---|
| Model | #Parameters | RMSE (Mean ±std) | Accuracy (Mean ±std) |
| ConvNet (5 epochs) | 11,079 | 1.70 ± 0.2083 | 23.00 ± 0.1746 |
| ConvNet (100 epochs) | 11,079 | 1.67 ± 0.3130 | 27.07 ± 0.5113 |
| ConvNet (200 epochs) | 11,079 | 1.02 ± 0.0768 | 40.97 ± 0.8136 |
| $\text{MST}_{\text{adaptive}}$(~4 epochs) | 10,000 | 1.21 ± 0.2067 | 47.72 ± 3.2052 |

**Table 1. Results for MNIST Digit Counting MNIST Task**

We evaluate each model in terms of root-mean-square error (RMSE) of the difference in actual and predicted number of digits (a lower RMSE indicates a better performance) and accuracy of correct digit count in images. Reported results are mean and standard deviation over a five-fold cross-validation.

input image and encode the entire image as a parallel spike train. To transform the image into a parallel spike train that can be fed into the MST model we use filter-overlap correction algorithm (FoCal) of Bhattacharya and Furber (2010). This method is an improved four-layer model of the early visual system using rank-order coding as originally proposed by Thorpe and Gautrais (1998). We then train the MST model to count the number of occurrences of digit 1 by generating one output spike for each instance of digit 1 (Figure 3A). We train the MST on targets 0–5 using five-fold cross-validation on 400 sample images. The learning rate is tuned manually to $\lambda = 0.00002$, which yields the best performance and training speed. For reference we compare the performance of the MST with a conventional computer vision model that uses a convolutional neural network (ConvNet) (Krizhevsky et al., 2012; Seguí et al., 2015; Fomoro, 2017; Kingma and Ba, 2014; Yu and Koltun, 2015). The ConvNet is trained similarly but provided 800 training samples and a larger learning rate of 0.01 to speed up the training process.

Counting, as a conceptual problem, is similar to a regression problem where we have no a-priori knowledge of the maximum number of desired targets present in an input. It is important to note that the ConvNet model used for comparison is built using prior knowledge about the distribution of the training set. The ConvNet is constrained to learn a categorical distribution over [0,5], where 5 is the maximum possible count of desired digits in the used training set of images. This has two implications. First, the ConvNet model will be unable to predict images that include more than five targets. However, in general for regression problems, the prediction targets are usually not bounded. Second, the counting error a ConvNet can make is constrained by the training bound, i.e. the maximum error is 5. In contrast, the MST model does not have any need for this prior knowledge or constraints. In principle it is capable of solving the general, true regression problem and can (after being trained) also make predictions for images that contain more than five occurrences of digit 1. It thus has to solve a more difficult learning problem. The maximum prediction error in this case is unbounded rendering the MST more vulnerable to prediction errors compared with the ConvNet. Figure 3B shows the performance of the ConvNet in terms of mean accuracy of correctly counted images. Despite the large learning rate, accuracy only slowly (but monotonically) improves over the course of 200 training epochs. In contrast, the performance of the MST in Figure 3C shows rapid learning, reaching similar mean accuracy as the ConvNet within only ~3 training epochs. The MST reaches a performance above chance level for each of the trained target categories 0–5 (Figure 3D). It also performs above chance level for images that contain six targets. This indicates that the MST is not only learning a categorical distribution over 0–5, as is the case for the ConvNet but also generalizes to a larger, previously unseen number of targets. We want to emphasize that the MST performs better than the ConvNet despite the advantages given to the latter in the form of a larger number of training samples and a higher learning rate. The results are further summarized in Table 1.

During our experiments we found that the choice of the spike encoding method has a big impact on the MST's performance. It is possible that, by applying better or more efficient encoding algorithms, the performance of the MST model can be further improved.

## Insect-Inspired Numerical Cognition During Visual Inspection Flights

We now consider a biologically motivated task following Vasas and Chittka (2019) and the original experiment conducted in honeybees by Howard et al. (2018). The objective in this experiment is to perform a "greater than" dual choice task on two stimulus images that show varying numbers of geometric shapes

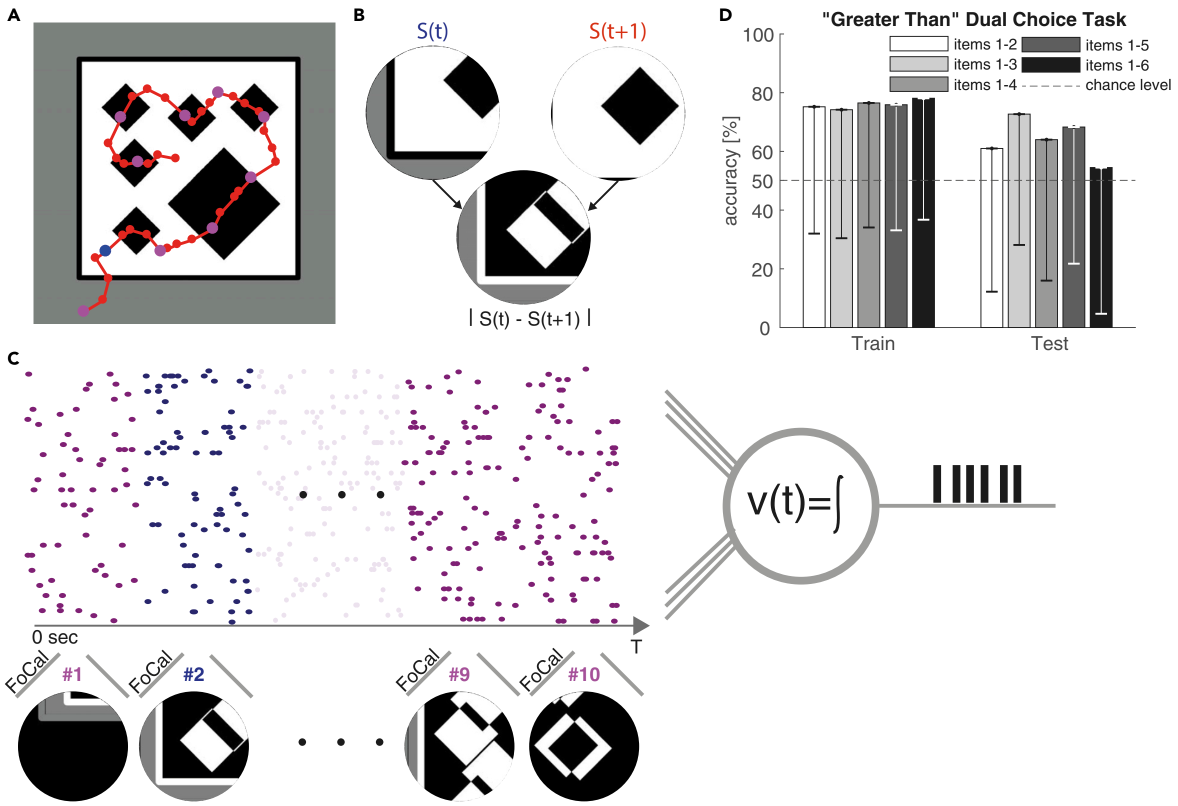


**Figure 4. Dual Choice "Greater than" Task Performed on Geometric Shapes Using a Visual Inspection Strategy Observed in Honeybees**

(A) Sample stimulus image with six diamond shapes and inspection trajectory (red) of a honeybee. The trajectory is sampled at 40 points [Vasas and Chittka (2019)] (all dots on the trajectory) and sub-sample at 10 points for the MST (purple and blue dots).

(B) Field of view (FOV) $S(t)$ and $S(t{+}1)$ of the honeybee during its inspection trajectory (at the blue dot and its subsequent red dot on the trajectory in (A), accordingly). Following the method of Vasas and Chittka (2019), input to the model is constructed as a derivative of the two subsequent FOV images: $FOV_{diff} = |S(t) - S(t+1)|$.

(C) Sequences the $FOV_{diff}$ are encoded into spatiotemporal spike patterns using rank-order coding (FoCal) and concatenated (without gaps) into the resulting parallel spike train. The MST is trained to match its number of output spikes to the number of geometric items in the original stimulus image shown in panel.

(D)Performance in the "greater than" dual choice task. The MST output (number of spikes), $\hat{y}_1, \hat{y}_2$ in response to two different stimulus images with number of items $y_1$,$y_2$, accordingly, is used and compared. When ($y_1 < y_2$)∧($\hat{y}_1 < \hat{y}_2$) the decision is considered correct (and vice versa for $y_1 > y_2$, for $\hat{y}_1 = \hat{y}_2$ a random decision was taken). Bars show mean accuracy − std and grouped by increasing maximum number of items present per image. Our results indicate that the MST can achieve mean accuracy that is comparable to that of honeybees reported in Howard et al. (2018).

(circles, squares, diamonds). The geometric shapes within a stimulus image are consistent, and the possible number of them range from 1 to 6.

In contrast to our previous task, here a stimulus image is not presented as single static input. Instead the input is a sequence of smaller images that mimic the 60° field-of-view (FOV) of honeybees hovering over the stimulus image at a distance of 2 cm (see Transparent Methods). The available corresponding dataset that consists of stimulus images and corresponding inspection flight trajectories recorded from behaving honeybees is highly imbalanced and limited to a total of 97 images. Figure 4A shows an example stimulus image with six diamond shapes and the inspection trajectory taken by one honeybee. This particular trajectory yields a sequence of ~40 FOV images (red dots). Following the same procedure as Vasas and Chittka (2019), the absolute value of the derivative $|S(t) - S(t+1)|$ of two subsequent FOV images $S(t), S(t+1)$ is computed as input to the model (see Figure 4B). To reduce computational cost for our MST model and to unify the varying sequence length across all stimuli, we sub-sample the trajectories to length 10 (magenta dots). In Vasas and Chittka (2019) a rate-based model was used, and the FOV images were encoded into a univariate time-series (representing a rate) that is fed into the model as a single presynaptic input. Because our MST is a spiking model we have to encode each FOV image into a spike train. We apply the same encoding strategy as used before in the counting MNIST task: each FOV image is encoded as a parallel spike train of 10,000 synapses using FoCal. All encoded FOV images are combined into a long parallel spike train by concatenation (see Figure 4C).

The task is divided into two steps. The MST counts the number of geometric items present in a stimulus image. The resulting count numbers are then compared to solve the "greater than" dual choice task. This differs from the original behavioral task by Howard et al. (2018) in which the honeybees were directly trained on the "greater than" decision rather than on precise counting.

To this end we trained the MST (using 10-fold cross-validation) to match the number of generated output spikes to the number of geometric items present. To evaluate the dual choice task we took two random stimulus images with a different number of items $y_1$,$y_2$ and fed these images as input into the trained MST. We then compared the true item count with the predicted item count $\widehat{y}_1, \widehat{y}_2$ of the MST. If $(y_1 < y_2) \wedge (\widehat{y}_1 < \widehat{y}_2)$ it was considered a correct decision and vice versa when $y_1 > y_2$. For undecidable cases where $\widehat{y}_1 = \widehat{y}_2$ a random decision was taken. This sampling process of decisions was repeated for 1,000 iterations. Our results (Figure 4D) show that the MST model is able to achieve comparable performance to the average performance of the honeybees (60%–70%) in the original task of Howard et al. (2018) in terms of mean accuracy of correct decisions. We want to emphasize that the MST performance could be achieved despite the very small and imbalanced training data. Moreover, the MST is trained on the problem of precise counting that is harder than the binary decision task. Although we have to acknowledge that the results show large error bars (due to the very limited training data), we conclude that our results provide a successful proof-of-concept. Using a larger and more balanced training set and better feature encoding would certainly reduce the variability and further improve the performance.

## DISCUSSION

### Counting as a Basis for Numerical Cognition

Numerical cognition is a general term that covers several sub-problems, for example numerosity, counting, relations (greater/smaller than), basic arithmetical operations, and many more. Although each individual sub-problem might appear fairly trivial to us as humans, it is yet not clear how this could be realized computationally on the level of spiking neurons or networks thereof. Despite the simplicity of these sub-problems they do provide a foundation for more complex concepts that humans make heavy use of and are relevant for behavioral decision making. For example, if one is able to count entities it might only be a small step toward combining that information to perform more advanced concepts such as empirical statistics and estimating (discrete) probabilities. Although the specific symbolic math concepts are unavailable to animals, they are still able to show basic numerical cognition and evaluate basic probabilities (Howard et al., 2018, 2019; Avarguès-Weber et al., 2012).

A first objective of the present work was to study whether a single neuron model has the computational power to support numerical cognition tasks. Specifically, we addressed cue detection and counting by handling neuronal input such that it generates an output spike count that matches the number of relevant cues in its input. In order to achieve this computationally, the presynaptic weights of the neuron need to be tuned. Given the fact that the parameter space is very large and many possible solutions may exist, manually tuning the parameters is usually not possible. It is therefore desirable to implement a plasticity mechanism that allows the neuron to tune its weights by learning from examples.

In this work we have explored the MST developed by Gütig (2016). This spiking neuron model can be trained by gradient-descent to produce a precise number of output spikes in response to multiple occurrences of patterns embedded in the presynaptic input. Different patterns are assigned to different target numbers of output spikes per pattern occurrence. Gütig (2016) showed that the MST can learn to detect different spike input patterns. It can further assign the correct number of output spikes matching the targets of individual patterns. The MST learns this despite the fact that the teaching signal is only provided as a single scalar value that is equal to the sum over all targets presented sequentially in the input. Departing from the homogeneous Poisson process studied in Gütig (2016), we confirmed MST performance for biologically more realistic (Farkhooi et al., 2011; Mochizuki et al., 2016; Nawrot, 2010) gamma processes as generators for input patterns on non-homogeneous background.

## Adaptive Local Learning Rule Benefits Model Robustness

We introduced a modification to the update rule of synaptic weights $\Delta\omega_i$. The *adaptive learning* introduces a dynamic, synapse-specific learning rate whose value at training step *t* depends on its history of values from previous training steps. We find that this modification allows the MST to learn a parameter set for the synaptic weights that shows less variability of the training and validation error as compared with the original *Momentum* method used in Gütig (2016). Low variability in validation error is generally a desired property for any learning algorithm because this commonly implies low variability in prediction or classification performance on new, unseen data.

At this point we are unable to provide a theoretically grounded explanation of the regularizing effect shown by *adaptive learning*. The deep learning community currently still lacks theoretical understanding of the effects of different gradient-descent optimizers, which is actively researched (Choromanska et al., 2015; Jin et al., 2017). We performed an empirical analysis of the weight changes $\Delta\omega_i$ over the course of training. Specifically, we used PCA to analyze the variance of $\Delta\omega_i$ for each synapse over all training steps. Our analysis reveals that for the *adaptive learning* a large number of weights are affected. In contrast, when using the originally proposed *Momentum* method, a much smaller subset of synapses show significant weight changes, and their distribution appears much more heavy-tailed with strong weight changes in few neurons. We conclude that modifying all synapses more uniformly appears to increase the likelihood that training converges toward good minima, independent from the initial random choice of $\omega_i$.

## Spike-Based Biological Learning versus Rate-Based Machine Learning

A second objective of this work was to explore possible advantages and disadvantages of a spike-based learning algorithm in comparison to a state-of-the-art deep learning architecture. Biological learning mechanisms enable animals to learn rapidly in a complex and dynamic environment. Instances where sensory cues coincide with reward or punishment during exploration may be sparse, i.e. they have to learn on very small sample sizes and slow learning could have fatal consequences. Single-trial learning, for instance, seems to be a fundamental ability found in different animals. Insects, for example, are able to form long-lasting associative memories upon a single coincident presentation of a sensory stimulus and a reinforcing stimulus (Pamir et al., 2014; Scheunemann et al., 2013; Zhao et al., 2019). Increased learning speed generally comes at the cost of increased generalization error and thus learning speed and high accuracy are in trade off.

We compared the biologically inspired spike-based learning algorithm of the MST with the deep learning architecture of a convolutional neural network, a standard computer vision model (ConvNet). We found that the MST is able to rapidly learn this task within ~3 training epochs of 320 samples each to achieve a mean accuracy of ~47% of correctly counted digits. Additional training did not improve accuracy. Conversely, the ConvNet, despite a 1000× larger learning rate and 100% more training samples per epoch, required >200 training epochs to achieve a similar accuracy. With additional training, the ConvNet achieved >80% accuracy for >5000 training epochs (not shown). Our results reflect a trade-off between very fast but less accurate learning with the spike-based MST method versus slow but increasingly accurate learning with the ConvNet. An additional aspect of biological relevance is the consumption of (computing) resources that are considerably higher for the ConvNet than for the single neuron MST. It is possible that in nature processing with spikes is generally more energy efficient, an important constraint in living organisms (Levy and Baxter, 1996; Niven and Laughlin, 2008; Niven, 2016).

Once trained, the ConvNet is only able to learn a categorical distribution over a fixed set of possible targets (here 0–5) that needs to be put into the design of the model a-priori. Similarly, previous related work of gradient-based learning in spiking network models is mostly concerned with solving classical classification tasks with pre-defined classes (Zenke and Ganguli, 2018; Bohte et al., 2000; Gütig and Sompolinsky, 2006; Memmesheimer et al., 2014). In this work we applied the single-neuron MST model to solve a regression problem. We show that the MST model does not have the limitation of the ConvNet. After being trained on targets 0–5 it was able to generalize to previously unseen images that contained digits 1 at 6 out of the 9 possible positions. This indicates that, in principle, the MST can solve full regression problems.

Differently from all other tasks presented in this work, the difficulty in this task is that each input stimulus is presented as a whole and not sequentially. This means that spike patterns associated with each occurring instance of digit 1 are distributed spatially (over different sets of synapses) instead of temporally. Due to the random positioning within the 3×3 grid, the patterns to be identified by the MST "*jump*" over different sets of synapses for different stimulus images that share the same training target make this task particularly hard to solve.

### Relational Operation Based on Counting

In the final task we went one step further and studied (precise) counting, as it allows other basic numerical cognition tasks to emerge. Assuming that a single neuron can count by relating the sum of its output spikes to the number of items present in a single stimulus, we show that this allows solving other numerical cognition tasks.

To this end we use a biologically motivated "greater than" dual choice task, performed by honeybees that employ a sequential inspection strategy. Honeybees are presented with stimulus images that show 1–6 different geometric shapes. Given two different stimulus images, the bees have to decide which of the two images contain more geometric items. Due to their limited FOV, the bees cannot perceive the stimulus image as a whole. Instead they perform a sequential inspection strategy, by hovering over the entire stimulus image. This results in a time series of FOV images, similar to a moving spot light. Using the MST, we approach this problem similarly and present a long parallel spike train that contains a sequence of FOV images.

In contrast to the honeybees the MST is trained to perform precise counting of the geometric shapes, similarly to the counting task we presented earlier. To perform the "greater than" dual choice task we present two different stimulus images and compare the number of output spikes of the MST. We show that the MST is able to achieve average success rates in terms of correct decisions that are comparable to those achieved by honeybees in the original experiment. Although our results do show much larger error bars than the honeybees, this is due to the following important difference that needs to be considered: the bees are explicitly trained on the (binary) "greater than" task. To the contrary, the underlying problem that the MST solves here is *precise counting*, which is harder to solve in general. Differently from Vasas and Chittka (2019), where input is provided as a univariate rate signal, our model uses parallel spike trains as input, which are derived from the FOV images. Although Vasas and Chittka (2019) used handcrafted features based on the assumption that the number of step-like changes in global image contrast is proportional to the number of scanned items, the MST has to learn which features are relevant and hence are useful to extract from the spatiotemporal input spike patterns. The MST has been trained on this task with a small dataset of ~70 samples per trained model. Increasing the training data will very likely result in better and more robust performance as well as smaller error bars.

### Conclusion

Action potentials represent an elemental discrete quantity used for information processing in nervous systems. We conclude from our study that action potentials produced by a single spiking neuron can support basic arithmetic operation of counting. The MST is a powerful single-neuron method that can be trained to solve regression problems on multivariate synaptic input. We successfully applied the MST to perform basic numerical recognition tasks on complex and noisy input. We suggest that using spikes to represent numerosity with a single neuron can be a beneficial strategy especially for small-brained organisms, which economize on their number of neurons.

### Limitations of the Study

The MST learning rule used in this work requires differentiation of the membrane potential (see Transparent Methods), which is considered to be biologically implausible. Gütig (2016) suggested an approximate formulation of the learning rule that uses correlation-based learning of presynaptic spikes and postsynaptic voltage, which is considered biologically plausible. In order to ensure comparability of our results with the results in the original work we here used the gradient-based learning rule that was evaluated in Gütig (2016) for all the experiments presented. Although in this work we specifically focused on the computational capabilities of the single-neuron model, the same model and learning rule could also be used to construct more complex and layered networks as shown in Gütig (2016). We leave the study of multiple interconnected MST neurons for future research.

One weakness of the MST identified in the course of our study is a tendency for overfitting. This can, for instance, be inferred from the insect-inspired numerical cognition task, where the MST can be trained to reach >80% in accuracy of precise counting on the training set, but performance on the test set remains low or even drops below chance level (data not shown). This indicates that the MST tends to learn more about the samples of the training set compared with learning features that would generalize to the test set. This, to some extent, is also the case for the MNIST task. A potential solution to this problem could be to introduce explicit regularization terms in the MST learning rule, similar to approaches realized in deep learning algorithms. During our experiments we further found that the choice of method for the multivariate spike encoding of images has a big impact on learning and prediction performance. The particular encodings we have tried (using the same datasets) are encoding the intensity of each pixel independently by a 1-s-long spike train, generated by a renewal process with mean intensity equal to the pixel's intensity. This does not produce the ideal robust spike patters that can be learned by the MST. We predict that improved or more sophisticated spike-encoding methods will boost performances.

## METHODS

All methods can be found in the accompanying Transparent Methods supplemental file.

## DATA AND CODE AVAILABILITY

To support further research we make our code and datasets publicly available at Rapp and Stern (2019).

## SUPPLEMENTAL INFORMATION

Supplemental Information can be found online at https://doi.org/10.1016/j.isci.2020.100852.

## ACKNOWLEDGMENTS

This work has been sparked during the OIST Computational Neuroscience Course and has been supported by accommodations and travel grants from the Okinawa Institute of Science and Technology (OIST) to H.R. and M.S. H.R. is supported by the German Research Foundation (grant no. 403329959 to MN) within the Research Unit "Structure, Plasticity and Behavioral Function of the Drosophila mushroom body" (DFG-FOR 2705, www.uni-goettingen.de/en/601524.html).

## AUTHOR CONTRIBUTIONS

Conceptualization, H.R., M.P.N., and M.S.; Methodology, H.R. and M.S.; Writing—Original Draft, H.R.; Writing—Review and Editing, H.R., MP.N., and M.S.

## DECLARATION OF INTERESTS

The authors declare no competing interests.

Received: May 23, 2019
Revised: November 24, 2019
Accepted: January 14, 2020
Published: February 21, 2020; corrected online: June 16, 2020

## REFERENCES

Avarguès-Weber, A., Deisig, N., and Giurfa, M. (2011). Visual cognition in social insects. Annu. Rev. Entomol. *56*, 423–443.

Avarguès-Weber, A., Dyer, A.G., Combe, M., and Giurfa, M. (2012). Simultaneous mastering of two abstract concepts by the miniature brain of bees. Proc. Natl. Acad. Sci. U S A *109*, 7481–7486.

Avarguès-Weber, A., and Giurfa, M. (2013). Conceptual learning by miniature brains. Proc. R. Soc. B Biol. Sci. *280*, 20131907.

Bengio, Y., Léonard, N., and Courville, A. (2013). Estimating or propagating gradients through stochastic neurons for conditional computation. arXiv:1308.3432.

Bhattacharya, B.S., and Furber, S.B. (2010). Biologically inspired means for rank-order encoding images: a quantitative analysis. IEEE Trans. Neural Netw. *21*, 1087–1099.

Bi, G.q., and ming Poo, M. (2001). Synaptic modification by correlated activity: Hebb's postulate revisited. Annu. Rev. Neurosci. *1*, 139–166.

Zhao, B., Sun, J., Zhang, X., Mo, H., Niu, Y., Li, Q., Wang, L., and Zhong, Y. (2019). Long-term memory is formed immediately without the need for protein synthesis-dependent consolidation in drosophila. Nat. Commun. *10*, 4550.

Bohte, S.M., Kok, J.N., Poutré, H.L.., 2000.Spikeprop: backpropagation for networks of spiking neurons, in: ESANN.

Caporale, N., and Dan, Y. (2008). Spike timing–dependent plasticity: a hebbian learning rule. Annu. Rev. Neurosci. *31*, 25–46.

Chittka, L., and Geiger, K. (1995). Can honey bees count landmarks? Anim. Behav. *49*, 159–164.

Chittka, L., and Niven, J. (2009). Are bigger brains better? Curr. Biol. *19*, R995–R1008.

Choromanska, A., LeCun, Y., Arous, G.B., 2015.Open problem: the landscape of the loss surfaces of multilayer networks, In: Conference on Learning Theory, pp. 1756–1760.

Dacke, M., and Srinivasan, M.V. (2008). Evidence for counting in insects. Anim. Cogn. *11*, 683–689.

Farkhooi, F., Muller, E., and Nawrot, M.P. (2011). Adaptation reduces variability of the neuronal population code. Phys. Rev. E *83*, 050905.

Fomoro, (2017). http://github.com/fomorians/counting-mnist.

Gütig, R. (2016). Spiking neurons can discover predictive features by aggregate-label learning. Science *351*, https://doi.org/10.1126/science.aab4113.

Gütig, R., and Sompolinsky, H. (2006). The tempotron: a neuron that learns spike timing-based decisions. Nat. Neurosci. *9*, 420–428.

Howard, S., Avarguès-Weber, A., Garcia, J., Greentree, A., and Dyer, A. (2018). Numerical ordering of zero in honey bees. Science *360*, 1124–1126.

Howard, S., Avarguès-Weber, A., Garcia, J., Greentree, A., and Dyer, A. (2019). Numerical cognition in honeybees enables addition and subtraction. Sci. Adv. *5*, eaav0961.

Huerta, R., Nowotny, T., García-Sanchez, M., Abarbanel, H.D.I., and Rabinovich, M.I. (2004). Learning classification in the olfactory system of insects. Neural Comput. *16*, 1601–1640.

Huh, D., and Sejnowski, T.J. (2017). Gradient descent for spiking neural networks. https://arxiv.org/abs/1706.04698.

Jin, C., Ge, R., Netrapalli, P., Kakade, S.M., Jordan, M.I., 2017.How to escape saddle points efficiently, In: Proceedings of the 34th International Conference on Machine Learning-Volume 70, JMLR. org. pp. 1724–1732.

Kingma, D.P., and Ba, J. (2014). Adam: a method for stochastic optimization. https://arxiv.org/abs/1412.6980.

Krizhevsky, A., Sutskever, I., and Hinton, G.E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems 25, F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, eds. (Curran Associates, Inc.), pp. 1097–1105. http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

LeCun, Y., and Cortes, C. (2010). MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/.

Lempitsky, V., Zisserman, A., 2010.Learning to count objects in images, in: Advances in Neural Information Processing Systems, pp. 1324–1332.

Levy, W.B., and Baxter, R.A. (1996). Energy efficient neural codes. Neural Comput. *8*, 531–543.

Memmesheimer, R.M., Rubin, R., Ölveczky, B.P., and Sompolinsky, H. (2014). Learning precisely timed spikes. Neuron *82*, 925–938.

Menzel, R., Fuchs, J., Nadler, L., Weiss, B., Kumbischinski, N., Adebiyi, D., Hartfil, S., and Greggers, U. (2010). Dominance of the odometer over serial landmark learning in honeybee navigation. Naturwissenschaften *97*, 763–767.

Mochizuki, Y., Onaga, T., Shimazaki, H., Shimokawa, T., Tsubo, Y., Kimura, R., Saiki, A., Sakai, Y., Isomura, Y., Fujisawa, S., et al. (2016). Similarity in neuronal firing regimes across mammalian species. J.Neurosci. *36*, 5736–5747.

Nawrot, M.P. (2010). Analysis and interpretation of interval and count variability in neural spike trains. In Analysis of Parallel Spike Trains (Springer), pp. 37–58.

Nicola, W., and Clopath, C. (2017). Supervised learning in spiking neural networks with force training. Nat. Commun. *8*, 2208.

Niven, J.E. (2016). Neuronal energy consumption: biophysics, efficiency and evolution. Curr. Opin. Neurobiol. *41*, 129–135.

Niven, J.E., and Laughlin, S.B. (2008). Energy limitation as a selective pressure on the evolution of sensory systems. J. Exp. Biol. *211*, 1792–1804.

O'Connor, P., Gavves, E., and Welling, M. (2017). Temporally efficient deep learning with spikes. https://arxiv.org/abs/1706.04159.

OpenAI, (2018). https://blog.openai.com/ai-and-compute/.

Pahl, M., Si, A., and Zhang, S. (2013). Numerical cognition in bees and other insects. Front. Psychol. *4*, 162.

Pamir, E., Szyszka, P., Scheiner, R., and Nawrot, M.P. (2014). Rapid learning dynamics in individual honeybees during classical conditioning. Front. Behav. Neurosci. *8*, 313.

Rapp, H., and Stern, M. (2019). Matlab implementation of multispiketempotron with adaptive learning. https://doi.org/10.5281/zenodo.3603129.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. https://arxiv.org/pdf/1506.01497.

Rose, G.J. (2018). The numerical abilities of anurans and their neural correlates: insights from neuroethological studies of acoustic communication. Philos. Trans. R. Soc. B Biol. Sci. *373*, 20160512.

Scheunemann, L., Skroblin, P., Hundsrucker, C., Klussmann, E., Efetova, M., and Schwarzel, M. (2013). Akaps act in a two-step mechanism of memory acquisition. J.Neurosci. *33*, 17422–17428.

Schmidhuber, J. (2015). Deep learning in neural networks: an overview. Neural Netw. *61*, 85–117.

Seguí, S., Pujol, O., and Vitrià, J. (2015). Learning to count with deep object features. https://arxiv.org/abs/1505.08082.

Simonyan, K., and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. https://arxiv.org/abs/1409.1556.

Skorupski, P., MaBouDi, H., Galpayage Dona, H.S., and Chittka, L. (2018). Counting insects. Philos. Trans. R. Soc. B: Biol. Sci. *373*, 20160513.

Song, S.M.K.D., and Abbott, L.F. (2000). Competitive hebbian learning through spike-timing-dependent synaptic plasticity. Nat. Neurosci. *3*, 919.

Thorpe, S., and Gautrais, J. (1998). Rank order coding. Comput. Neurosci. *1*, 113–118.

Tieleman, T., and Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by arunning average of its recent magnitude. COURSERA: Neural networks for machine learning *4*, 26–31.

Vasas, V., and Chittka, L. (2019). Insect-inspired sequential inspection strategy enables an artificial network of four neurons to estimate numerosity. iScience *11*, 85–92.

Yu, F., and Koltun, V. (2015). Multi-scale context aggregation by dilated convolutions. https://arxiv.org/abs/1511.07122.

Zenke, F., and Ganguli, S. (2018). Superspike: Supervised learning in multilayer spiking neural networks. Neural Comput. *30*, 1514–1541.

iScience, Volume *23*

## Supplemental Information

# Numerical Cognition Based on Precise Counting with a Single Spiking Neuron

**Hannes Rapp, Martin Paul Nawrot, and Merav Stern**

## Supplemental Information

### *Transparent Methods*

To support further research we make our code and data-sets publicly available at Rapp and Stern (2019).

### *Multispike Tempotron Model*

The Multi-Spike Tempotron (MST) is a current-based leaky integrate-and-fire neuron model (Gütig, 2016). Its membrane potential, $V(t)$, follows the dynamical equation:

$$V(t) = \underbrace{V_{rest}}_{:=0} + \sum_{i=1}^{N} \omega_i \sum_{t_i^j < t} \overbrace{K(t - t_i^j)}^{\text{exp. PSP kernel}} - (\underbrace{\vartheta}_{:=1} - V_{rest}) \sum_{t_s^j < t} e^{-\frac{t - t_s^j}{\tau_m}} \tag{1}$$

where $t_i^j$ denotes the time of spike number $j$ from the input source (presynaptic) number $i$, and $t_s^j$ denotes the time of postsynaptic spike number $j$ of the Tempotron neuron model. For mathematical convenience the resting potential is chosen to be $V_{rest} = 0$ and the spiking threshold $\vartheta = 1$. Thus equation 1 can be simplified to:

$$V(t) = \overbrace{\sum_{i=1}^{N} \omega_i \sum_{t_i^j < t} K(t - t_i^j)}^{\text{unreset sub-threshold potential } V_0} - \vartheta \sum_{t_s^j < t} e^{-\frac{t - t_s^j}{\tau_m}} \tag{2}$$

Every input spike at $t_i^j$ contributes to the postsynaptic potential (PSP) by the following causal kernel:

$$K(t - t_i^j) = \begin{cases} V_{norm}(e^{-\frac{t - t_i^j}{\tau_m}} - e^{-\frac{t - t_i^j}{\tau_s}}) & \text{if } t \geq t_i^j \\ 0 & \text{if } t < t_i^j \end{cases} \tag{3}$$

multiplied with the synaptic weight $\omega_i$ of input synapse $i$. These synaptic input weights are learned via the gradient decent algorithm. The kernel is normalized to have its peak value at 1 with $V_{norm} = \frac{\eta^{\frac{\eta}{(\eta-1)}}}{(\eta-1)}$ and $\eta = \frac{\tau_m}{\tau_s}$ where $\tau_m$ and $\tau_s$ are the membrane time constant and the synaptic decay time constant. The kernel is made causal by setting it to 0 for $t < t_i^j$. When $V(t)$ crosses the spiking threshold $\vartheta$ the neuron emits a spike and is reset to $V_{rest} = 0$ by the last term in equation 2.

In order to have the neuron emit the required number of $k$ postsynaptic spikes in response to some presynaptic spike pattern the weights $\omega_i$ are modified. Since the required number of postsynaptic spikes are non-differentiable discrete numbers the gradient for adjusting the weights is derived from the spiking threshold using an auxiliary objective function, the spike-threshold surface (STS). The STS is a step function $\mathbb{R}^+ \mapsto \mathbb{N}_0$, which maps each threshold value $\vartheta$ to the number of output spikes ($\vartheta \mapsto STS(\vartheta)$) that will be generated by the neuron with this threshold value. The STS for a presynaptic input can be described by the decreasing sequence of critical thresholds values $\vartheta_k^*$:

$$\vartheta_k^* = sup\{\vartheta \in \mathbb{R}^+ STS(\vartheta) = k\}, k \in \mathbb{N} \tag{4}$$

The critical threshold $\vartheta_k^*$ denotes the threshold value at which the neuron's number of generated output spikes jumps from $k - 1$ to $k$. The number of generated output spikes remains constant when $\vartheta$ is between two critical threshold values: $STS(\vartheta_{k+1}^* < \vartheta < \vartheta_k^*) = k$. Additionally, a neuron does not fire any output spike if its threshold is larger than the maximum postsynaptic voltage ($V_{max}$). In this case the STS is zero: $STS(\vartheta > V_{max}) = 0$. The first output spike is generated when $\vartheta = V_{max}$, thus the critical threshold for $k = 1$ spike is $\vartheta_1^* = V_{max}$. Generally, all $\vartheta_k^*$ are voltage values and can be described by the neuron's membrane equation 2 which is a function of the synaptic weights $\omega_i$ of the neuron. Hence, all critical thresholds are also a function of $\omega_i$ and thus differentiable with respect to them. The goal is to adjust $\vartheta_k^*$ (by modifying the synaptic weights $\omega_i$) whenever the number of generated spikes does not match the desired training target. In our case, the specific $k$ of desired output spikes is provided as supervised teaching signal. For each presynaptic input where the number of output spikes did not match the desired training target a training step is performed to adjust the number of output spikes towards $k$: $\Delta k = |k_{generated} - k_{target}|$ and $\eta = sign(k_{generated} - k_{target})$ indicates whether the neuron should increase or decrease its number of output spikes by $\Delta k$.

To simplify notation, from now on we denote $\vartheta^*$ as the desired critical threshold, e.g. $\vartheta^* = \vartheta^*_k$ for the desired $k$ of a specific presynaptic input.

The gradient of the critical threshold can be found by:

$$\Delta\omega = \eta\lambda\vec{\nabla}_{\vec{\omega}}\vartheta^* \tag{5}$$

Where $\eta \in \{-1, 1\}$ controls whether to increase or decrease the number of output spikes towards the $k$ required spikes, $\lambda$ is the learning rate parameter that controls the size of the gradient step to take in each training step and $\vec{\nabla}_{\vec{\omega}}\vartheta^*$ is the gradient of the critical spiking threshold with respect to the synaptic weights. To evaluate the expression in eq. 5 the properties of the critical spike time $t^*$ is used where by definition of the neuron equation 2 and $\vartheta^*$ the following identity is satisfied:

$$\vartheta^* = V(t^*) = V(t_s^j) \quad \text{where } t_s^j \text{ are all spike times before } t^* \tag{6}$$

In what follows a recursive expression is derived for the gradient in equation 5 using equations 2 and 3. For notional clarity the recursive expression for the gradient is derived for a single component $\omega_i$ of the vector $\vec{\omega}$. The generalization to $\vec{\omega}$ is immediate.

Let $m$ denote the the number of output spikes the neuron fires before $t^*$: $t_s^j < t^*$ for $j \in \{1, ..., m\}$. Using the identities in 6, for each synapse $i$ the derivative of $\vartheta^*$ has the following properties:

$$\vartheta_i^{*\prime} \equiv \frac{\mathrm{d}}{\mathrm{d}\omega_i}\vartheta^* = \frac{\mathrm{d}}{\mathrm{d}\omega_i}V(t^*) = \frac{\mathrm{d}}{\mathrm{d}\omega_i}V(t_s^j) \tag{7}$$

And the derivative of $\vartheta^*$ follows the equation:

$$\vartheta_i^{*\prime} = \frac{\partial}{\partial\omega_i}V(t^*) + \sum_{j=1}^{m}\frac{\partial}{\partial t_s^j}V(t^*)\frac{\mathrm{d}}{\mathrm{d}\omega_i}t_s^j \tag{8}$$

In the last equation the vanishing term $\frac{\partial}{\partial t^*}V(t^*)\frac{\mathrm{d}}{\mathrm{d}\omega_i}t^* = 0$ has been dropped. This relationship is true because $V(t^*)$ is either a local maximum with $\frac{\partial}{\partial t^*}V(t^*) = 0$ or $t^*$ is the arrival time of an inhibitory input spike that does not depend on $\omega_i$.

Similarly for each $k \in \{1, .., m\}$ the following relationship holds:

$$\frac{\mathrm{d}}{\mathrm{d}\omega_i} V(t_s^k) = \frac{\partial}{\partial \omega_i} V(t_s^k) + \sum_{j=1}^{k} \frac{\partial}{\partial t_s^j} V(t_s^k) \frac{\mathrm{d}}{\mathrm{d}\omega_i} t_s^j \tag{9}$$

from which the following equations are obtained:

$$\frac{\mathrm{d}}{\mathrm{d}\omega_i} t_s^k = \frac{1}{\dot{V}(t_s^k)} \left[ \vartheta_i^{*\prime} - \frac{\partial}{\partial \omega_i} V(t_s^k) - \sum_{j=1}^{k-1} \frac{\partial}{\partial t_s^j} V(t_s^k) \frac{\mathrm{d}}{\mathrm{d}\omega_i} t_s^j \right] \tag{10}$$

$$\dot{V}(t_s^k) = \frac{\partial}{\partial t} V(t) \bigg|_{t=t_s^{k-}} \text{ evaluated from the left before spike reset} \tag{11}$$

To solve equation 8 for $\vartheta_i^{*\prime}$, the right hand side of eq 10 is refactored to:

$$\frac{\mathrm{d}}{\mathrm{d}\omega_i} t_s^k = \frac{1}{\dot{V}(t_s^k)} \left[ \vartheta_i^{*\prime} A_k + B_k \right] \tag{12}$$

The coefficients $A_k, B_k$ are given by the following recursive equations:

$$A_k = 1 - \sum_{j=1}^{k-1} \frac{A_j}{\dot{V}(t_s^j)} \frac{\partial}{\partial t_s^j} V(t_s^k) \tag{13}$$

$$B_k = -\frac{\partial}{\partial \omega_i} V(t_s^k) - \sum_{j=1}^{k-1} \frac{B_j}{\dot{V}(t_s^j)} \frac{\partial}{\partial t_s^j} V(t_s^k) \tag{14}$$

Similarly for $t^*$ the analogous recursion formula is defined:

$$A_* = 1 - \sum_{j=1}^{m} \frac{A_j}{\dot{V}(t_s^j)} \frac{\partial}{\partial t_s^j} V(t^*) \tag{15}$$

$$B_* = -\frac{\partial}{\partial \omega_i} V(t^*) - \sum_{j=1}^{m} \frac{B_j}{\dot{V}(t_s^j)} \frac{\partial}{\partial t_s^j} V(t^*) \tag{16}$$

Inserting equation 12 into 8 the derivative of $\vartheta_i^{*\prime}$ for each vector component $i$ of $\omega$ can be expressed as:

$$\vartheta_i^{*\prime} = -\frac{B_*}{A_*} \tag{17}$$

To calculate $A_*$ and $B_*$ all times $t_x \in \{t_s^1, t_s^2 ... t_s^m, t^*\}$ are considered at which the voltage reaches the spiking threshold $\vartheta$. At these time points, due to the spiking and reset, the membrane potential equation 2 reduces to the form:

$$V(t_x) = \frac{V_0(t_x)}{C_{t_x}} \tag{18}$$

with

$$V_0(t) = \sum_{i=1}^{N} \omega_i \sum_{t_i^j < t} K(t - t_i^j) \quad \text{unreset sub-threshold potential} \tag{19}$$

$$C_{t_x} = 1 + \sum_{t_s^j < t_x} e^{-\frac{t_x - t_s^j}{\tau_m}} \tag{20}$$

and gives the following derivatives:

$$\begin{aligned} \frac{\partial}{\partial \omega_i} V(t_x) &= \frac{1}{C_{t_x}} \frac{\partial}{\partial \omega_i} V_0(t_x) & (21) \\ &= \frac{1}{C_{t_x}} \sum_{t_i^j < t_x} K(t_x - t_i^j) & (22) \end{aligned}$$

$$\begin{aligned} \frac{\partial}{\partial t_s^k} V(t_x) &= \frac{-V_0(t_x)}{C_{t_x}^2} \frac{e^{-\frac{t_x - t_s^k}{\tau_m}}}{\tau_m} \quad \text{for } t_s^k < t_x & (23) \\ \dot{V}(t_x) &= \frac{1}{C_{t_x}^2} \left[ C_{t_x} \frac{\partial}{\partial t_x} V_0(t_x) + \frac{V_0(t_x)}{\tau_m} \sum_{t_s^j < t_x} e^{-\frac{t_x - t_s^j}{\tau_m}} \right] & (24) \end{aligned}$$

Where in our implementation the temporal derivative $\dot{V}(t_x)$ is estimated numerically instead of using its analytical expression.

*Momentum and Adaptive learning*

The learning rate $\lambda$ is global for all synaptic weights. Hence, the gradient descent takes an equal size step along all directions. If this parameter is too small the training process will take very long, but if it's too big the algorithm might miss an optimum within the error surface and never converge to a good solution. Hence, tuning this learning rate is important to achieve decent training speed. A possible approach (Gütig, 2016) to avoid these problems is to update the weights according to exponential moving average of current and past gradients (up to training step t), using the *Momentum* heuristic:

$$\begin{aligned} \Delta\omega^{Momentum} &= \alpha \Delta\omega(t-1) + \Delta\omega(t) \\ &= \alpha \Delta\omega(t-1) + \eta \lambda \vec{\nabla}_{\vec{\omega}} \vartheta^*, & (25) \end{aligned}$$

where $\alpha$ is the *Momentum* meta-parameter to control the exponential smoothing effect. In practice, a common heuristic in the machine learning community is to choose $\alpha$'s value as 0.999 while tuning the global learning rate $\lambda$.

*Adaptive input weight learning and gradient smoothing*

We propose here to use an adaptive learning approach for the weight updates instead of the *Momentum* heuristic. The proposed algorithm fits each input synapse with its own update rate and by doing so it takes into account that each synapse contributes to the overall update with a different level of importance. For example, updates should be larger for directions that provide more consistent information across examples. The RMSprop (Root Mean Square (back-)propagation) (Tieleman and Hinton, 2012) is a possible approach to achieve this. It was successfully used in deep learning for training mini-batches. It computes an adaptive learning rate per synapse weight $\omega_i$ as a function of its previous gradient steps :

$$\begin{aligned} v_i(t) &= \gamma v_i(t-1) + (1-\gamma)(\Delta\omega_i(t))^2 \\ \Delta\omega_i^{Adaptive}(t) &= \frac{\eta\lambda}{\sqrt{v_i(t)}}\vec{\nabla}_{\omega_i}\vartheta^* \end{aligned} \tag{26}$$

The dynamical variable $v_i(t)$ gives the synapse specific (e.g. local) learning rate for the current training step $t$. The value of this variable depends on the exponential moving average of current and past squared gradients (up to training step $t$). The meta-parameter $\gamma$ controls the degree of exponential smoothing similarly to $\alpha$ of the Momentum method above. Setting $\gamma = 1$ would be similar to vanilla gradient descent where only the gradient of current training step $t$ is used to update. In practice a common heuristic for the choice of $\gamma$ in the deep learning community (also suggested by Tieleman and Hinton (2012)) is 0.999 and instead only tuning the global learning rate $\lambda$.

At this point we cannot provide a theoretically grounded explanation for the regularizing effect we see and report in the results section when using adaptive learning instead of Momentum. Theoretically grounded explanations of the effects of different gradient-decent optimizers are a very recent and ongoing research field in the machine learning community. We thus conducted an empirical analysis of the weight updates and report our findings in Figure 2C and conclusions in the discussion.

*Detection of spatio-temporal input spike patterns.*

In this task we study the general case of counting arbitrary, task dependent patterns. To this end we use 1sec long spike trains generated from point processes as a model of complex spatio-temporal patterns that represent features of task dependent activity. An input to the MST model consists of a sequence of such patterns, each of which assigned with a specific target $\mathcal{R}_i$. The patterns are superimposed onto a 10sec long spike train of background activity. Similar to the task in (Gütig, 2016) the MST model is trained to respond with spikes for each pattern occurrence where the number of spikes per pattern depends on its assigned target $\mathcal{R}_i$. For each data-set a training set of 200 samples and a separate validation set of 50 samples is generated. Each pattern is associated with a fixed, positive integer target $\mathcal{R}_i \in [0, 9]$. For each data-set the patterns are generated from a different renewal process. Out of the 9 patterns, 5 patterns are considered to be *task-related* and are associated with some positive target $\mathcal{R}_i$. The remaining 4 patterns are considered to be distractor patterns with target 0. The training target for each of such input spike train is determined as the sum over all individual targets $\Sigma_i \mathcal{R}_i$ of each occurring pattern.

For each data-set, at the end of each training epoch, the error in the MST performance is calculated as the mean absolute difference between the target input spike count and the actual MST response across all training trails (training error) or testing trials (validation error),

It has been shown that in-vivo cortical spiking activity is typically more regular than Poisson (Mochizuki et al., 2016; Nawrot, 2010). In general any correlated stimuli input is expected to deviate from Poisson (Farkhooi et al., 2011). Moreover, input is generally non-homogenous, i.e. time-varying. However in (Gütig, 2016) only homogeneous Poisson statistic of input patterns and background were considered.

All patterns are generated as 1sec long spike trains by drawing instantaneous firing rates from three different point processes (renewal processes): $\Gamma_1$ representing the homogeneous Poisson process, $\Gamma_5$, and $\Gamma_{15}$ represent Gamma-Processes with a fixed intensity (or rate) of $\lambda = 0.89$ spike events per second.

Input spike trains of 10 sec duration and 500 presynaptic inputs are generated by simulating a 10 sec long spike train of background activity using renewal processes and patterns are superimposed onto this background activity. The number of patterns to appear within a sequence is drawn from a Poisson distribution of mean 5 patterns. These patterns are randomly positioned in time within those 10 sec but are not allowed to overlap (an example of an input spike-train is shown in fig. 1A).

We evaluate learning under different noise conditions, where one condition uses homogenous Poisson background activity and the other condition uses inhomogenous Poisson background activity. The homogenous background activity is drawn from a stationary Poisson process ($\lambda = 0.3$ spikes/sec) while for the inhomogenous case the instantaneous firing rates are slowly modulated by $\lambda(t) = sin(\frac{10\pi}{10000}t) + (\frac{4\pi}{10}\xi(t))$ where $\xi(t)$ is noise drawn from a standard normal distribution.

The free meta-parameters for Momentum and adaptive learning are set to be $\alpha = 0.999$ and $\gamma = 0.999$ respectively. These are heuristic values taken from current deep learning frameworks and in practice are treated as constant parameters. Thus, the only real free parameter is the global learning rate $\lambda$. Since the objective of this task is to study the effect of the two different update methods, we are not concerned to determine the optimal learning rate that would give the best possible, absolute numbers in terms of training error. The described effect in the results section is independent of the specific choice we made $\lambda = 0.001$, although the absolute numbers vary.

*Counting handwritten digits*

This task considers the problem of estimating numerosity. Specifically the problem of counting the occurrences of digit 1 within an image showing 9 random MNIST (LeCun and Cortes, 2010) digits positioned within a $3x3$ grid. Following Seguí et al. (2015) we generated new images of size 50x50 pixels. Each image is subdivided into a $3x3$ grid where each grid cell shows a randomly chosen (with replacement), single MNIST digit. Out of the 9 possible cells, up to 6 cells can be occupied by digit 1. This yields samples with possible targets from $0-6$. The generated data set is only roughly balanced, containing $\sim 200$ samples for each target $0-6$. The model is supposed to learn to count the number of occurrences of the digit 1 by generating one output spike per each occurrence. The training target is provided by a single scalar label of the number of digits 1 in the image. All models are trained using 5-fold cross-validation. While the training set for the MST model comprises 400 samples, the ConvNet is provided with 800 samples. Additionally, the ConvNet is provided with a much larger learning rate of $\lambda = 0.01$ to accelerate training, while the MST is manually tuned to use learning rate of $\lambda = 0.00002$. To train the ConvNet we use the ADAM (Kingma and Ba, 2014) optimizer which has been found to be an effective optimizer for training ConvNets. For the MST model we use our adaptive learning rate method where the meta-parameter is set to $\gamma = 0.999$. The MST model is trained for max. 30 epochs as it does not improve further after this. The ConvNet is trained for max. 200 epochs. For all models, the training is considered to be converged at that epoch before the validation error diverges for the first time. While the ConvNet shows monotonic decrease of validation error, the MST fluctuates.

For the Multi-Spike Tempotron the images have to be encoded as spike trains. This is done by using *Filter-Overlap Correction Algorithm* (FoCal) (Bhattacharya and Furber, 2010), a 4-layer model of the early visual system that uses an improved rank-order coding originally proposed by Thorpe and Gautrais (1998). Encoding a single 50$x$50px image thus yields a spike train with $4 \times 50^2 = 10000$ synapses. The encoding algorithm makes use of spatial correlations in order to reduce the amount of redundant information. This is similar to the convolutional filters embedded in current deep neural networks (Simonyan and Zisserman, 2014; Krizhevsky et al., 2012). For reference, we train a conventional ConvNet architecture that has been shown to successfully accomplish this task when trained on 100000 samples. The architecture uses several layers (conv1 - MaxPool - conv2 - conv3 - conv4 - fc - softmax) and includes recently discovered advances like strided and dilated convolutions (Yu and Koltun, 2015).

The free meta-parameters for the MST model, Adaptive learning parameter $\gamma$ and global learning rate $\lambda$, are set to be $\gamma = 0.999$ and $\lambda = 0.0001$. This learning rate has been determined manually, by step-wise decreasing from 0.1 by factor of 10 until reaching the best trade-off between learning speed and convergence of validation error. For the choice of $\gamma$ we refer to the explanation given in the method section above.

*Insect-inspired numerical cognition during visual inspection flights*

Following Vasas and Chittka (2019); Howard et al. (2018) we consider estimation of numerosity of geometric shapes during a sequential inspection strategy employed by insects. We use 97 sample trajectories from sequential inspection flights from real honeybees, taken from supplements of Vasas and Chittka (2019). The available trajectories cover samples from 0 to 6 items (we removed 0 since it only had a single trajectory). Following Vasas and Chittka (2019) the trajectories have been used to extract a sequence of single images with a field of view (FOV) of 60° and 2cm distance to the inspected image. Thus each time point of a scanning trajectory yields a $183x183$ pixel image. Particularly, the absolute difference of each image $S$ between two successive time points $t$ and $t+1$ (1st derivative) of the trajectory is used: $FOV_{diff} = |S(t) - S(t+1)|$. While the proper way would be to use $|S(t-1) - S(t)|$ we decided to exactly follow the method used in Vasas and Chittka (2019). Differently from Vasas and Chittka (2019) the sequence of derivative images is down sampled to obtain sequences of equal length of 10 derivative images. This is done to reduce the computational cost as well as removing some redundant information from overlapping field of views between two successive time steps (a very coarse approximation of a working memory). All images are further down-scaled by factor 0.25 to $46x46$ pixels. This additional preprocessing is done to further reduce computational cost and to reduce the number of free parameters (synapses) in the MST model. To obtain spike trains from the image sequences, each $FOV_{diff}$ image is encoded as a short parallel spike train using Filter-overlap Correction (FoCal) algorithm

(Bhattacharya and Furber, 2010). FoCal resembles a 4-layer early visual system and is an improved rank-order coding scheme of images originally proposed by Thorpe and Gautrais (1998). The resulting parallel spike trains per $FOV_{diff}$ image are finally concatenated (without gaps) into a single long parallel spike train. Using this encoding results in parallel spike trains with 8468 input synapses to the MST. The MST model is trained (supervised) to fit its numbers of output spikes to the precise item count of geometric shapes. We used the adaptive learning method described above with $\gamma = 0.999$ (see explanation in methods section above), $\lambda = 0.00002$ (manually tuned) and performed a 10-fold, stratified cross-validation and trained for max. 25 epochs. We consider a model's training to be converged at that epoch before the validation error diverges for the first time. This generally was the case after 4-7 epochs. We assess the performance on a 'greater than' dual choice task following the original experiments of Howard et al. (2018). To this end, we randomly choose two independent samples of numerosity $(y_1, y_2)$ and feed the corresponding images into a randomly chosen, trained instance of the MST (10-fold cross-validation yields 10 independent models in total). A prediction by the MST $\hat{y}_1, \hat{y}_2$ is considered to be correct if $(y_1 > y_2) \wedge (\hat{y}_1 > \hat{y}_2)$ (and vice versa). In undecidable cases where $\hat{y}_1 = \hat{y}_2$ a random decision is made (coin-flip). This sampling process is repeated for 1000 random pairs, independently and separately for the training and testing data sets.

## 2.2 A spiking neural program for sensory-motor control during foraging in flying insects.

# A spiking neural program for sensory-motor control during foraging in flying insects.

**Hannes Rapp**[a] **and Martin Paul Nawrot**[a]

[a]Computational Systems Neuroscience, Institute of Zoology, University of Cologne, Zülpicher Strasse 47b, Cologne, Germany

This manuscript was compiled on June 22, 2020

**Foraging is a vital behavioral task for living organisms. Behavioral strategies and abstract mathematical models thereof have been described in detail for various species. To explore the link between underlying neural circuits and computational principles we present how a biologically detailed neural circuit model of the insect mushroom body implements sensory processing, learning and motor control. We focus on cast & surge strategies employed by flying insects when foraging within turbulent odor plumes. Using a spike-based plasticity rule the model rapidly learns to associate individual olfactory sensory cues paired with food in a classical conditioning paradigm. We show that, without retraining, the system dynamically recalls memories to detect relevant cues in complex sensory scenes. Accumulation of this sensory evidence on short time scales generates cast & surge motor commands. Our generic systems approach predicts that population sparseness facilitates learning, while temporal sparseness is required for dynamic memory recall and precise behavioral control. Our work successfully combines biological computational principles with spike-based machine learning. It shows how knowledge transfer from static to arbitrary complex dynamic conditions can be achieved by foraging insects and may serve as inspiration for agent-based machine learning.**

mushroom body | sparse coding | navigation | artificial intelligence | spiking neural network

Navigating towards a food source during foraging requires dynamical sensory processing, accumulation of sensory evidence and appropriate high level motor control. Navigation based on an animals olfactory sense is a challenging task due to the complex spatiotemporal landscape of odor molecules. A core aspect of foraging is the acquisition of sensory cue samples in the natural environment where odor concentrations vary rapidly and steeply across space. Experimental access to the neural substrate is challenging in freely behaving insects. Biologically realistic models thus play a key role in investigating the relevant computational mechanisms. Consequently, recent efforts at understanding foraging behavior have focused on identifying viable computational strategies for making navigational decisions (1).

An odor plume is often considered a volume wherein odor concentration is generally above some behavioral threshold. At macroscopic scales and in a natural environment, however, plumes are turbulent (2, 3). In turbulent conditions a plume breaks up into complex and intermittent filamentous structures that are interspersed with clean air pockets or below behavioral threshold concentration patches (4, 5). The dispersing filaments form the cone-like shape of the macroscopic plume where the origin of the cone yields the position of the odor source. When entering the cone, flying insects encounter odor filaments as discrete, short-lived sensory events in time.

Several features have been derived from the statistics of an odor plume that provide information regarding the location of the odor source (3, 4). The mean concentration varies smoothly in lateral and longitudinal directions of time-averaged (and laminar) plumes. However, for behavioral strategies animals cannot afford the time it takes to obtain stable macroscopic estimates of mean concentrations (2). Hildebrand and colleagues (6) proposed the time interval between odor encounters as an informative olfactory feature while (3) suggested intermittency, the probability of the odor concentration being above some behavioral threshold, as the relevant feature. However, similarly to estimating mean concentration, acquiring a sufficient number of samples for stable estimates of these quantities exceeds the time typically used to form behavioral decisions (2). Hence, obtaining time averaged quantities is not an optimal strategy to guide navigational decisions as concluded by (7).

Most animals perform searches at large distances from the odor source where the intermittency of plumes poses a more severe problem as available sensory cues become more sparse in space and time. Thus, strategies that exploit brief, localized sensory cues for navigation have been studied by several groups. One strategy for medium and long-range navigation that has consistently been observed across species of flying insects emerges from a sequence of chained sensori-motor reflexes: casting & surging (8). Encountering a whiff of odor triggers an upwind surge behavior, during which the insect travels parallel to the wind direction. After losing track of the plume it evokes a crosswind cast behavior, in which a flight path perpendicular to the direction of air flow is ex-

## Significance Statement

Deep learning networks based on dense tensor operations have demonstrated great success in pattern recognition tasks. Contrary, living organisms remain superior in mastering non-stationarities and can generalize their experience to rapidly adapt behavior. This paper demonstrates the benefits of using biological spiking neural networks, sparse computations and local learning rules. It highlights the functional roles of temporal- and population sparse coding for rapid associative learning, precise memory recall and transformation into navigational output. We show how memory formation generalizes to perform precise memory recall under dynamic, non-stationary conditions giving rise to non-trivial foraging behavior in a complex natural environment. Results suggest how principles of biological computation could benefit agent-based machine learning to deal with non-stationary scenarios.

Conceptualization, H.R., MP.N.; Methodology, H.R and MP.N.; Writing Original Draft, H.R., Writing Review and Editing, H.R. and MP.N.

The authors do not declare any conflicts of interest.

[2]To whom correspondence should be addressed. E-mail: martin.nawrotuni-koeln.de

ecuted. Performing repeated casts by U-turning allows the insect to reenter and locate the plume in order to trigger the next upwind surge (8–10). As the subject approaches the source it increasingly makes use of visual cues for navigation as the plume narrows down. (8).

A number of studies have proposed abstract mathematical models for optimal search algorithms that assumed different types of relevant navigational cues. The infotaxis method proposed in (11) depends on extensive memory and priors regarding a plume's structure. Contrary, in (8) only local cues are used. A standard algorithm for navigational problems in robotics is simultaneous localisation and mapping (SLAM), which has been used in (12) to study olfactory navigation in bumblebees. An algorithm that works without space perception has been proposed by (13) using a standardized projection of the probability of source position and minimization of a free energy along the trajectory. Finally, the work of (10) compares several models and shows that it is difficult to discriminate between different models based on behavioral responses. A recent work by (7) using information-theoretic analysis shows that plumes contain both, spatial and temporal information about the source's position.

While all of these previous mathematical methods for olfactory search algorithms have proven to successfully solve this task based on the respective assumptions, they share the same major drawback: none of them uses the computational substrate of the brain, spiking neurons and networks thereof. Instead, all methods make heavy use of symbolic math and advanced mathematical concepts that are not available to the biological brain. It is further unclear how and to what extend these methods could be implemented or learned by the nervous system. Additionally, the problem of navigation and foraging is often considered as an isolated task, independent from sensory processing.

Our approach distills recent experimental results to formulate a biologically plausible and detailed spiking neural network model supporting adaptive foraging behavior. We thereby take advantage of the rapidly accumulating knowledge regarding the anatomy (e.g. (14–16)) and neurophysiology (e.g. (17–19)) of insect olfaction and basic computational features (20, 21). We follow the idea of compositionality, a widely used concept in mathematics, semantics and linguistics. According to this principle, the meaning of a complex expression is a function of the meanings of its constituent expressions (Frege principle (22)). In the present context of foraging and navigation this means dynamically recombining memories of individual sensory cues present within a plume.

## Results

We approach the problem of foraging by decomposition into four components: First, sensory processing with temporal sparse and population sparse coding in the mushroom body (MB). Second, associative learning for assigning a valence to individual odor identities. Third, the time-dependent detection of valenced cues resulting from encounters of discrete odor filaments to provide an ongoing and robust estimate of sensory cue evidence. Fourth, the translation into online motor command signals to drive appropriate behavior.

For sensory processing we use a three-layer spiking neural network model of the insect olfactory pathway (see Fig 1). The generic blueprint of the insect olfactory system is homologous across species and comprises three successive processing stages (see Materials and Methods for details): The periphery with olfactory receptor neurons (ORNs), the antennal lobe (AL) and the MB. Excitatory feed-forward connections across layers from ORNs to projection neurons (PNs), from ORNs to local interneuron (LNs), and from PNs to the MB Kenyon cells (KCs) are fixed. Lateral inhibition within the AL uses fixed synaptic weights from LNs to PNs. For neuron numbers and their connectivity patterns we here rely on the adult *Drosophila melanogaster* where anatomical knowledge is most complete (14, 23, 24). A single MB output neuron (MBON) receives input from all Kenyon cells and plasticity at the synapses between KCs and the MBON enables associative learning (25, 26).

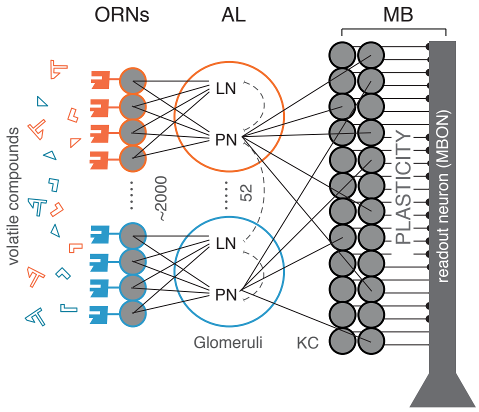


**Fig. 1. Spiking network model of the insect olfactory system.** Olfactory receptor neurons (ORNs, $N = 2080$) at the antenna bind and respond to volatile odorant compounds. ORNs expressing the same (one of 52 different) genetic receptor type converge onto the same glumerus in the antennal lobe (AL). Each of the 52 glomeruli comprises one projection neuron (PN) and one local interneuron (LN). Each LN forms lateral inhibitory connections with all PNs. PNs randomly connect to a large population of Kenyon Cells (KC, $N = 2000$) where each KC receives input from on average $\sim 6$ random PNs. All KCs project to a single MBON via plastic synapses.

**Sparse coding in space and time.** The olfactory system transforms a dense olfactory code in the AL into a sparse stimulus code at the MB level. In the large population of KCs, a specific odor stimulus is represented by only a small fraction of all KCs (population sparseness) and each stimulus-activated KC responds with only a single or very few action potentials (temporal sparseness). In our model, temporal sparseness is achieved through the cellular mechanisms of spike-frequency adaptation (SFA) implemented at two levels of the system.

ORNs show clear stimulus response adaptation that has been attributed to the spike generating mechanism (27). Based on this experimental evidence we introduced a slow and weak SFA conductance in our model ORNs (see Materials and Methods). At the level of the MB, KCs have been shown to express strong SFA-mediating channels (18). This is matched by the SFA parameters of our model KCs (see Materials and Methods, (21)). As an effect of cellular adaptation in ORNs and KCs, odor stimulation (Fig 2 A) results in temporally precise and adaptive responses across all layers of the network (Fig 2 B). The effect of SFA implemented in ORNs is transitive and thus carries over to the postsynaptic PN and

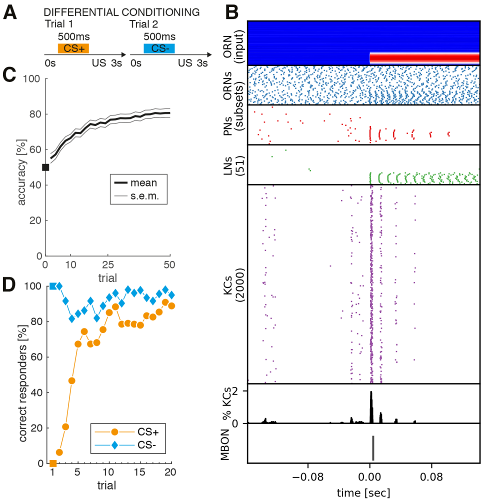


**Fig. 2. Rapid associative learning expressed in neuronal plasticity and conditioned response behavior. A:** Sketch of differential conditioning protocol. In appetitive trials a first sensory cue (conditioned stimulus, CS+, orange) is paired with a reward (unconditioned stimulus, US). In aversive trials, a second sensory cue (CS-, blue) is paired with a punishment. Both trial types are presented randomized within blocks (see Materials and Methods). **B:** Sensory input and neuronal responses across all four circuit layers (ORN, AL, MB, MBON) in response to a CS+ odor presentation during the 10th training trial. Stimulus onset is at $t = 0$ s. From top to bottom: Model input is provided through independent noise current injection into the ORNs. The stimulus-induced input currents are clearly visible (hot colors) on top of the background noise for the subset of ORNs that are sensitive to the CS+ odor (stimulus profile). Stimulus response is clearly visible by an increase in the spiking activity across all neuron populations. For ORNs (blue) and PNs (red) relevant subsets of $60$ and $35$ neurons are shown. The population of $2000$ KCs (magenta) show a temporal and spatial sparse odor response. Only 2% of all KCs are activated during a brief transient response following stimulus onset (black histogram). The MBON generates a single action potential in response to cue onset, which is the correct learned response to the CS+ odor. **C:** Learning performance of the MBON across $N = 100$ independent models as a function of the number of training trials. In any given trial the MBON response was correct if exactly one action potential was generated during CS+ presentation or if no action potential was generated during CS- presentation. **D:** The behavioral learning curve expresses the percentage of individuals that showed a correct behavior in the respective CS+ or CS- trial. The behavioral output is binary with either response or no response. The model triggers a response if the MBON generates one or more spikes.

LN populations in agreement with experimental observations across species (28–31).

In the KC population the background firing rate is very low ( 0.4 Hz). This is partially due to the outward SFA conductance and in agreement with experimental results (17). The KC population response is highly transitive where individual responding cells generate only a single or very few response spikes shortly after stimulus onset. This is in good qualitative and quantitative agreement with the temporal sparse KC spike responses measured in various species (17, 28, 32).

Population sparse stimulus encoding at the level of KCs is supported by two major factors. First, the sparse divergent-convergent connectivity between the PNs and the 20 times larger population of KCs is the anatomical basis for sparse odor representation (15, 20, 21, 33). Second, lateral inhibition mediated by the LNs in the AL (34) facilitates decorrelation of odor representations (34) and contributes to population sparseness (21). The sparse code in the KC population has been shown to reduce the overlap between different odor representations (35, 36) and consequently population sparseness is an important property of olfactory learning and plasticity models in insects (37–41). The system response to a single odor presentation in Fig. 2B) demonstrates the transformation of a dense olfactory code at the ORN and PN layers into a population sparse representation at the KC layer where less than $< 2\%$ of the total KC population is active at any time during stimulus presentation. This is in good agreement with quantitative estimates in the fruit fly (23, 42).

**Few-shot learning rapidly forms an associative memory of single cues with rewards.** Many insects exhibit a rapid learning dynamics when trained in classical olfactory conditioning tasks. They typically acquire high retention scores (test accuracy $> 60\%$) for a binary conditioned response (CR) behavior within only very few trials (e.g. (43–45)).

We here mimic a standard experimental lab protocol for differential conditioning (or acuity learning) to form associative memories and to generate a binary CR behavior by training our network (Fig. 1). Across successive learning trials we present two different odors in pseudo-random trial order (Fig. 2 A). Each trial constitutes a single odor presentation for 500 ms followed by a reinforcing stimulus (US) occurring shortly after the stimulus presentation. The CS+ odor is paired with a reward, the CS- odor with a punishment (see Materials and Methods). In order to establish a neural representation of the odor valence at the MB output (46–49) the MBON is trained (25, 26) to elicit exactly one action potential in response to the CS+ stimulus that is paired with the reward, and zero action potentials when the CS- stimulus is presented (see Materials and Methods). The system response to a single CS+ stimulus after nine conditioning trials is shown in Fig. 2 B.

In a first step we quantify the learning performance by considering the accuracy of the MBON response. MBON output is counted as correct if exactly one spike is generated during a CS+ trial and zero spikes during a CS- trial. The average accuracy over $N = 100$ independently trained model instances across successive trials is shown in Fig. 2 C. The learning dynamics shows a steep and steady increase indicating that an accurate memory is formed rapidly reaching up to 80% accuracy after 50 (25× CS+ and 25× CS-) training trials.

Next, we consider the behavioral learning curve, i.e. the acquisition of a binary CR behavior across successive learning trials. In each trial the model generates a behavioral response if the MBON produces one or more action potentials in response to the stimulus. No response is generated if the MBON remains silent. A CR is counted as correct if the MBON generates a response to the CS+ cue or no response to the CS- cue. The learning curve in Fig. 2 D represents the percentage of correctly responding individuals across $N = 100$ independently trained models. The untrained model, by default, does not generate any output spike, consequently 100% of the independent models correctly respond to CS- trials from the beginning (Fig. 2 D, blue). The red curve shows a rapid learning success where up to 70% of individuals generated the correct, appetitive CR to the CS+ stimuli within only $3 - 5$ trials. The learning curve saturates after $\sim$ 10 trials with an asymptotic value of $\sim$ 80% correct responders. This reproduces the rapid learning dynamics of insects in classical conditioning experiments and fits qualitatively and quantitatively the CR behavior in honeybees (for review see (44)).

We conclude that our statically configured sensory network model with a single plastic readout neuron is capable to successfully form associative memories by few-shot learning, replicating the classical conditioning experiments in the typical lab situation. The computational mechanism of population sparseness implemented in our model increases discriminability of the two different stimuli supporting a rapid learning dynamics and a high accuracy of memory recall.

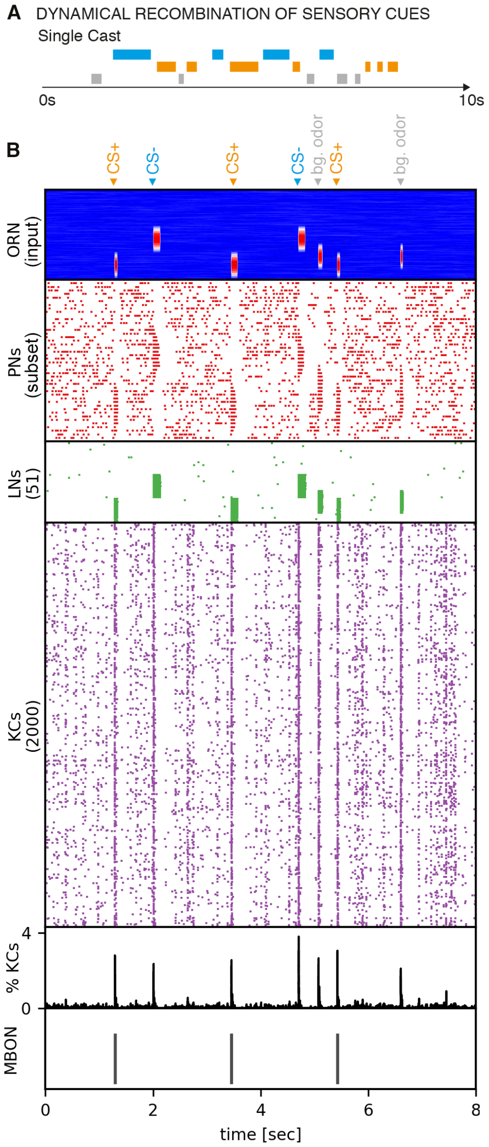


**Fig. 3.** Recognition of valenced odor cues in complex dynamic scenes. **A:** Sketch of the dynamical memory recall task mimicking the sensory experience during a natural foraging flight. In each single trial of $10s$ duration the model encounters multiple (on average five) cues of different odor identities including the CS+ odor (orange), the CS- odor (blue) and background odors (gray). **B** Network response to one example input sequence made up of three CS+ cues, two CS- cues and two distractor cues (bg odor) as indicated at the top. The ORN input sequence indicates the fluctuating duration of odor cues. Transient PN and LN response porfiles faithfully represent individual odor cue onsets in time and odor identity across neuronal space. The KC population shows clear responses to all individual odor cues albeit with $< 4\%$ of activated cells at any time. The MBON correctly produced a single action potential in response to each of the three CS+ cues and zero output else.

**Robust dynamic memory recall and odor-background segregation in complex sensory scenes.** We now challenge our previously trained model (Fig. 2) in a novel task asking whether the already learned odor associations can be reactivated in a complex and dynamic olfactory scene. To this end we mimicked the encounter of odor filaments in a turbulent odor plume during a foraging flight (Fig. 3 A). For this we presented random sequences of non-overlapping olfactory cues within $T = 10\,\text{s}$ (see Materials and Methods). Each cue is of variable duration in the range between 100 ms and 500 ms. Odor identity of each cue is randomly assigned to either the CS+, CS- odor or one out of three additional background odors (Fig. 4 A). The use of non-overlapping cues follows the rationale that, in nature, filaments originating from different odors do not mix perfectly (50).

The objective in this memory recall task is to correctly detect the occurrences of the positively valenced odor (CS+) by means of a single MBON action potential as model output while no output should be generated for all other cues (CS- or distractor odors). Fig. 3 B shows the system's response to a singe random stimulus sequence where the MBON correctly generated a single action potential in response to each of three CS+ encounters. For quantification of task accuracy we considered the overall response to a given sequence to be correct if the number of action potentials generated by the readout neuron is equal to the number of CS+ cues.

For assessing model performance we systematically vary task difficulty by varying the number of possible background odors (between one and three) and their similarity with the CS+ odor (Fig. 4). In a first task variant background odor activation profiles are rather distinct from the CS+ odor and more similar to the CS- odor (Fig. 4 A). Accuracy of the model response is computed across 200 test sequences as shown in Fig. 4 B. We find that our previously trained model successfully generalizes to this new task with ∼80% accuracy for different sequence complexity in terms of identity and number of background odors. In a second task variant we reversed the odor contingency of the CS+ and CS- odors during initial differential conditioning. Thus, the reward predicting odor CS+ is now more similar to two of the background odors while similarity with the third background odor remains unchanged (Fig. 4 C). In this more challenging case accuracy reduces to ∼ 50 % of sequences for which the model produced the correct number of MBON output spikes. Note that the accuracy measure in Fig. 4 is based on the correct cumulative spike count during a complete trial of 10 s. The more similar a background odor stimulus profile is to the CS+ odor, the more likely the model will produce false positive (FP) action potentials in response to such a similar odor and thus a total spike count that is higher than the number of CS+ occurrences. This is reminiscent of the effect observed in insects and other animals in odor discrimination tasks where perceptually similar odors are more difficult to distinguish from previously learned CS+ odors than perceptually dis-similar odors during memory retention tests. This might be overcome if similar odors are used during the initial differential conditioning.

We conclude that our network model is able to recall previously learned neural representation of odors and signal their valence in a temporally dynamic setting where the rewarded and punished odors appear with up to five times shorter durations and within an unpredictable temporal cue sequence of previously unknown background odors. The models thus also solves the problem of odor vs. background segmentation under quasi natural conditions (51).

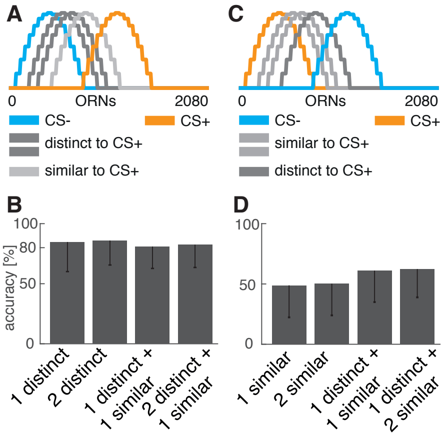


**Fig. 4. Performance in the dynamic memory recall task. A:** Input activation profiles across ORNs for five different odors. The model had been previously trained in the differential conditioning protocol using the orange CS+ and the blue CS- odors. In the dynamic memory recall task (see text) background odors (gray) were presented as distractor cues along with CS+ and CS- cues. In the first task variant two background odor profiles were rather distinct from CS+ (dark gray) and one is more similar (light gray). **B:** Task accuracy across $200$ trials in four different scenarios. A single trial consists of a random temporal sequence of sensory cue presentations during $10\,\mathrm{s}$. Each single cue is of random duration between $100 - 500\,\mathrm{ms}$. The single trial model response was correct if the MBON generated exactly as many action potentials as CS+ cues had been presented. The four task scenarios varied number (between one and three) and type (distinct vs. similar) of distractor cues as indicated. **C:** In the second task variant the same odors were used for cue presentation in a random temporal sequence. However, the model had been trained with reversed CS+/CS- odor contingency such that two distractor odors (light gray) are now more similar to the CS+ odor (orange). **D:** As in (B) but for reversed CS+/CS- odor contingency where distractor cues where overall more similar to the CS+ odor, increasing task difficulty.

**Accumulation of sensory evidences informs motor control in foraging.** We now consider the situation of foraging within a natural environment (Fig. 5 A). The objective is to locate the food source, which emits an attractive odor (CS+), by utilizing the sensory cues present in its turbulent odor plume. We show that cast & surge behavior can emerge by accumulation and exploitation of sensory evidence of sequentially experienced individual cues.

For this task we assume that thin odor filaments within a cross-wind plane of the concentric odor plume are approximately Gaussian distributed. This is a reasonable assumption, particularly in a wind-tunnel setting with laminar flow as typically used in experimental settings (8, 52). When the insect performs a cast through the plume, it encounters filaments as short-lived discrete, sequential events where each encounter represents a single sensory cue (see sketch in Fig. 5B). Therefore, in our simulation of casting flights the agent encounters sequences of cues and distractors where cue onsets for the CS+ odor are drawn from a Gaussian distribution while distractor cue onsets appear uniformly distributed over time (see Materials and Methods). We further assume that the subject has already formed an association of food with the attractive odor, either through learning or through some genetically predetermined innate valence. To this end we again use the trained model from the classical conditioning task above (Fig. 2) without any further re-training.

We simulate 4 consecutive casting trajectories where the agent senses odor cues of sequentially experienced filament encounters. Ongoing accumulation of sensory evidence (Fig. 5C) by low-pass filtering of the readout neuron's output assumes positive values shortly after entering the plume cone and further increases while approaching the plume's center line. When travelling beyond the center line sensory evidence slowly decreases until the agent leaves the plume cone boundary. When sensory evidence drops to zero and after a fix delay, the agent initiates a U-turn motor command to perform another cross-wind cast.

Responses from our model's readout neuron precisely follow the ground truth of CS+ odor cues as shown by 10 random casting trajectories in Fig. 5D. Performing analysis by averaging of sensory evidence across these 10 casting trajectories yields an average evidence (Fig. 5E) that faithfully resembles the underlying, true Gaussian profile of the simulated filaments.

We conclude that the model output provides an accurate and robust estimate of sensory evidence that can be used to reason about a plume's spatial extend and center line. Both information are crucial to generate appropriate motor commands for U-turn and upwind surge behavior, necessary to successfully execute the cast & surge strategy. Apart from the existence of filaments inside a plume and absence outside a plume's cone, our model does not make any specific assumption regarding the plume's structure and statistics. It thus provides a generic mechanism implemented in a neural system to perform cast & surge behavior during foraging flights.

## Discussion

**Distinct functional roles for population and temporal sparse stimulus encoding.** Population sparseness improves discriminability of different stimuli to facilitate associative learning. This has been demonstrated in theory and experiment (15, 20, 33, 36). We have shown, that our neural network model implements this feature in a biologically realistic way and our results confirm the functional role of population sparseness to support rapid and robust memory acquisition through associative learning. Experimental (36) and theoretical (20, 53) studies in the fruit fly strongly suggests that inhibitory feedback through the anterior paired lateral (APL) neuron improves population sparseness in the KC population. The APL is a GABAergic neuron that broadly innervates the KC population and likely receives input from KCs in the MB output region. Inhibitory feedback from MB output onto MB input has also been demonstrated in other species and blocking of feedback inhibition in the MB reduced population sparseness in the honeybee. Including an inhibitory feedback loop in our model would further improve robustness of population sparseness and thus not change our core findings.

Our model demonstrates how temporal sparseness can be exploited to generate short patterned signaling of cue identity. This enables perception of high temporal stimulus dynamics. In our model this is achieved independently of the duration of individual stimulus incidents and their distribution in time and makes temporally precise and robust sensory evidence available. It allows for the ongoing computation of derived estimates such as cue distributions or changes in cue

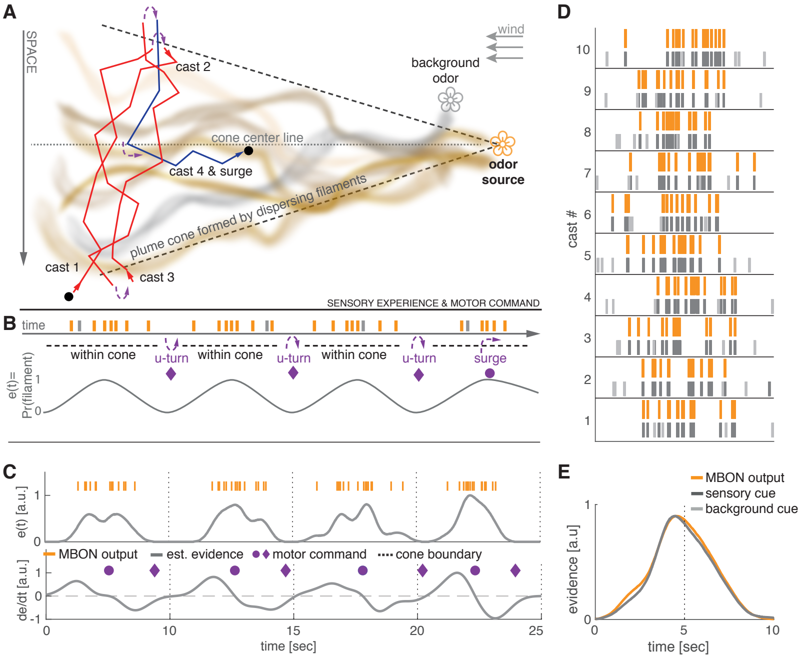


**Fig. 5. Dynamical sensory processing and motor control serving chemotaxis A:** Sketch of a typical olfactory experimental setup in a wind tunnel with a pleasant odor source (orange flower) and a second distractor source (gray flower). Due to turbulence the odor molecules emitted by a single source form dispersing, intermittent filaments within a cone-like boundary that constitutes the odor plume. The plume is modeled as Gaussian distributed filaments. A behaving model insect (here *Drosophila melanogaster*) performs stereotypic cast & surge behavior to locate the food source. This constitutes alternating between scanning crosswind and U-turning after running past the plume cone boundary where no filaments are present. Eventually, after several casts (here 3) it surges upwind until it loses track of the plume cone and starts over. **B:** Filament encounters during this behavior result in sequential brief on/off stimulations of the olfactory system. The probability of encountering filaments is $> 0$ within the plume and zero outside of the plume. Sensory evidence $e(t)$ can be viewed as a likelihood function of filament encounters that increases towards the plume's center line and is zero outside of the plume. The properties of this function can be used to generate optimal motor commands for chemotaxis. **C:** Evidence $e(t)$ and derivative $\frac{de}{dt}$ over 4 simulated successive casting trajectories estimated from the MBON spiking activity. *U-turn* motor commands (purple diamonds) are generated when $e(t)$ runs below a fixed threshold $(0.01)$ and *surge* motor commands (purple circles) are generated when $\frac{de}{dt}$ turns negative. The motor commands generated by the model match well with the theoretically optimal commands as sketched in panel B. **D:** Spiking activity of the MBON (orange) in response to $10$ casting trajectories. The MBON reliably predicts the true sensory cues of positively valenced filaments (dark gray) and ignores background cues (light gray). **E:** Smooth PSTH computed over $10$ casting trials recovers an accurate estimate of the true underlying sensory cue distribution simulated as Gaussian distribution.

density. Maintaining temporally sparse representations mechanistically supports the principle of compositionality (or Frege principle (22)), where an atomic stimulus entity is represented and can be learned by the readout neuron before processing this output. For example by estimation of densities or recombination with other entities to form composite perception or memory read-out. The temporal stimulus dynamics remains intact throughout the system even after learning of stimulus relevance. Thus valence is encoded with the same dynamics and faithfully captures occurrences of relevant cues. This allows compression of code to relevant stimuli while retaining full stimulus dynamics of the external world. Compression of code along the sensory processing pipeline is particularly relevant for small-brained animals like insects, which need to economize on their neuronal resources.

**Odor-background segregation: a joint effect of temporal and population sparse cue representation.** The task presented in Fig. 3 implicitly addresses the issue of odor-background segregation. This refers to the problem that in nature cues of multiple odors of different sources are present, either in terms of mixtures or stimulus onset asynchrony due to turbulent conditions (50, 51). For behavior it is relevant to reliably isolate and detect the relevant cues from any background or distractor cues. The results presented in Fig. 4 show that this works nicely in our system. This is achieved by exploiting the joint effect of temporal and population sparseness. Optimal discrimination of cue representation is guarantied by population sparseness and temporal precision by means of temporal sparseness. Our plastic output neuron requires population sparseness for learning and the plasticity rule (25, 26) allows for temporally precise memory recall. We predict that our model can solve the challenge of odor-background segregation.

**Rapid learning within few trials.** The ability of insects to quickly form associative memories after 3-5 trials has been demonstrated experimentally (44). However, in general few-shot learning remains a difficult task for computational models including insect inspired models (54). We find that, when compared with learning dynamics data of real insects (44) our model is able to show realistic learning dynamics that matches with the experimental observations. Due to frequent changes in the environment it might be a better strategy to trade-off fast and reasonable accurate learning against slow and high precision learning. Additionally, acquisition of training samples might be costly or they generally occur very sparsely.

Few-shot learning capabilities are also an active area of research in machine learning. Particularly current deep learning methods require massive amounts of training samples to successfully learn a classification model. For example, the popular benchmark data sets ImageNet and CIFAR10 for image classification contain 14 million and 60 million sample images, respectively. The *Google News dataset* used to train language models contains 100 billion words and learning to play the Space Invaders Atari game by deep reinforcement learning requires sampling of $>$ 500.000 game frames from the environment. Clearly, these are numbers a biological organism cannot afford to accumulate. In fact few-shot learning likely is a fundamental skill for survival. We have demonstrated that our neurobiologically motivated approach using spike-based computations is capable to perform few-shot learning with similar speed as insects. We further showed that our model can transfer learned associations to novel, complex combinations that have not been part of the training data (transfer leaning).

**Innate vs. learned behavior.** Cast & surge behavior belongs to the innate behavioral repertoire of air-borne insects and emerges from a set of sensori-motor reflexes (8). It can be considered as a base strategy which guarantees survival. The base system can be modulated and improved throughout an animal's lifespan by experience-based learning. This is superior to alternative strategies that would solely rely on learning appropriate behaviors and thus require constant re-training as is the case in machine learning approaches. Here, we assumed that our readout neuron is tuned to a pleasant odor. In the present work this tuning is learned (adaptive process) in a classical conditioning task. However, a tuning can generally be learned by other mechanisms, e.g. reinforcement learning. We demonstrated that the existence of such a tuned neuron allows cast & surge foraging behavior to emerge.

There are other ways how such a tuned neuron can come about, for example due to genetically predetermined wiring or during development from larval to adult stage. The cast & surge behavior can be executed on innately valenced olfactory cues and our suggested model for motor control during cast & surge (Fig. 5A+B) also works for innate valenced stimuli. Learning is important to adapt behavior to changing environmental circumstances and associative learning provides a means to learn new valences on demand in such situations. Our model learns odor valences at the mushroom body output and it has been shown that MBONs signal odor valence (46–49). We suggest that this valence is then used downstream to execute higher level functions of motor control. At this processing stage it might be integrated with innate valences and other necessary sensory modalities to form behavioral decisions.

**Implications for other sensory systems.** Sparse stimulus encoding has been identified as a powerful principle used by higher order brain areas to encode and represent features of the sensory environment in invertebrate (17, 23, 42) and vertebrate (55–58) systems. Sensory systems with similar coding principles may share similar mechanisms when it comes to learning and multi-modal sensory integration. The mushroom body is a center for integration of multi-modal sensory information. Thus, our model can be extended to incorporate input from different sensory modalities. It is known that olfactory search and foraging strategies do not solely rely on olfactory cues, but require additional sensory information from at least visual cues and wind direction. Extending our model to include additional sensory processing systems for vision and wind direction can provide a comprehensive functional model to study foraging and navigation.

**Potential improvement through multiple readout neurons..** Our current approach only comprises the simplest case of a single readout neuron. This model can be extended to multiple readout neurons. Different readout neurons can be tuned to different odors or groups of odorants. This would allow foraging for different types of food sources and further be useful for multi-modal sensory integration and learning of valences of multiple odors. Another way to use multiple readout neurons is to create an ensemble learning model. Particularly, one can perform bootstrap aggregation (bagging) to decrease variance of predictions. With this technique, multiple, independent readout neurons can be trained for the same target and their outputs are averaged to produce a single output. This approach can be useful when the level of noise increases due to different input models used to drive the network. Another possible extension is to use a single readout neuron to code for multiple odors by associating different number of action potentials to different odors (e.g. 2 or 3). The choice of model for the readout neuron and the plasticity rule allows to do this (25).

**Top-down motor control and lateral horn.** The model currently lacks a neural implementation of sensory evidence integration and generation of motor commands. Integration of sensory evidence is modeled by low-pass filtering of the readout neuron's spike train and its derivative is numerically estimated. In (59) it has been shown that a single compartment Hodgkin-Huxley neuron can operate in two computational regimes. One is more sensitive to input variance and acts like a differentiator while in the other regime it acts like an integrator. Similarly (60) has shown that the subthreshold current of neurons can encode the integral or derivative of their inputs based on their tuning properties. This and other suggested mechanistic implementations (e.g. (61)) could serve as basis for estimating the low-pass filtered sensory evidence and its derivative solely using neural computations. The initiation of a turning behavior based on a time-dependent evidence signal could be implemented e.g. through dis-inhibition of motor command neurons. The mechanism for U-turning could rely on either cell-intrinsic properties such as SFA where a neuron initiates a fast turning movement that decays with a fixed time constant, or through state-switching dynamics in neuronal populations.

**Relevance for machine learning and artificial intelligence..** Learning and building artificial intelligent agents capable of interacting with their environment are major objectives in the field of machine learning (ML) and artificial intelligence (AI). Deep artificial neural networks (62) have demonstrated great success over the recent years. Particularly, in the domains of image recognition, natural language processing and deep reinforcement learning (63). Despite their success, when applied to agent-based systems, their major drawback becomes evident. They are very specific, single-purpose perceptual systems and poorly generalize to new tasks or changes in an agent's environment (non-stationarities). A few methods to overcome this problem have been proposed, this includes retraining on new tasks, meta-learning and transfer-learning. In the context of deep learning this refers to the method of training a base network on features that are general to all tasks. Afterwards the pre-trained base network is used and the learned features are repurposed to only train a classification layer on the new tasks. However, it turned out that re-training brings up another weakness of deep neural networks, catastrophic forgetting (64). This term refers to the fact, that after a model has been trained on one task and gets re-trained on a second task, it will completely forget everything it has learned on the previous task. In this work we used a method similarly to the latter approach of transfer-learning but without any additional retraining and we used spike-based learning in an improved implementation (26) of the Multispike Tempotron (25). We predict that spike-based methods inspired by biological learning will become increasingly important for artificial intelligence.

## Materials and Methods

Code and data sets will be made available through our github profile at: `https://github.org/nawrotlab`

***Spiking network model.*** All neurons of the olfactory network are modeled as conductance-based leaky integrate-and-fire neurons with spike frequency adaptation (SFA). Specifically, the membrane potential follows the dynamical current balance equation 1. On threshold crossing a hard reset of the membrane potential is performed by 2. SFA is modeled as outward current by term 4 of equation 1. Strength of the adaptation current is modeled by a constant ($b$) decrease on each threshold crossing. Input to the model is modeled as direct, time-dependent current injection of shot-noise to all ORN neurons by the term $I_{stim}(t)$. All simulations of the network are carried out using BRIAN2 (65) simulator. The membrane potential of each neuron within a population is initialized randomly $\in [V_{rest}, V_{threshold}]$. To avoid any artifacts the network is brought to equilibrium by driving the network for 2 sec with background activity only before starting the actual simulation.

$$\begin{aligned}
C_m \frac{\mathrm{d}v}{\mathrm{d}t} &= g_l(E_l - v) \\
&\quad + g_e(E_e - v) - g_i(E_i - v) \\
&\quad - g_{Ia}(E_{Ia} - v) \underbrace{+ I_{stim}(t)}_{\text{only for ORNs}} && [1] \\
v &= V_{rest} \quad \text{on threshold crossing} && [2] \\
\tau_{Ia} \frac{\mathrm{d}g_{Ia}}{\mathrm{d}t} &= -g_{Ia} && [3] \\
g_{Ia} &= g_{Ia} - b \quad \text{on threshold crossing} && [4]
\end{aligned}$$

For this work the number of neurons within each layer and connectivity schemes are chosen to match the numbers found in the adult *Drosophila melanogaster* (14, 24). Our model comprises 2080 explicitly modeled olfactory recepter neurons (ORNs) organized in 52 different receptor types. ORNs of the same receptor type converge onto the same Glomerulus (52) by feedforward excitatory synapses. Each Glomerulus is formed by a projection neuron (PN) and local interneuron (LN). LNs provide lateral inhibition to all other PNs and LNs. PNs randomly project to a large population (2000) of Kenyon cells (KC) with excitatory synapses such that each KC on average receives input from 6 random PNs. This sparse random convergence implements population sparse responses. The single, plastic mushroom body output neuron is fully connected to all KCs.

We used the cellular mechanism of spike-frequency adaptation (SFA) to achieve temporal sparseness. ORNs are configured to have slow and weak spike-frequency adaptation in accordance with experimental findings (27, 30). For PNs and LNs SFA has been turned off and KCs are set to produce fast and strong adaptation currents (18, 66). The property of temporal sparseness can also be achieved by an alternative implementation through feedback inhibition as proposed by (53) and (67).

The synaptic weights of all connections within the network have been manually determined such that an average background firing rate of $8 - 10$ Hz is achieved in the LN population.

***Stimulus response profile of ORNs.*** The stimulus response profile of ORNs is determined by the ORN tuning curves. We follow a similar method as used in (21) where cyclical tuning over receptor types is modeled as half period sine waveforms. Our model comprises $N_{type} = 52$ receptor types and supports 52 different stimuli (e.g. different odors). Where $k_{type}$ refers to the receptor type index ($\in [0, 51]$) and $k_{odor}$ to the stimulus index ($\in [0, 51]$). $N_{orn} = 15$ determines the number of receptor types activated by a stimulus. The tuning strength $r$ of the ORNs can be computed as 0.5 cycle of a sine wave with peak amplitude $r_{max} = 1$. In the present work all tuning profiles are normalized to have a peak amplitude of 1.

$$x = \frac{k_{type} - k_{odor} \mathrm{mod} N_{type}}{N_{orn} + 1} \quad [5]$$

$$r = r_{max} \left\{ \begin{array}{ll} sin(x\pi) & \text{for } 0 < x < 1 \\ 0 & \text{else} \end{array} \right. \quad [6]$$

***Model input.*** Input to the mushroom body model is modeled as time-dependent, direct current injection into all ORN neurons. In the absence of any stimuli ORNs exhibit spontaneous activity (27). The model input thus consists of spontaneous background activity and stimulus related activity. To generate the background activity, a current time-series is generated for each ORN by simulating shot noise. For each ORN neuron, background activity events are generated from a Poisson process with high rate ($\lambda = 300$) (independent Poisson processes are drawn for each individual neuron). Events of the Poisson process are filtered by a low-pass filter with $\tau = 0.6$ sec. Using this shot-noise model is consistent with experimental findings of odor transduction at the ORNs (27). To induce stimulus related activity to this time-series of ORN $j$ it is multiplied point-wise with a stimulaton protocol time-series $s_j(t)$ which is rescaled by a constant determined by the tuning strength ($r_j \in [0, 1]$) to the specific odor of the ORN. This results in a current time series where during stimulus the current magnitude is increased proportional to the ORNs tuning strength and otherwise remains at the magnitude of the background activity.

We define a stimulation protocol function $s(t)$, which is a step function taking on the value 1 at all time points $t$ where a stimulus or sensory cue is active. For each ORN a rescaled instance of the stimulation protocol is defined as $s_j(t) = r_j s(t)$, where the scaling parameter $r_j \in [0, 1]$ is given by the stimulus response profile (eq. 6) of the ORN to the specific stimulus.

$$s(t) = \left\{ \begin{array}{ll} 1 & \text{if some stimulus is present} \\ 0 & \text{else} \end{array} \right.$$

***Sequences of sensory cues.*** Each sequence has a duration of 10 seconds. Sequences of sensory cues are generated by drawing the total number of cues within a single sequence from a Poisson distribution with mean $\lambda = 8$. Onset times of the cues between 0 and 10 seconds are drawn from a random uniform distribution and it is assured that there is no temporal overlapp between cues. A stimulus relates to a single sensory cue and its duration is drawn uniformly between $[1, 200]$ milliseconds. Finally, each sensory cue is associated with a random odor drawn from a fixed set of possible odors (random sampling with replacement and equal probability). This results in sequences with random number of sensory cues, random onset, random duration and randomized odor and distractor combinations.

***Model of sensory cues within (gaussian) plume.*** The same procedure is used as above to simulate the experience of sensory cues during a single casting trajectory within a turbulent odor plume. The number of pleasant cues experienced in a casting trajectory is drawn from a Poisson distribution with mean $\lambda = 14$. The cue onset times are drawn from a gaussian distribution with $\mu = 5, \sigma = 1.5$. The number of distractor cues is drawn from a Poisson distribution with mean $\lambda = 5$ and are distributed uniformly in time. Duration of both, pleasant and distractor cues, is drawn uniformly between $[100, 500]$ ms. In total 200 different casting trajectories have been generated using this procedure.

***Readout Neuron & Learning rule.*** To fit the readout neuron to the stimuli such that it generates 1 spike for pleasant odor stimuli (CS+) and 0 spikes for any other stimuli (CS-) we use a modified implementation of the Multispike Tempotron (25, 26). Thus, the readout neuron is modeled as voltage-based leaky integrate-and-fire neuron with soft reset following the dynamical equation 7. Incoming spikes evoke exponentially decaying post-synaptic potentials. When the membrane potential reaches the spiking threshold at some time $t_0$ an output spike is generated and the membrane potential is reset by the last term of equation 7.

$$V(t) = \underbrace{V_{rest}}_{:=0} + \sum_{i=1}^{N} \omega_i \sum_{t_i^j < t} \overbrace{K(t - t_i^j)}^{\text{exp. PSP kernel}} \quad [7]$$

$$- (\underbrace{\vartheta}_{:=1} - V_{rest}) \sum_{t_{sp}^j} e^{-\frac{t - t_{sp}^j}{\tau_m}}$$

The dynamical equation can be decomposed into two parts, the unreset sub-threshold potential $V_0(t)$ (eq. 8) minus the remaining terms for the soft-reset. The neuron is trained to generate 1 spike for pleasant odor stimuli (CS+1) and 0 spikes for any other stimuli (CS-). To fit the desired neural code, a training step is performed after each stimulus presentation. A training step is performed only if the number of spikes generated in response to a stimulus was not correct. The training target is given by the difference between number of output spikes the model generated and the number of output spikes associated with the stimulus. We denote the desired critical threshold value, the voltage value that generates $d = 1$ spike, as $\vartheta^*$ and the time point where this voltage value is reached by $t^*$ (more generally: the critical threshold value to generate $d$ spikes). We briefly sketch the idea and intuition of the Multispike Tempotron learning rule. For detailed derivation of the rule we refer to (25) and the section *The $\vartheta^*$ gradient*. The Multispike Tempotron training algorithm works by differentiating the membrane potential of the the critical threshold wrt. to the synaptic weights ($\vec{\omega}$). This can be done since $\vartheta^*$ is a regular voltage value, that can be expressed by the neuron's dynamical equation Eq. (7), with the special identities shown in equation 10. This allows to take the full derivative as shown in equation 11.

$$V_0(t) = \sum_{i=1}^{N} \omega_i \sum_{t_i^j < t} K(t - t_i^j) \quad \textit{unreset sub-thresh. potential} \quad [8]$$

$$V(t) = V_0(t) - \vartheta \sum_{t_{sp}^j} e^{-\frac{t - t_{sp}^j}{\tau_m}} \quad [9]$$

$$\vartheta^* = V(t^*) = V(t_{sp}^j) \quad \text{critical thresh. that makes d spikes} \quad [10]$$

$$\nabla_{\vec{\omega}} \vartheta^* = \frac{\partial}{\partial \omega} V(t^*) + \sum_{j=1}^{m} \frac{\partial}{\partial t_{sp}^j} V(t^*) \frac{\mathrm{d}}{\mathrm{d}\omega} t_{sp}^j \quad [11]$$

The gradient of the critical threshold with respect to a single synapse $i$ is given by equation 12.

$$(\vartheta_i^*)' = \frac{\mathrm{d}}{\mathrm{d}\omega_i} \vartheta^* = \frac{\mathrm{d}}{\mathrm{d}\omega_i} V(t^*) = \frac{\mathrm{d}}{\mathrm{d}\omega_i} V(t_{sp}^j) \quad [12]$$

$$(\vartheta_i^*)' = \frac{\partial}{\partial \omega_i} V(t^*) + \sum_{j=1}^{m} \frac{\partial}{\partial t_{sp}^j} V(t^*) \frac{\mathrm{d}}{\mathrm{d}\omega_i} t_{sp}^j \quad \text{recursive expr. exists} \quad [13]$$

**ACKNOWLEDGMENTS.** This research is supported by the German Research Foundation (grant no. 403329959 to MN) within the Research Unit 'Structure, Plasticity and Behavioral Function of the Drosophila Mushroom Body' (DFG-FOR 2705, www.uni-goettingen.de/en/601524.html).

1. KL Baker, et al., Algorithms for olfactory search across species. *J. Neurosci.* **38**, 9383–9389 (2018).
2. J Murlis, JS Elkinton, RT Cardé, Odor plumes and how insects use them. *Annu. Rev. Entomol.* **37**, 505–532 (1992).
3. JP Crimaldi, MB Wiley, JR Koseff, The relationship between mean and instantaneous structure in turbulent passive scalar plumes. *J. Turbul.* **3**, 1–24 (2002).
4. A Celani, Odor landscapes in turbulent environments. *Phys. Rev. X* **4** (2014).
5. EG Connor, MK McHugh, JP Crimaldi, Quantification of airborne odor plumes using planar laser-induced fluorescence. *Exp. Fluids* **59**, 137 (2018).

6. N Vickers, T Christensen, T Baker, J Hildebrand, Odour-plume dynamics influence file brain's olfactory code. *Nature* **410**, 466–470 (2001).
7. SD Boie, et al., Information-theoretic analysis of realistic odor plumes: What cues are useful for determining location? *PLOS Comput. Biol.* **14**, e1006275 (2018).
8. F van Breugel, MH Dickinson, Plume-tracking behavior of flying drosophila emerges from a set of distinct sensory-motor reflexes. *Curr. Biol.* **24**, 274–286 (2014).
9. JA Riffell, et al., Flower discrimination by pollinators in a dynamic chemical environment. *Science* **344**, 1515–1518 (2014).
10. R Pang, F van Breugel, M Dickinson, JA Riffell, A Fairhall, History dependence in insect flight decisions during odor tracking. *PLOS Comput. Biol.* **14**, e1005969 (2018).
11. M Vergassola, E Villermaux, BI Shraiman, "infotaxis" as a strategy for searching without gradients. *Nature* **445**, 406–409 (2007).
12. B Baddeley, et al., What can be learnt from analysing insect orientation flights using probabilistic slam? *Biol. Cybern.* **101**, 169–182 (2009).
13. JB Masson, Olfactory searches with limited space perception. *Proc. Natl. Acad. Sci.* **110**, 11261–11266 (2013).
14. Y Aso, et al., The neuronal architecture of the mushroom body provides a logic for associative learning. *Elife* **3**, e04577 (2014).
15. SJ Caron, V Ruta, L Abbott, R Axel, Random convergence of olfactory inputs in the drosophila mushroom body. *Nature* **497**, 113 (2013).
16. CS Xu, et al., A connectome of the adult drosophila central brain. *bioRxiv* (2020).
17. I Ito, RCy Ong, B Raman, M Stopfer, Sparse odor representation and olfactory learning. *Nat. Neurosci.* **11**, 1177 EP – (2008).
18. H Demmer, P Kloppenburg, Intrinsic membrane properties and inhibitory synaptic input of kenyon cells as mechanisms for sparse coding? *J. neurophysiology* **102**, 1538–1550 (2009).
19. P Szyszka, RC Gerkin, CG Galizia, BH Smith, High-speed odor transduction and pulse tracking by insect olfactory receptor neurons. *Proc. Natl. Acad. Sci.* **111**, 16925–16930 (2014).
20. A Litwin-Kumar, KD Harris, R Axel, H Sompolinsky, L Abbott, Optimal degrees of synaptic connectivity. *Neuron* **93**, 1153 – 1164.e7 (2017).
21. R Betkiewicz, B Lindner, MP Nawrot, Circuit and cellular mechanisms facilitate the transformation from dense to sparse coding in the insect olfactory system. *eNeuro* **7** (2020).
22. J Hintikka, A hundred years later: The rise and fall of frege's influence in language theory. *Synthese*, 27–49 (1984).
23. GC Turner, M Bazhenov, G Laurent, Olfactory representations by drosophila mushroom body neurons. *J. neurophysiology* **99**, 734–746 (2008).
24. Sy Takemura, et al., A connectome of a learning and memory center in the adult drosophila brain. *eLife* **6** (2017).
25. R Gütig, Spiking neurons can discover predictive features by aggregate-label learning. *Science* **351** (2016).
26. H Rapp, MP Nawrot, M Stern, Numerical cognition based on precise counting with a single spiking neuron. *iScience*, 100852 (2020).
27. KI Nagel, RI Wilson, Biophysical mechanisms underlying olfactory receptor neuron dynamics. *Nat. neuroscience* **14**, 208 (2011).
28. M Stopfer, V Jayaraman, G Laurent, Intensity versus identity coding in an olfactory system. *Neuron* **39**, 991–1004 (2003).
29. RI Wilson, GC Turner, G Laurent, Transformation of olfactory representations in the drosophila antennal lobe. *Science* **303**, 366–370 (2004).
30. S Krofczik, R Menzel, MP Nawrot, Rapid odor processing in the honeybee antennal lobe network. *Front. computational neuroscience* **2**, 9 (2009).
31. H Watanabe, H Ai, F Yokohari, Spatio-temporal activity patterns of odor-induced synchronized potentials revealed by voltage-sensitive dye imaging and intracellular recording in the antennal lobe of the cockroach. *Front. systems neuroscience* **6**, 55 (2012).
32. E Gruntman, GC Turner, Integration of the olfactory code across dendritic claws of single mushroom body neurons. *Nat. neuroscience* **16**, 1821 (2013).
33. RA Jortner, SS Farivar, G Laurent, A simple connectivity scheme for sparse coding in an olfactory system. *J. Neurosci.* **27**, 1659–1669 (2007).
34. RI Wilson, Early olfactory processing in drosophila: mechanisms and principles. *Annu. review neuroscience* **36**, 217–241 (2013).
35. SX Luo, R Axel, LF Abbott, Generating sparse and selective third-order responses in the olfactory system of the fly. *PNAS; Proc. Natl. Acad. Sci.* **107**, 10713–10718 (2010).
36. AC Lin, AM Bygrave, A De Calignon, T Lee, G Miesenböck, Sparse, decorrelated odor coding in the mushroom body enhances learned odor discrimination. *Nat. neuroscience* **17**, 559 (2014).
37. R Huerta, T Nowotny, Fast and robust learning by reinforcement signals: explorations in the insect brain. *Neural computation* **21**, 2123–2151 (2009).
38. J Wessnitzer, JM Young, JD Armstrong, B Webb, A model of non-elemental olfactory learning in drosophila. *J. computational neuroscience* **32**, 197–212 (2012).
39. P Ardin, F Peng, M Mangan, K Lagogiannis, B Webb, Using an insect mushroom body circuit to encode route memory in complex natural environments. *PLoS computational biology* **12**, e1004683 (2016).
40. F Peng, L Chittka, A simple computational model of the bee mushroom body can explain seemingly complex forms of olfactory learning and memory. *Curr. Biol.* **27**, 224–230 (2017).
41. J Müller, M Nawrot, R Menzel, T Landgraf, A neural network model for familiarity and context learning during honeybee foraging flights. *Biol. cybernetics* **112**, 113–126 (2018).
42. KS Honegger, Raa Campbell, GC Turner, Cellular-resolution population imaging reveals robust sparse coding in the Drosophila mushroom body. *The J. neuroscience : official journal Soc. for Neurosci.* **31**, 11772–85 (2011).
43. M Bitterman, R Menzel, A Fietz, S Schäfer, Classical conditioning of proboscis extension in honeybees (apis mellifera). *J. comparative psychology* **97**, 107 (1983).
44. E Pamir, P Szyszka, R Scheiner, MP Nawrot, Rapid learning dynamics in individual honeybees during classical conditioning. *Front. Behav. Neurosci.* **8** (2014).
45. L Scheunemann, et al., Akaps act in a two-step mechanism of memory acquisition. *J. Neurosci.* **33**, 17422–17428 (2013).
46. D Owald, et al., Activity of defined mushroom body output neurons underlies learned olfactory behavior in drosophila. *Neuron* **86**, 417 – 427 (2015).
47. MF Strube-Bloss, MP Nawrot, R Menzel, Mushroom body output neurons encode odor–reward associations. *J. Neurosci.* **31**, 3129–3140 (2011).
48. Y Aso, et al., Mushroom body output neurons encode valence and guide memory-based action selection in drosophila. *Elife* **3**, e04580 (2014).
49. MF Strube-Bloss, MP Nawrot, R Menzel, Neural correlates of side-specific odour memory in mushroom body output neurons. *Proc. Royal Soc. B: Biol. Sci.* **283**, 20161270 (2016).
50. A Sehdev, P Szyszka, Segregation of unknown odors from mixtures based on stimulus onset asynchrony in honey bees. *Front. Behav. Neurosci.* **13** (2019) Article Number: 155.
51. A Grabska-Barwińska, et al., A probabilistic approach to demixing odors. *Nat. Neurosci.* **20**, 98 EP – (2016).
52. A Sehdev, YG Mohammed, T Triphan, P Szyszka, Olfactory object recognition based on fine-scale stimulus timing in drosophila. *iScience* **13**, 113–124 (2019).
53. C Assisi, M Stopfer, M Bazhenov, Optimality of sparse olfactory representations is not affected by network plasticity. *PLOS Comput. Biol.* **16**, e1007461 (2020).
54. CB Delahunt, JN Kutz, Putting a bug in ml: The moth olfactory network learns to read mnist. *Neural Networks* **118**, 54 – 64 (2019).
55. T Hromádka, MR DeWeese, AM Zador, Sparse representation of sounds in the unanesthetized auditory cortex. *PLoS Biol.* **6**, e16 (2008).
56. WE Vinje, Sparse coding and decorrelation in primary visual cortex during natural vision. *Science* **287**, 1273–1276 (2000).
57. J Wolfe, AR Houweling, M Brecht, Sparse and powerful cortical spikes. *Curr. opinion neurobiology* **20**, 306–312 (2010).
58. JS Isaacson, Odor representations in mammalian cortical circuits. *Curr. opinion neurobiology* **20**, 328–331 (2010).
59. BN Lundstrom, S Hong, MH Higgs, AL Fairhall, Two computational regimes of a single-compartment neuron separated by a planar boundary in conductance space. *Neural Comput.* **20**, 1239–1260 (2008).
60. S Ratté, M Lankarany, YA Rho, A Patterson, SA Prescott, Subthreshold membrane currents confer distinct tuning properties that enable neurons to encode the integral or derivative of their input. *Front. Cell. Neurosci.* **8** (2015).
61. BP Tripp, C Eliasmith, Population models of temporal differentiation. *Neural Comput.* **22**, 621–659 (2010) PMID: 19922294.
62. J Schmidhuber, Deep learning in neural networks: An overview. *Neural Networks* **61**, 85–117 (2015).
63. TJ Sejnowski, The unreasonable effectiveness of deep learning in artificial intelligence. *PNAS; Proc. Natl. Acad. Sci.* (2020).
64. J Kirkpatrick, et al., Overcoming catastrophic forgetting in neural networks. *Proc. Natl. Acad. Sci.* **114**, 3521–3526 (2017).
65. M Stimberg, R Brette, DF Goodman, Brian 2, an intuitive and efficient neural simulator. *eLife* **8**, e47314 (2019).
66. DG Wustenberg, et al., Current-and voltage-clamp recordings and computer simulations of kenyon cells in the honeybee. *J. neurophysiology* **92**, 2589–2603 (2004).
67. JEM Bennett, A Philippides, T Nowotny, Learning with reward prediction errors in a model of the drosophila mushroom body. *bioRxiv* (2019).

## 2.3 Area-specific processing of cerebellar-thalamo-cortical information in primates.

Biol Cybern
DOI 10.1007/s00422-017-0738-6

ORIGINAL ARTICLE

Biological Cybernetics

# Area-specific processing of cerebellar-thalamo-cortical information in primates

**Abdulraheem Nashef**[1] · **Hannes Rapp**[2] · **Martin P. Nawrot**[2] · **Yifat Prut**[1,3]

Received: 28 February 2017 / Accepted: 12 October 2017

**Abstract** The cerebellar-thalamo-cortical (CTC) system plays a major role in controlling timing and coordination of voluntary movements. However, the functional impact of this system on motor cortical sites has not been documented in a systematic manner. We addressed this question by implanting a chronic stimulating electrode in the superior cerebellar peduncle (SCP) and recording evoked multiunit activity (MUA) and the local field potential (LFP) in the primary motor cortex ($n = 926$), the premotor cortex ($n = 357$) and the somatosensory cortex ($n = 345$). The area-dependent response properties were estimated using the MUA response shape (quantified by decomposing into principal components) and the time-dependent frequency content of the evoked LFP. Each of these signals alone enabled good classification between the somatosensory and motor sites. Good classification between the primary motor and premotor areas could only be achieved when combining features from both signal types. Topographical single-site representation of the predicted class showed good recovery of functional organization. Finally, the probability for misclassification had a broad topographical organization. Despite the area-specific response features to SCP stimulation, there was considerable site-to-site variation in responses, specifically within the motor cortical areas. This indicates a substantial SCP impact on both the primary motor and premotor cortex. Given the documented involvement of these cortical areas in preparation and execution of movement, this result may suggest a CTC contribution to both motor execution and motor preparation. The stimulation responses in the somatosensory cortex were sparser and weaker. However, a functional role of the CTC system in somatosensory computation must be taken into consideration.

**Keywords** Motor control · Thalamocortical · Cerebellum · Multiunit activity · Local field potential · Machine learning

## 1 Introduction

The cerebellum is considered to play a prominent role in dictating motor timing and coordination (Holmes 1939; Ivry and Keele 1989). Consistent with this theory, cerebellar patients often exhibit poorly timed and uncoordinated movements (Spencer et al. 2003). In primates, the motor cortex may well be the most significant target of the cerebellum (Asanuma et al. 1983a, b). Specifically, the cerebellar-thalamo-cortical (CTC) system was found to be a prominent route through which the cerebellum can shape motor cortical activity and subsequently motor behavior (Hore and Flament 1988; Ivanusic et al. 2005; Meyer-Lohmann et al. 1975). The system originates in the deep cerebellar nuclei and is relayed

Martin P. Nawrot and Yifat Prut shared senior authorship.

This article belongs to a Special Issue on Neural Coding.

**Electronic supplementary material** The online version of this article (https://doi.org/10.1007/s00422-017-0738-6) contains supplementary material, which is available to authorized users.

✉ Yifat Prut
yifatpr@ekmd.huji.ac.il

Martin P. Nawrot
mnawrot@uni-koeln.de

[1] Department of Medical Neurobiology, IMRIC, Hadassah Medical School, The Hebrew University, 91120 Jerusalem, Israel

[2] Computational Systems Neuroscience, Institute of Zoology, University of Cologne, Cologne, Germany

[3] Edmond and Lily Safra Center for Brain Sciences, The Hebrew University, 91904 Jerusalem, Israel

Published online: 02 November 2017

via the motor thalamus, where it makes particularly effective synaptic contacts (Aumann et al. 1994), to finally terminate in the motor cortex.

Anatomical studies have explored the detailed organization of thalamic termination onto the motor cortex. It was shown that the origins of the thalamic input to the premotor areas derived from different sources compared to those that innervate the primary motor areas (Orioli and Strick 1989; Schell and Strick 1984; Strick 1986). Further, the CTC pathway also terminates in somatosensory areas that are adjacent to the central sulcus, i.e., area 3a (Padberg et al. 2009). These cortical areas, which are the main target of the CTC system, exhibit differential involvement in motor control. The primary motor cortex is strongly related to motor execution and encoding of time-dependent movement parameters (Moran and Schwartz 1999; Morrow et al. 2007; Rickert et al. 2009; Scott et al. 2001; Shalit et al. 2012), whereas the premotor areas were shown to express specifically strong preparatory activity (Godschalk et al. 1985; Weinrich and Wise 1982). Area 3a was shown to take part in proprioception (Hore et al. 1976; Padberg et al. 2007; Phillips et al. 1971; Wang et al. 2007) and is interconnected with the motor cortical areas (Corinna Darian-Smith et al. 1993; Huffman and Krubitzer 2001; Jones et al. 1978). Hence the response pattern of the CTC system in these areas is expected to have different spatiotemporal characteristics.

To test this hypothesis, we implanted a stimulating electrode in the superior cerebellar peduncle (SCP) and recorded multiple unit activity (MUA) and the local field potential (LFP) in motor and sensory areas of 3 monkeys. We found significant responses to SCP stimulation in the motor and sensory areas. A classifier that was trained on the dataset was able to distinguish responses in the motor from those in the sensory areas using the power content of the evoked LFP or the response pattern of the MUA. Both signal types provided a similar success rate, but there was no improvement in performance when using these signals together. By contrast, the classifier poorly identified M1 versus PM sites, indicating that despite the anatomical differences, the two areas receive very similar input. A possible interpretation of this finding is that the CTC system affects both the preparation and execution of motor commands.

## 2 Methods

### 2.1 Data acquisition

Data were obtained from three *Macaca fascicularis* monkeys (2 females and a male, 3.5–7 kg). Monkey care and surgical procedures were in accordance with the Hebrew University Guidelines for the Use and Care of Laboratory Animals in Research, supervised by the Institutional Committee for Animal Care and Use. Two monkeys were trained to sit in a primate chair and perform a two-dimensional isometric wrist task, as previously described (Yanai et al. 2007, 2008). A third monkey (monkey C) was trained to perform a planar reaching task by wearing an exoskeleton (Kinarm by Bkin).

After training, a recording chamber (21 × 21 mm) was attached to the skull of the monkeys above the hand-related motor cortex in a surgical procedure under general anesthesia. After a recovery and re-training period, we made extracellular recordings of motor cortical activity. During recording sessions, glass-coated tungsten electrodes (impedance 300–800 kΩ at 1 kHz) were inserted through the chamber to different cortical sites, mostly in the primary motor cortex (M1). The signal obtained from each electrode was amplified (×10 K) and fed through two different online bandpass filters (300–6000 Hz for the single-unit data and 1–250 Hz for the LFP). The signal was then digitized at different sampling rates for the two signals (single unit: 25–32 kHz, LFP: 1 kHz).

### 2.2 Insertion of superior cerebellar peduncle (SCP) stimulating electrode

After implanting the cortical chamber over the hemisphere contralateral to the working hand, we used an MRI scan of the monkey to plan the trajectory for a stimulating electrode to reach the ipsilateral SCP. In one monkey, insertion of stimulating electrodes (Pt–Ir parylene-coated monopolar electrodes, We-Sense LTD, Nazareth, ISRAEL) was done under anesthesia and the location was verified using a second, post-procedure MRI scan. In the other two monkeys, we implanted a small chamber above the estimated insertion point. The trajectory was planned based on an MRI scan, and a bipolar concentric electrode was inserted (NSEX100, David Kopf Instruments). These two different methods of implantation yielded very consistent results in terms of cortical and thalamic response pattern and response frequency.

### 2.3 Cortical mapping

Each recording site was mapped based on the motor response evoked when applying intra-cortically a brief train of microstimuli through the recording electrodes (50 ms of biphasic pulses applied at 300 Hz). Sites where the threshold level for obtaining the motor response was below 15 μA were defined as sites within the primary motor cortex (M1). Sites in which motor responses were obtained at high threshold levels and were located more than 3 mm anterior to the central sulcus were defined as premotor sites (PM). Finally, sites located posterior to the central sulcus and required either high-amplitude stimulation to obtain a motor response or produce no response at all were defined as somatosensory sites (SS).

## 2.4 SCP stimulation

We tested each recording site for its neural response to SCP stimulation by applying biphasic stimulation pulses (each phase was 0.2 ms in duration) through the SCP electrode. A single set of stimuli was composed of about 200 pulses delivered at 3 Hz (i.e., a 333-ms time interval between successive single stimuli). Each site was tested at 2–3 levels of stimulus intensities (in the range of 50–400 $\mu$A). A detailed description of the range of intensities used for each monkey is provided in Supplemental Table S2. For the purpose of this analysis for each session, we only considered stimuli that were applied at maximal stimulation intensity when computing the stimulus-triggered averages. This was done to obtain the most prominent response available for each site.

## 2.5 Data analysis

### *2.5.1 Stimulus-triggered average of MUA*

To extract the multiunit activity (MUA) from the full signal recorded at each cortical site, we first removed the stimulus artifacts caused by the SCP stimuli using a previously described algorithm (Ruach et al. 2015).

After the artifact removal stage, the compound signal was high-pass-filtered with a 1 kHz infinite impulse response (IIR) filter. This was done to make sure the signal did not include remains of the local field potential signal. The signal was then rectified, and the stimulus-triggered response was computed (from − 5 to + 100 ms around stimulus time). The response was normalized by subtracting the prestimulus baseline level and then dividing by the standard deviation computed for the prestimulus baseline. The normalized response was smoothed by convolution with a Gaussian window spanning 5 ms and subsequently down-sampled to 8 kHz resolution.

### *2.5.2 Computing the stimulus-triggered average of LFP*

The first step in the LFP analysis was manual verification of signal stability. Recording sites in which the LFP was substantially contaminated by large amplitude noise were removed from further analysis. Previously, we found that the stimulus artifact in the LFP signal was substantially smaller than in the MUA signal and there was no need for an explicit artifact removal process. For each site, the LFP data were low-pass-filtered offline using an eight-pole Butterworth low-pass filter with a cutoff frequency of 250 Hz. We then computed the stimulus-triggered LFP response (− 50 to + 250 ms around stimulus onset time). In the final stage, we calculated the standard score (or $z$-score) of the LFP traces as follows: from the average LFP data, we subtracted the baseline activity (averaged across 10 ms before the stimulation onset) and divided it by the standard deviation in the baseline period.

### *2.5.3 Statistical data analysis*

After data acquisition and preprocessing, a statistical analysis of the data was performed using Python 3 and recent versions of the NumPy, SciPy and sklearn packages. We pooled the recording sites separately for the three cortical areas M1, PM, SS in all three monkeys. For the LFP responses, we conducted a time–frequency analysis, whereas for MUA responses, which consist solely of high-frequency components, a principal component analysis (PCA) was used to identify the temporal components that varied the most.

For the time–frequency analysis, the standard scores of the LFP responses (Fig. 1a–c) were decomposed into the time–frequency domain by a 32-ms sliding (Hanning) window (overlap 30 ms). A short-time fast Fourier transform (STFFT) was used to obtain a spectrogram of signal amplitudes (i.e., the square root of the signal power). The spectrograms were averaged across recording sites within each cortical area. To highlight the stimulus-triggered changes in the time–frequency domain, normalized spectrograms were computed by dividing each spectrogram by the same time-averaged prestimulus spectrogram computed from all recordings. For the classification analysis, we used the LFP spectrogram in a window from 0 to 45 ms after stimulus onset since the spectrograms did not show prolonged change in relative power from 45 ms onwards (cf. Supplemental Figure S1).

The standard scored MUA responses were decomposed into principal components by singular value decomposition (SVD). The single-site MUA response vectors (Fig. 2d) were decomposed into principal components, which can be interpreted as a prototype time series. For area-specific classification, the MUA responses were projected onto the first 3 PCs and cropped to a window of 0–15 ms after the stimulus to obtain the three-dimensional PC features.

### *2.5.4 Area-specific response classification*

For the area-specific classification of LFP and MUA responses, the same random forest (RF) classifier with 15 base estimators was used throughout as implemented by the Python sklearn package. The dataset contained 1619 responses in total with 932 labeled instances of M1, 352 of PM and 343 of SS. The random forest classifier is an ensemble method using instances of classification and regression trees (CART) as

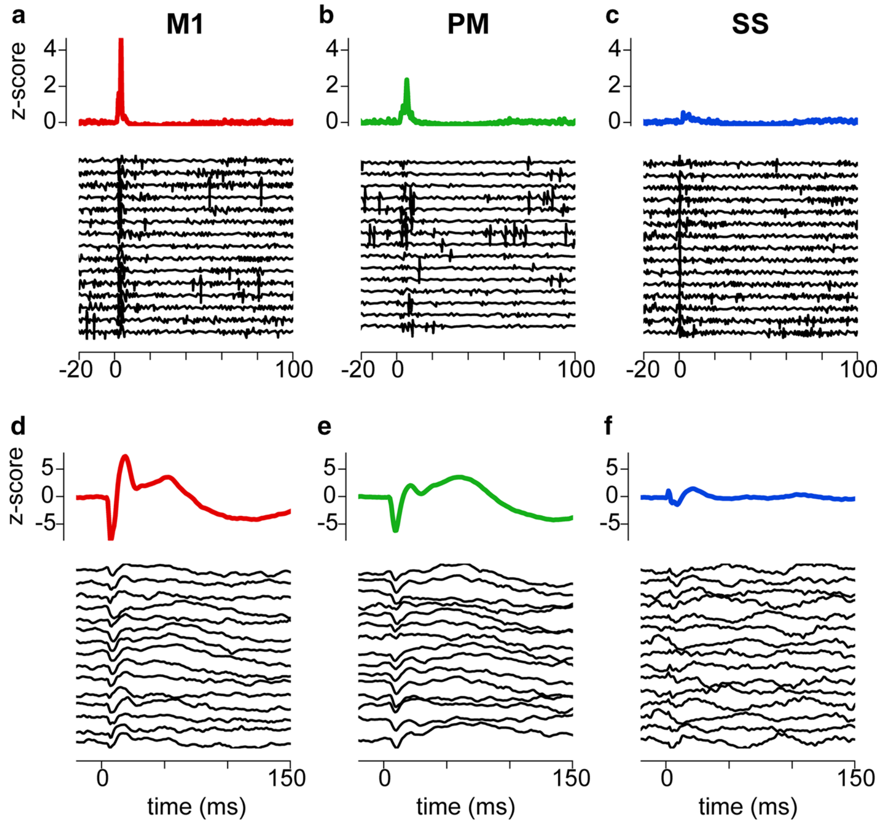


**Fig. 1** Evoked response to SCP stimulation in different cortical areas. **a–c** Single traces and mean responses of MUA for cortical areas M1 (**a** red), PM (**b** green) and SS (**c** blue). **d–f** Responses of LFP recorded from the same sites as MUA data. Each response was normalized to $z$-score values by subtracting the prestimulus baseline level and dividing by the standard deviation computed for the same time range (color figure online)

base estimators (Breiman 2001; Breiman et al. 1984). CART models subsequently partition the feature space into a set of lower-dimensional subspaces and then fit a simple model (e.g., a constant) to each subspace. The choice of which features to use and the specific split to grow the tree is made by a greedy algorithm to minimize a cost function. Tree construction ends at a predefined stopping criterion. Random forests apply bootstrap aggregating (bagging) where repeatedly, a random subsample with replacement of the training set is used to fit a simple individual CART model. To decorrelate the bagged trees, the algorithm only uses a random sample of features for splitting. For an unseen sample, the predicted class is determined by the mode of the classes of the individual trees.

We chose 18 base estimators with a depth of 5. Tree-based classifiers have a built-in feature selection and are fast to train and robust against over-fitting and class imbalance, which made them a good choice for this particular dataset. To evaluate the classification performance, the tenfold cross-validation method was used to calculate the cross-validated confusion matrices and the Matthews correlation coefficient metric for 2-category and k-category problems (Gorodkin 2004; Matthews 1975). To obtain the cortical maps, a leave-one-out cross-validation was applied to obtain predictions for each single recording site.

## 3 Results

Data were collected from the motor and somatosensory cortical areas in three monkeys (923 M1 sites, 352 PM sites and 343 SS sites). Of all the recorded M1 sites, 86.8% expressed a significant MUA response to SCP stimulation, whereas for PM and SS sites 89.8 and 38.2% of recorded sites were significantly responsive to SCP stimulation, respectively. Figure 1 presents examples of responses obtained from three single sites: M1 (red), PM (green) and SS (blue). A clear response can be seen for both the single trace data and the mean response.

### 3.1 Area-specific response properties of MUA

The averaged MUA response to SCP stimulation exhibited specific characteristics across the different recording areas (Fig. 2a–c). For example, the average response evoked in M1 demonstrated particularly high amplitude at short latency that was followed by a long period of inhibition. Responses in the SS cortex clearly had lower average amplitude and showed less pronounced inhibition. However, the single-site evoked responses varied considerably in shape and latency when compared to the mean response (Fig. 2d). To quantify the systematic variation in evoked MUA by SCP stimula-

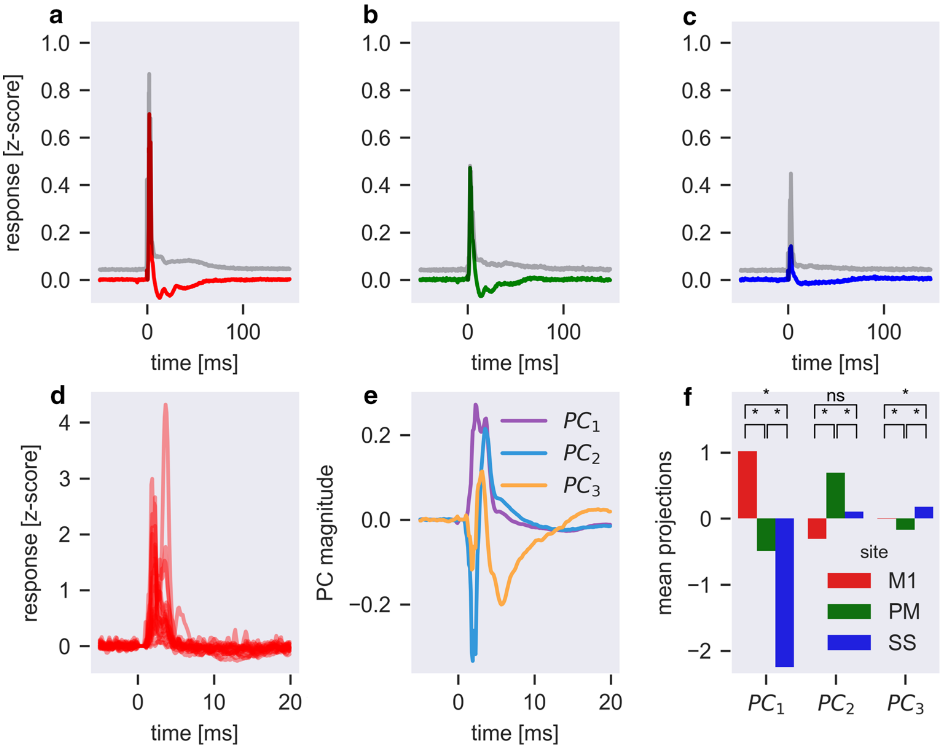


**Fig. 2** MUA responses to CTC stimulation differ across cortical areas M1, PM, SS. **a**–**c** Mean responses of MUA for cortical areas M1 (red), PM (green) and SS (blue) averaged across trials and recording sites and respective standard deviations (gray). **d** Trial-averaged MUA responses from 20 randomly chosen recording sites in M1 indicate a considerable variability across single-site response profiles. **e** The first 3 principal components (PC1–PC3) as computed across response profiles from all single recording sites pooled across cortical areas. **f** Projections of response profiles onto PC1–PC3 for cortical areas M1 (red), PM (green) and SS (blue). Each bar indicates the average magnitude of the projection coefficients for a specific PC computed from all individual recording sites in one cortical area. Asterisk indicates a significant difference ($P < 0.01$, two-sided Wilcoxon rank-sum tests) for PC projections between the respective cortical areas (color figure online)

tion across the different recording areas, we decomposed all the single-site data using principal component analysis. The first three PCs (which accounted for 30.3% of the total variance) are shown in Fig. 2e. From the fourth PC onwards, the explained variance per component was < 4% and components beyond the ninth PC each explained < 1% of the variance and thus were not taken into consideration. For each individual response, we computed its projection onto PC1–PC3. Figure 2f shows the average coefficients separately for all three cortical areas. On average, responses in each site appeared to be composed of a unique linear combination of the three PCs. Pairwise testing for the distribution of coefficients was significant ($P < 0.01$, Wilcoxon rank-sum test) for 8 out of the 9 combinations (Fig. 2f), indicating good separability of individual responses according to the respective cortical area.

To further quantify the area-specific response differences, we applied a machine learning approach. We chose the random forest (RF) classifier method (see Sect. 2) for this task because it automatically performs feature selection and allows for a meaningful interpretation of the most discriminant features. We trained and tested the classifier on the MUA response from two cortical areas (pairwise classification). As features, we used the projections onto the first three principal components. Figure 3 summarizes the classifier's performance for all three combinations of the two cortical areas as evaluated by tenfold cross-validation (see Sect. 2). We quantified performance by the confusion matrices (Fig. 3a–c). Although the classifier performed reasonably well when classifying response samples from M1 versus SS (Fig. 3b) and PM versus SS (Fig. 3c) with a correct class prediction in the range of 68–88%, performance was strikingly poor for the classification of M1 versus PM responses (Fig. 3a) with 95% of all samples assigned to M1 (predicted class) irrespective of their true class origin (M1 or PM). We further evaluated the relative importance (or weight) of features in the decision tree. Figure 3d shows the projections onto the first three principal components with weights. Each feature made a unique contribution to the classifier's performance pairwise when distinguishing between different cortical areas. Specifically, we found that the third PC contributed the most to separating responses from M1 versus

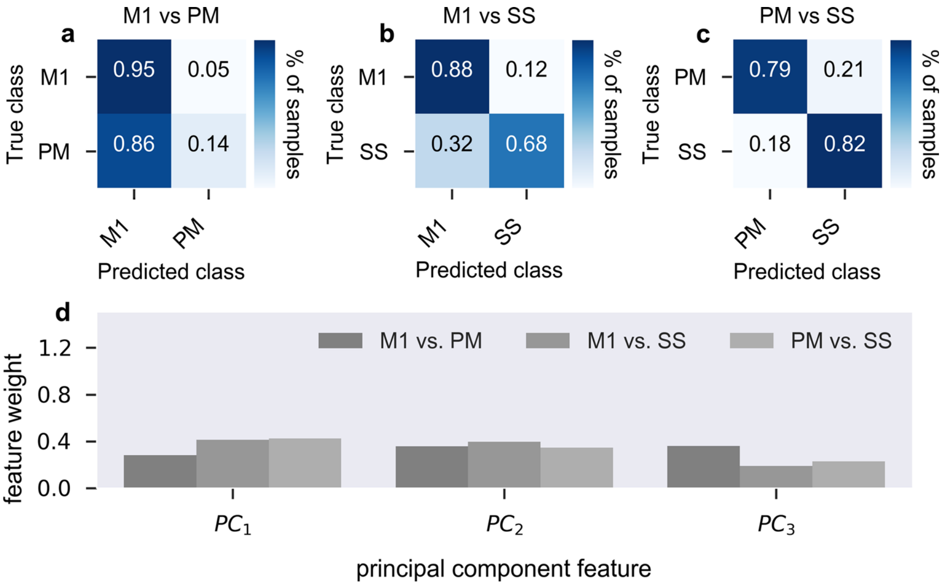


**Fig. 3** Pairwise decoding of cortical area from individual MUA responses. **a**–**c** Performance of pairwise classification of MUA responses for any two cortical areas (two-class problem). **a** Confusion matrices reveal that with a 91% false-positive rate (off-diagonal entries) discrimination between the two motor areas M1 and PM is poor. **b**, **c** Good discriminability of motor cortical (M1, PM) and somatosensory responses with low false-positive rates. **d** Classification in **a**–**c** was based on PCs. Bars represent the relative importance of the three PCs (features) with the highest discriminative power as revealed by pairwise classification with tenfold cross-validated decision trees

those obtained from PM areas. However, the first PC was the most informative in classifying both M1 versus SS as well as PM versus SS sites. Using more than the first 3 PCs did not improve classification results, and using the first 3 PCs yielded better performance than for any combination of only 2 PCs (see Supplemental Table S3).

### 3.2 Area-specific time–frequency components of the LFP signal

The mean response of the LFP to SCP stimulation (Fig. 4a–c) was substantially longer than the one found for the MUA data. This was expected since LFP is considered to represent synaptic currents, not all of which map into action potentials. This signal is further assumed to average activity over larger cortical areas than MUA data. A time–frequency analysis (see Sect. 2) of the evoked LFP (Fig. 4d) revealed on average a clear stimulus-induced increase in signal amplitude in the lower frequency ranges around 20 ms after stimulus onset. To compare the time–frequency components across the three cortical areas, we normalized the SCP-induced response spectrogram to the time-averaged amplitudes just before stimulation (see Sect. 2). The normalized spectrograms revealed a response over a broad frequency range with characteristics that appeared specific to M1, PM and SS (Fig. 4e–g). The standard deviation across all three spectrograms in Fig. 4f highlights the time–frequency domains that showed the most variation across cortical areas.

Next, we used the time–frequency components of the evoked LFP for pairwise classification of the responses at single recording sites (Fig. 5) with respect to the cortical area from which they originated. We used the same random forest method and cross-validation procedure as for the classification of MUA responses. As input features, we used all bins (bin width 2 ms) of the single-site spectrograms in the range of 0–45 ms (see Sect. 2).

In line with the results of the MUA classification (Fig. 3), we obtained good classification performance with correct class prediction in the range of 77–94% for the separation of motor areas (M1, PM) from the somatosensory cortex, as is evident from the confusion matrices in Fig. 5b, c. However, the classifier performed poorly in separating M1 and PM responses and misclassified about half the PM responses as M1 responses (confusion matrix in Fig. 5a).

The relative importance of each bin in the spectrogram (i.e., each single feature) for the correct classification was evaluated. This confirmed that the frequency content of the LFP provided little information for correctly classifying M1 versus PM responses (Fig. 5d). For the correct classification of motor responses (M1, PM) versus SS, the most pertinent contribution was obtained for time–frequency bins that corresponded to high gamma frequencies around 10–15 ms after stimulation (Fig. 5e) or a few ms later (Fig. 5f).

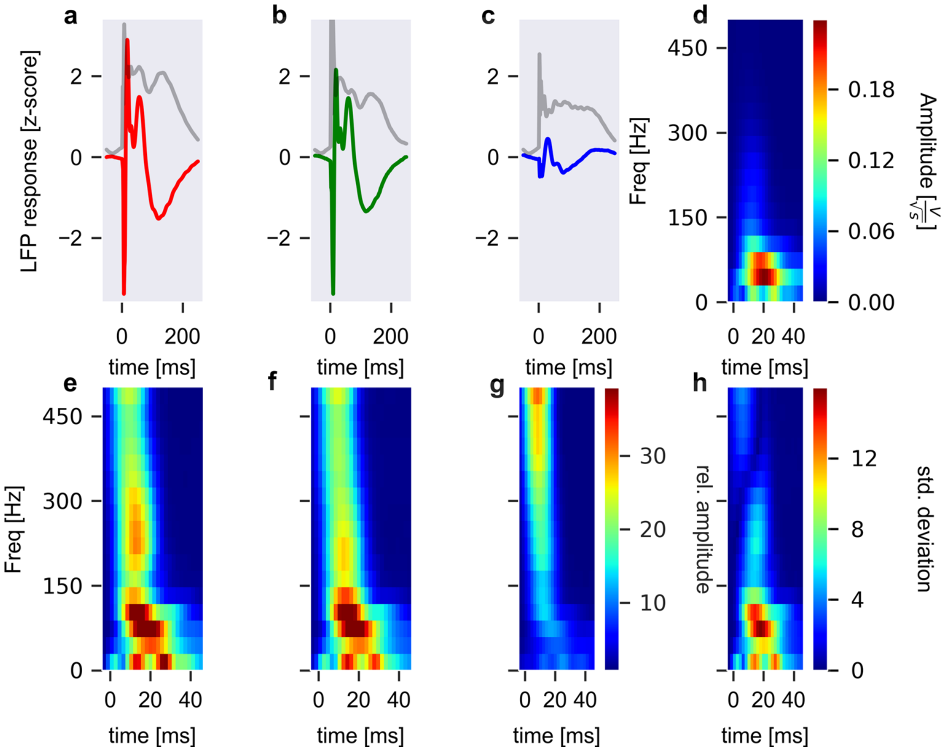


**Fig. 4** Area-specific time–frequency components of LFP responses to CTC stimulation. **a**–**c** Mean LFP response for cortical areas M1 (red), PM (green) and SS (blue) averaged across trials and recording sites. Gray curves indicate the standard deviation across trial-averaged responses of different recordings sites within the respective area. **d** Average spectrogram of LFP responses in M1. Color code represents absolute magnitude. **e**–**g** Spectrograms for M1, PM and SS normalized to the time-average amplitudes during a 100-ms prestimulus time interval. **h** Standard deviation across spectrograms **e**–**g** indicates the time–frequency range with dominant response differences in the beta and gamma range (color figure online)

### 3.3 Multiclass decoding performances for different feature sets

We next compared the pairwise decoding performances when using different sets of features extracted from the MUA and LFP signals or combinations of features from both signal types. To quantify the classification result, here we used the Matthews correlation coefficient (MCC), a single quantity that summarizes the entries in the confusion matrix and corrects for cross-class differences in sample size (see Sect. 2). A MCC of 0 corresponds to a classification at chance level. Pairwise decoding (Fig 6a) was best when using the raw MUA together with the time–frequency analysis of the LFP. The synergistic gain from this combination of MUA and LFP features was high in the most difficult case of classifying M1 versus PM and achieved a performance of MCC = 0.72, which was close to the performance on the simpler classification tasks of M1 versus SS (MCC = 0.73) and PM versus SS (MCC = 0.78). This performance gain from combining MUA and LFP features indicates that at least in the motor cortices, LFP and MUA signals from the same site carry partly independent information. To separate SS from M1 or PM, the time–frequency features of the LFP alone were as good as in combination with MUA features.

To further quantify the area-specific response differences, we trained RF classifiers on all three classes simultaneously (M1, PM, and SS). Here again, a classification based on LFP features yielded better performances than classifications based on MUA features. Overall, the combination of features extracted from the raw MUA and the time–frequency analysis of the LFP yielded the best performances and may thus be considered a robust choice.

### 3.4 Cortical topography of classification performance

We tested whether the ability of the classifier to correctly identify the cortical allocation of single sites was dependent on the single-site spatial location. To this end, we used leave-one-out cross-validation. This procedure allowed us to obtain a class prediction (i.e., a prediction of the cortical area) for each single recording site. We first compared the actual cortical location of single sites with their predicted allocation (color coded) based on the classification for all monkeys separately as shown in Fig. 7. There was generally good

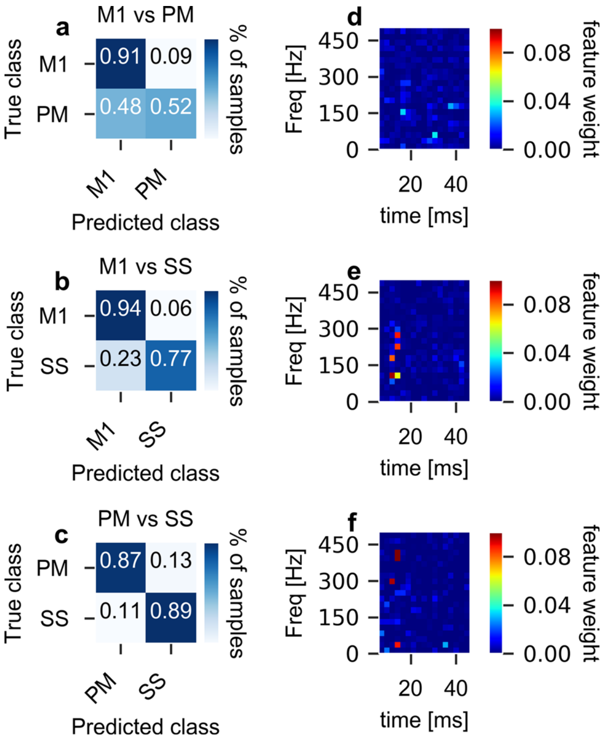


**Fig. 5** Pairwise decoding of cortical area from the LFP spectrogram. **a**–**c** Confusion matrices for pairwise classification. Classification of M1 versus PM (**a**) showed only moderate success with about 50% of recording sites in PM misclassified as belonging to M1 and a false-positive rate of 57%. Classification of M1 versus SS (**b**) and PM versus SS (**c**) shows good performance with correct classification rates in the range of 77–94%. **d**–**f** Each of the matrices reflects the feature importance (color coded) of the respective time–frequency bin for the pairwise classification of the respective cortical areas based on tenfold cross-validated decision trees (color figure online)

agreement between the actual and predicted locations of single cortical sites. Nonetheless, the match was not perfect and points of disagreement were evenly distributed within different areas. This means that the misallocated sites were not confined to the transition zones between adjacent sites. We further explored the decoding performance by plotting the confidence of predicting the correct area for this site at each single recording site (Fig. 8). Here we found generally high confidence within the primary motor cortex but somewhat lower levels of confidence for sites located in the premotor and somatosensory areas. However, there were many exceptions to this general trend that was broadly distributed across all three areas, suggesting there was no core cortical area in which responses to SCP stimulation were easily classified.

## 4 Discussion

Many studies of the cerebellar-thalamo-cortical (CTC) system have provided essential information regarding the anatomical organization of the system (Asanuma et al. 1983b; Holsapple et al. 1991; Sakai et al. 1996; Shinoda et al. 1993, 1982). Several studies have further addressed the impact of the cerebellar output on the motor cortex and showed that inactivation of the deep cerebellar nuclei was followed by a decrease in the phasic discharge of motor cortical neurons during task performance as well as impairments in motor performance (Hore and Flament 1988; Meyer-Lohmann et al. 1975). It was thus suggested that the cerebellar control of motor output is mediated by shaping the response pattern of motor cortical activity.

Here we aimed to systematically study the unique properties of cerebellar impact on motor cortical activity by measuring neural responses to SCP stimulation in identified sites located in the primary motor, premotor and somatosensory cortices. This approach drew on previous studies that have documented the functional connectivity between remote brain sites in behaving primates (Cheney and Fetz 1985; Yanai et al. 2007; Zinger et al. 2013).

We applied objective classification methods on the obtained evoked signals (LFP and MUA) to identify the most informative features of the evoked response which help distinguish between responses obtained from different areas. We found that optimal classification was obtained when using either the time–frequency content of the LFP signal or the response pattern of the MUA response. However, the two signals had practically no additive predictive power, indicating that they were strongly correlated. We further found that using the response amplitude alone yielded poor results when attempting to distinguish between different recording sites. Finally, we found that although the classifier performed well in distinguishing between motor cortical and somatosensory sites, it performed poorly in distinguishing between the primary motor and premotor sites, suggesting that the two areas are under cerebellar control of comparable efficacy. The limited success of M1-to-PM classification may also reflect variability between monkeys which was averaged out when constructing a classifier that was based on the entire dataset. Some of this cross-subject variability could be explained by the mismatch between the actual cortical map and the map drawn for each monkey based on sulci location. These differences varied across monkeys and may thus have contributed to the observed misclassification. Nonetheless, it is also possible that part of the variability was related to differences in proficiency in task performance and/or specific motor strategies employed by each monkey, since previous studies have shown that motor skills can affect the formation of motor maps (see Nudo 2013 for a review). Irrespective of the origin of this inter-subject variability, it may deteriorate the overall performance of the classifier. Future larger datasets for single monkeys could directly confirm or reject this scenario.

Single-site responses evoked by SCP stimulation in M1 and PM were by and large indistinguishable when using clas-

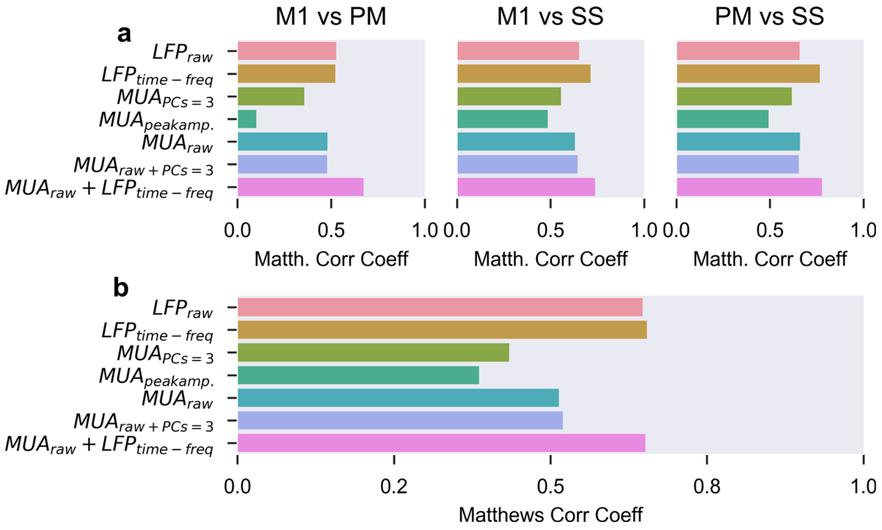


**Fig. 6** Classification performance for different sets of features. **a** Performance of pairwise classification quantified by the MCC for 6 different feature sets that were extracted from the MUA or LFP responses alone and for 1 set (lowest bars) that combined features from both signal types. Combining MUA and LFP features considerably improved classification of M1 versus PM sites reaching a MCC of 0.72. Pairwise classification of motor areas (M1, PM) versus somatosensory areas (SS) showed good performance. Even the response amplitude alone had predictive power. The best performance was achieved when combining MUA and LFP features. **b** MCC for classification into all three cortical areas. Feature sets extracted from the LFP alone resulted in considerably better performance than for feature sets extracted from MUA alone. Combining features from MUA and LFP (lowest bar) led to performance comparable to that of the two LFP feature sets. MCC = 0 represents the expectation value. LFP$_{\mathrm{raw}}$ uses the amplitude values of the $z$-scored LFP response as features; LFP$_{\mathrm{time--freq}}$ corresponds to time–frequency features based on the spectrogram bins as in Figs. 3 and 4; MUA$_{\mathrm{PCs=3}}$ uses the amplitude values projected onto the 3 most informative principal components; MUA$_{\mathrm{peakamp}}$. Only uses the peak amplitude of each MUA response as features; MUA$_{\mathrm{raw}}$ corresponds to the amplitude values of the $z$-scored MUA response; MUA$_{\mathrm{raw+PCs=3}}$ combines the respective feature sets; MUA$_{\mathrm{raw}}$ + LFP$_{\mathrm{time--freq}}$ combines the respective feature sets from both signal types

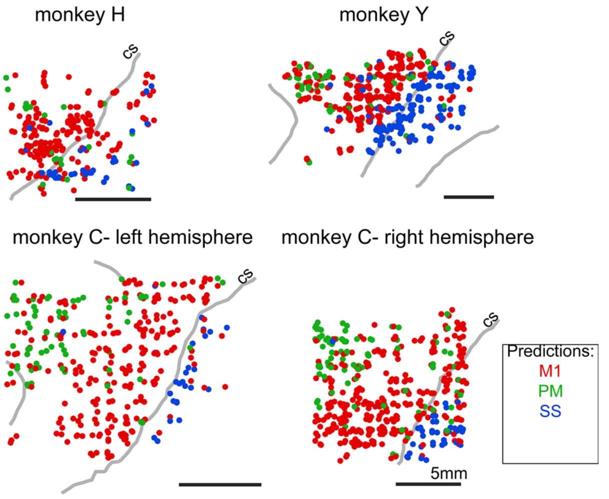


**Fig. 7** Topography of single-site class predictions mapped onto the cortical surface. Shown are the predicted recording site in each monkey and for each single recording site as indicated by color (red = M1, green = PM, blue = SS). Topography of predicted cortical area for all sites is largely consistent with the topography of the true cortical positions. CS: central sulcus; left of CS is anterior (motor areas), right of CS is posterior (somatosensory areas). Classification was based on the time–frequency features of the LFP and used the leave-one-out cross-validation procedure (color figure online)

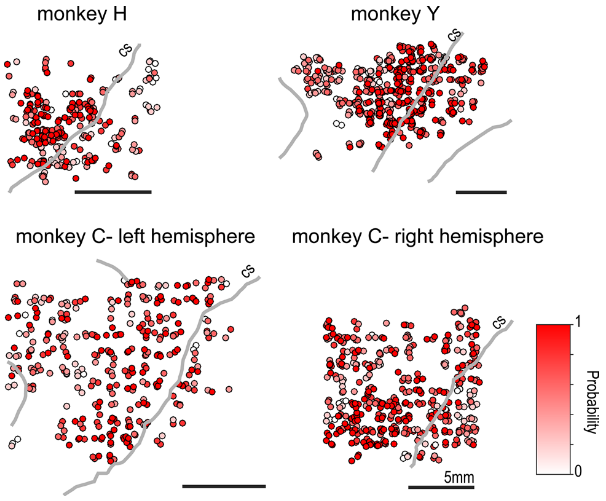


**Fig. 8** Cortical mapping of true class predictions. The color code indicates the probability of predicting the correct class for each single response site (i.e., cortical area M1, PM or SS). For example, a white circle indicates that for this position the classifier predicted the corresponding true cortical area with very low probability, whereas a red circle indicates that for this site the classifier could predict the correct cortical area with high probability. Same classification procedure as in Fig. 7 (color figure online)

sification based on features from MUA or LFP responses alone. Only the combination of both signal types allowed for a good pairwise classification. The spatial distribution of the single-site class prediction across the cortical map indicated that although M1 sites were often correctly classified, the probability of correctly classifying PM sites varied considerably. This result suggests that cerebellar activation exerts comparable impact on these two motor cortical areas, which is somewhat at odds with previous claims that M1 is the main target of the CTC system (Matelli et al. 1989).

Two caveats should be noted here. First, SCP stimulation provides non-selective activation of a large fraction of the pathway. However, during normal activation it is possible that different sources contributing to the SCP are activated in a unique manner such that the final impact of CTC activation on M1 and PM during self-initiated motor actions is in fact different.

Some evidence for differences between impact of the CTC on M1 versus PM could be observed in the decomposition of site-specific average MUA responses to the 3 principal components. We found that for M1, the dominant feature was fast, sharp onset. In contrast, for PM the early component was canceled out so that the evoked response was temporally shifted as compared to the M1 response. This temporal order of motor cortical activation is consistent with the anatomy of thalamocortical fibers which were found to create patches of terminations that span large distances in the rostrocaudal axis of the motor cortex (Shinoda et al. 1993). It is thus possible that the temporal gradient, which exists in the response of M1 versus PM sites, is in fact too fine to be well captured using our shape-based classifier.

Many studies have shown that M1 and PM are involved in different aspects of motor control (cf. Sect. 1). The fact that the CTC system appears to robustly recruit neurons in these two areas may suggest that the system affects both motor preparation and motor execution. The exact features of motor control the CTC system are involved in require further research.

Even though the effect of cerebellar stimulation on areas of the somatosensory cortex was generally small and spatially confined, we did find some somatosensory sites that responded strongly to SCP stimulation. Previous anatomical studies have shown connectivity between the cerebellar receiving motor thalamus and somatosensory area 3a (Padberg et al. 2009). In parallel, a high degree of coupling was reported between cerebellar activity and sensory processes (Gao et al. 1996; Proville et al. 2014). Sensory processing is considered to play an important role in motor control, by relaying peripheral proprioceptive information to motor centers (Burchfiel and Duffy 1972; Heath et al. 1976; Mountcastle and Powell 1959) and possibly by active involvement in movement control (Fromm and Evarts 1982; Matyas et al. 2010; Nelson 1987; Petreanu et al. 2012). Consistent with this notion, subjects with sensory deficits display impaired control of multijoint movements (Ghez and Sainbosrg 1995; Sainburg et al. 1993), similar to findings in cerebellar ataxia patients (Bastian et al. 1996; Bo et al. 2008; Martin et al. 2000). Our findings show some degree of connectivity between the CTC system and somatosensory cortex, which may partially mediate motor integration and/or the modulation of sensory processing related to proprioception.

To summarize, we found that the CTC system can affect different aspects of movement via parallel channels that terminate broadly on the motor cortex and to some extent on the somatosensory cortex as well. Our classifier was able to extract the most informative aspects of the evoked response from both the LFP and MUA signals. Previous studies of long-range pathways have mainly used staining and labeling methods to document the macroscopic anatomy of the system (Asanuma et al. 1983a, b; Holsapple et al. 1991; Orioli and Strick 1989; Schell and Strick 1984; Shinoda et al. 1993). Our methods of identifying (Ruach et al. 2015) and classifying stimulus-triggered evoked responses thus provide additional information about the dynamics of the evoked response and the impact of the pathway on cortical activity *in vivo*.

**Acknowledgements** This research was funded by the German Israeli Foundation (GIF, Grant I-1224-396.13/2012 to MN and YP) within the project *Multiple Time Scales of Signals and Noise in the Motor Hierarchy*. Additional funding was received from the Israel Science Foundation (ISF-1787/13), the generous support of the Baruch Foundation (YP), and the German Science Foundation under the *Institutional Strategy of the University of Cologne* within the *German Excellence Initiative* (DFG-ZUK 81/1) (MN).

## References

Asanuma C, Thach W, Jones E (1983a) Anatomical evidence for segregated focal groupings of efferent cells and their terminal ramifications in the cerebellothalamic pathway of the monkey. Brain Res Rev 286:267–297

Asanuma C, Thach WT, Jones EG (1983b) Distribution of cerebellar terminations and their relation to other afferent terminations in the ventral lateral thalamic region of the monkey. Brain Res Rev 286:237–265

Aumann T, Rawson J, Finkelstein D, Horne M (1994) Projections from the lateral and interposed cerebellar nuclei to the thalamus of the rat: a light and electron microscopic study using single and double anterograde labelling. J Comp Neurol 349:165–181

Bastian AJ, Martin TA, Keating JG, Thach WT (1996) Cerebellar ataxia: abnormal control of interaction torques across multiple joints. J Neurophysiol 76:18

Bo J, Block HJ, Clark JE, Bastian AJ (2008) A cerebellar deficit in sensorimotor prediction explains movement timing variability. J Neurophysiol 100:2825–2832. https://doi.org/10.1152/jn.90221.2008

Breiman L (2001) Random forests. Mach Learn 45:5–32. https://doi.org/10.1023/A:1010933404324

Breiman L, Friedman J, Olshen R, Stone C (1984) Classification and regression trees. Wadsworth Statistics/Probability, Belmont

Burchfiel JL, Duffy FH (1972) Muscle afferent input to single cells in primates somatosensory cortex. Brain Res 45:241–246

Cheney PD, Fetz EE (1985) Comparable patterns of muscle facilitation evoked by individual corticomotoneuronal (CM) cells and by single intracortical microstimuli in primates: evidence for functional groups of CM cells. J Neurophysiol 53:786–804

Darian-Smith C, Darian-Smith I, Burman K, Ratcliffe N (1993) Ipsilateral cortical projections to areas 3a, 3b, and 4 in the macaque monkey. J Comp Neurol 335:14

Fromm C, Evarts EV (1982) Pyramidal tract neurons in somatosensory cortex: central and peripheral inputs during voluntary movement. Brain Res 238:6

Gao J-H, Parsons LM, Bower JM, Xiong J, Li J, Fox PT (1996) Cerebellum implicated in sensory acquisition and discrimination rather than motor control. Science 272:545–547

Ghez C, Sainbosrg R (1995) Proprioceptive control of interjoint coordination. Can J Physiol Phamacol 73:273–284

Godschalk M, Lemon R, Kuypers H, Van der Steen J (1985) The involvement of monkey premotor cortex neurones in preparation of visually cued arm movements. Behav Brain Res 18:143–167

Gorodkin J (2004) Comparing two K-category assignments by a K-category correlation coefficient. Comput Biol Chem 28:367–374. https://doi.org/10.1016/j.compbiolchem.2004.09.006

Heath CJ, Hore J, Phillips CG (1976) Inputs from low-threshold muscle and cutaneous afferents of hand and forearm to areas 3a and 3b of baboon's cerebral cortex. J Physiol Lon 257:199–227

Holmes G (1939) The cerebellum of man. Brain J Neurol 62:1–30

Holsapple J, Preston J, Strick P (1991) The origin of thalamic inputs to the hand representation in the primary motor cortex. J Neurosci 11:2644–2654

Hore J, Flament D (1988) Changes in motor cortex neural discharge associated with the development of cerebellar limb ataxia. J Neurophysiol 60:1285–1302

Hore J, Preston JB, Cheney PD (1976) Responses of cortical neurons (areas 3a and 4) to ramp stretch of hindlimb muscles in the baboon. J Neurophysiol 39:17

Huffman KJ, Krubitzer L (2001) Area 3a: topographic organization and cortical connections in marmoset monkeys. Cereb Cortex 11:19

Ivanusic JJ, Bourke DW, Xu ZM, Butler EG, Horne MK (2005) Cerebellar thalamic activity in the macaque monkey encodes the duration but not the force or velocity of wrist movement. Brain Res 1041:181–197. https://doi.org/10.1016/j.brainres.2005.02.005

Ivry R, Keele S (1989) Timing functions of the cerebellum. J Cogn Neurosci 1:136–152

Jones EG, Coulter JD, Hendry SHC (1978) Intracortical connectivity of architectonic fields in the somatic sensory, motor and parietal cortex of monkeys. J Comp Neurol 181:57

Martin JH, Cooper SE, Hacking A, Ghez C (2000) Differential effects of deep cerebellar nuclei inactivation on reaching and adaptive control. J Neurophysiol 83:1886–1899

Matelli M, Luppino G, Fogassi L, Rizzolatti G (1989) Thalamic input to inferior area 6 and area 4 in the macaque monkey. J Comp Neurol 280:468–488. https://doi.org/10.1002/cne.902800311

Matthews BW (1975) Comparison of the predicted and observed secondary structure of T4 phage lysozyme. Biochim Biophys Acta 405:442–451

Matyas F et al (2010) Motor control by sensory cortex. Science 330:1240–1243

Meyer-Lohmann J, Conrad B, Matsunami K, Brooks VB (1975) Effects of dentate cooling on precentral unit activity following torque pulse injections into elbow movements. Brain Res 94:237–251

Moran DW, Schwartz AB (1999) Motor cortical representation of speed and direction during reaching. J Neurosci 82:2676–2692

Morrow MM, Jordan LR, Miller LE (2007) Direct comparison of the task-dependent discharge of M1 in hand space and muscle space. J Neurophysiol 97:1786–1798

Mountcastle VB, Powell TPS (1959) Central nervous mechanisms subserving position sense and kinesthesis. Bull Johns Hopkins Hosp 105:173–200

Nelson RJ (1987) Activity of monkey primary somatosensory cortical neurons changes prior to active movement. Brain Res 406:402–407

Nudo RJ (2013) Recovery after brain injury: mechanisms and principles. Front Hum Neurosci 7:887. https://doi.org/10.3389/fnhum.2013.00887

Orioli PJ, Strick PL (1989) Cerebellar connections with the motor cortex and the arcuate premotor area—an analysis employing retrograde transneuronal transport of WGA-HRP. J Comp Neurol 288:612–626

Padberg J, Cerkevich C, Engle J, Rajan AT, Recanzone G, Kaas J, Krubitzer L (2009) Thalamocortical connections of parietal somatosensory cortical fields in macaque monkeys are highly divergent and convergent. Cereb Cortex 19:27

Padberg J et al (2007) Parallel evolution of cortical areas involved in skilled hand use. J Neurosci 27:10

Petreanu L et al (2012) Activity in motor-sensory projections reveals distributed coding in somatosensation. Nature 489:299–303. https://doi.org/10.1038/nature11321

Phillips C, Powell T, Wiesandanger M (1971) Projections from low threshold muscle afferents of hand and forearm to area 3a of baboon's cortex. J Physiol 217:419–446

Proville RD et al (2014) Cerebellum involvement in cortical sensorimotor circuits for the control of voluntary movements. Nat Neurosci 17:1233–1239. https://doi.org/10.1038/nn.3773

Rickert J, Riehle A, Aertsen A, Rotter S, Nawrot MP (2009) Dynamic encoding of movement direction in motor cortical neurons. J Neurosci 29:13870–13882. https://doi.org/10.1523/JNEUROSCI.5441-08.2009

Ruach R, Mitelman R, Sherman E, Cohen O, Prut Y (2015) An assumption-free quantification of neural responses to electrical stimulations. J Neurosci Methods. https://doi.org/10.1016/j.jneumeth.2015.07.005

Sainburg RL, Poizner H, Ghez C (1993) Loss of proprioception produces deficits in interjoint coordination. J Neurophysiol 70:12

Sakai ST, Insae M, Tanji J (1996) Comparison of cerebellothalamic and pallidothalamic projections in the monkey *(Macaca fuscata)*—a double anterograde labeling study. J Comp Neurol 368:215–228

Schell G, Strick P (1984) The origin of thalamic inputs to the arcuate premotor and supplementary motor areas. J Neurosci 4:539–560

Scott SH, Gribble PL, Graham KM, Cabel DW (2001) Dissociation between hand motion and population vectors from neural activity in motor cortex. Nature 413:161–165

Shalit U, Zinger N, Joshua M, Prut Y (2012) Descending systems translate transient cortical commands into a sustained muscle activation signal. Cereb Cortex 22:1904–1914. https://doi.org/10.1093/cercor/bhr267

Shinoda Y, Kakei S, Futami T, Wannier T (1993) Thalamocortical organization in the cerebello-thalamo-cortical system. Cereb Cortex 3:421–429

Shinoda Y, Yamazaki M, Futami T (1982) Convergent inputs from the dentate and the interpositus nuclei to pyramidal tract neurons in the motor cortex. Neurosci Lett 34:111–115

Spencer RM, Zelaznik HN, Diedrichsen J, Ivry RB (2003) Disrupted timing of discontinuous but not continuous movements by cerebellar lesions. Science 300:1437–1439. https://doi.org/10.1126/science.1083661

Strick PL (1986) The organization of thalamic inputs to the "premotor" areas. Prog Brain Res 64:99–109. https://doi.org/10.1016/s0079-6123(08)63405-6

Wang X, Zhang M, Cohen IS, Goldberg ME (2007) The proprioceptive representation of eye position in monkey primary somatosensory cortex. Nat Neurosci 10:7

Weinrich M, Wise SP (1982) The premotor cortex of the monkey. J Neurosci 2:1329–1345

Yanai Y, Adamit N, Harel R, Israel Z, Prut Y (2007) Connected corticospinal sites show enhanced tuning similarity at the onset of voluntary action. J Neurosci 27:12349–12357. https://doi.org/10.1523/JNEUROSCI.3127-07.2007

Yanai Y, Adamit N, Israel Z, Harel R, Prut Y (2008) Coordinate transformation is first completed downstream of primary motor cortex. J Neurosci 28:1728–1732. https://doi.org/10.1523/JNEUROSCI.4662-07.2008

Zinger N, Harel R, Gabler S, Israel Z, Prut Y (2013) Functional organization of information flow in the corticospinal pathway. J Neurosci 33:1190–1197

# Chapter 3

# Conditional Probability Echo-State Neural network (CPESN)

## 3.1 Introduction

Electroencephalography (EEG) is a popular method to record voltage fluctuations on the surface of the brain. Although it is often used to assess brain dysfunction like epileptic seizures in clinical contexts, the origins of the EEG signal are still poorly understood, and only very few generative models (Wulsin et al., 2011) exist. Here we introduce Conditional Probability Echo-State Networks (CPESN) as a new means of modeling EEG, or similar neurophysiological data. We train the CPESN on EEG time series recorded from epileptic dogs [1]. We demonstrate that the trained CPESN outperforms standard time series models, such as the autoregressive process or multilayer perceptron regression in generating signals that match the power spectrum and other statistics of the recorded EEG. By this method, which is adapted and extended from particle physics (Feindt, 2008), a neural network is trained to represent the conditional probability density function of the future values of a stochastic process, given the population activity of a Echo-State Network (Jaeger, 2001) driven by the value of the process' previous time step and tuned to the *edge of chaos* (Toyoizumi and Abbott, 2011). Once the CPESN is trained, samples drawn from the trained model have very similar statistics to samples of the original process. Moreover, the CPESN can be used as a predictive model if generated samples are presented as inputs to the CPESN iteratively (closed-loop). Such a model may then be used for forecasting and generation of time-series, in order to predict brain dysfunctions, like epileptic seizures or other neurological events. In contrast to common machine learning approaches which learn to predict the most likely future value from presented samples, the CPESN provides an estimate of the conditional probability density function (or posterior distribution), and thus has an intrinsic representation of the process' stochasticity. This aspect might be crucial for modeling highly variable neurophysiological time series such as EEG.

In the following the architecture and methodology of the CPESN is described followed by results of a first test-drive. While this is ongoing work, the progress made so far is significant and the results are sound to be considered as proof of concept of the general idea and thus are included into this thesis. As such, some results might be partly inconclusive and certainly require more rigorous investigation and assessment.

[1] American Epilepsy Society Seizure Prediction Challenge, `https://www.kaggle.com/c/seizure-prediction/data`

## 3.2 Method

In this section, first the general architecture of the CPESN is introduced. Afterwards the mathematical background is develop for preprocessing of the target variable $t$ which will later be used as training target by a neural network. To this end, a continuous random variable $t \in \mathbb{R}$ with probability density function $f(t)$ is considered. A non-linear transformation is introduced which transforms $t$ into a new, uniformly distributed random variable $s$. It is shown, that this transformation is the cumulative distribution function $F(t)$ and it is described how it can be estimated from data. Afterwards, it is shown how this transformed variable $s$ can be used to fit a logistic regression model using a neural network, suitable to predict the *complete* transformed probability density function $g(s|x)$ conditioned on some input data $x$. Finally it is described how to obtain $f(t|x)$, the complete conditional distribution density function of the original random variable $t$ conditioned on observations $x$.

### 3.2.1 CPESN architecture

The overall architecture of the CPESN model is depicted in Fig. 3.1. The problem of time series forecasting can be considered as modelling the conditional probability of the signal value at time step $t$ conditioned on some autoregressive history $x$. The CPESN comprises two components, a random recurrent neural network (RNN) to generate temporal features and a feed-forward artificial neural network (ANN) to estimate conditional probability density functions. For the CPESN, the autoregressive history $x$ is encoded in the memory of a random recurrent neural network which has been tuned to the edge of chaos (Toyoizumi and Abbott, 2011), similarly to an RNN used in Echo-State networks (Jaeger, 2001) (Fig. 3.1B). The idea is that the rich and partially random dynamics of this RNN captures the autoregressive history on multiple time scales and thus no specific order of the autoregressive history needs to be chosen. The subsequently connected feed-forward ANN (Fig. 3.1C) is trained to represent the full conditional probability distribution $f(t|x)$. This can be achieved through logistic regression of the ANN and fitting a smooth b-spline through the ANNs output (Fig. 3.1D).

### 3.2.2 Transformation of $t$ into a uniformly distributed random variable $s$

A non-linear transformation $F : t \to s,\ s \in [0, 1]$ of a continues random variable $t$ is introduced, which transforms $t$ into a new random variable $s$, such that $s$ is uniformly distributed in $[0, 1]$ with density function $g(s) = 1$.

**Theorem 1.** *The transformed random variable*

$$s = F(t) = \int_{-\infty}^{t} f(t')\,\mathrm{d}t' \tag{3.1}$$

*is uniformly distributed in* $[0, 1]$ *and its density is* $g(s) = 1$.

*Proof.* The function $F$ is the cumulative distribution function (CDF) of the random variable $t$, which is a strictly increasing function with the following properties:

$$\lim_{t\to-\infty} F(t) = 0 \tag{3.2}$$

$$\lim_{t\to\infty} F(t) = 1 \tag{3.3}$$

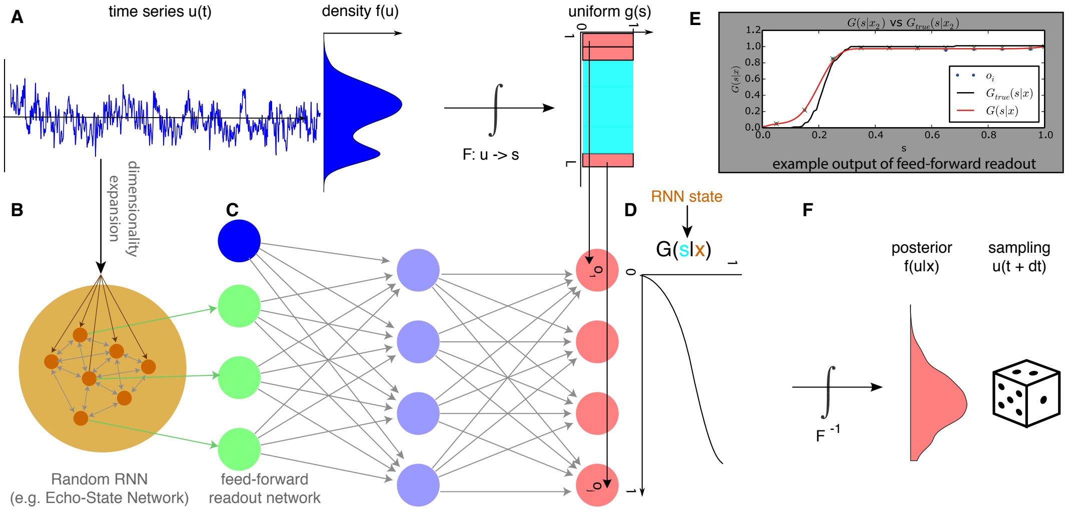


FIGURE 3.1: **A:** Stochastic time series $u(t)$ is used as input to the CPESN and its marginal density over signal values $f(u)$ is estimated. The marginal density can be transformed into a uniform density $g(s)$ by the function $F$. Prediction of time-series can be modelled as the conditional probability $f(u(t)|x)$, where the conditioning variable $x$ is an autoregressive history of $u(t)$. **B+C:** Architecture of the CPESN. At each time step, the signal value $u(t)$ is fed into a random RNN to generate temporal features $x$ of population activity on multiple time scales. A multi-layer artificial neural network is used and trained to estimate the conditional probability density function $f(u|x)$. This is achieved by logistic regression on equidistant points of the transformed uniform density $g(s)$. **D:** The outputs $o_j$ of the neural network can be assembled to a continuous cumulative distribution function $G(s|x)$ by performing a smooth b-spline fit. **E:** Real example of a smooth b-spline fit over the network output units $o_j$. **F:** Using the inverse transform $F^{-1}$ of $F$ the full conditional probability density $f(u|x)$ in the original signal domain can be recovered and sampled from.

With the definition in 3.1 and equations 3.2,3.3 it follows that $s \in [0,1]$. Since $F(t)$ is a strictly increasing function, it is invertible by $F^{-1}(s)$ which again is strictly increasing. As the transformed variable $s$ is a random variable, it can be described in terms of its cumulative distribution function:

$$G(s) = \int_0^s g(s')\, \mathrm{d}s' \tag{3.4}$$

Using equation 3.4 and properties from increasing functions of random variables and their distribution, this can be rewritten as

$$G(s) = F(F^{-1}(s)) = s$$

and one obtains the density $g(s)$ as the derivative of $G(s)$ with respect to $s$:

$$\begin{aligned} g(s) &= \frac{\mathrm{d}}{\mathrm{d}s} G(s) \\ &= \frac{\mathrm{d}}{\mathrm{d}s} s \\ &= 1 \end{aligned}$$

□

### 3.2.3 Computing transformation $F : t \to s$

Since $F(t)$ is the cumulative distribution function of $t$, it can easily be estimated from a set of $m$ samples $t_i$, $i \in [0, m]$ as:

$$s = F(\hat{t}) = P(t \leq \hat{t}) = \frac{\mid \{t_i | t_i \leq \hat{t}\} \mid}{m}$$

where $P(t \leq \hat{t})$ is the probability, that the random variable $t$ takes on a value less than or equal to $\hat{t}$.

Algorithmically, this can be implemented as follows:

- sort $t_i$ in incremental order
- for each $t_i$: find its index $k$ in the sorted list (e.g. binary search). $k$ is equivalent to the number of samples less than $t_i$.
- compute $F(t_i) = \frac{k}{m}$
- perform a linear interpolation on the tuples $(t_i, F(t_i))$, $i \in [0, m]$

For the inverse transformation $F^{-1}(s)$, the tuples are swapped $(F(t_i), t_i)$, $i \in [0, m]$ and a separate linear interpolation is performed.

### 3.2.4 Logistic Regression

The objective is to fit a logistic regression model, in this case a neural network, which can predict the complete probability density function $f(t|x)$ of the random variable $t$ conditioned on observations $x$. To do this, an appropriate binary target vector is designed and a neural network is trained for logistic regression. As shown in 3.2.2, $t$ can be transformed into a uniformly distributed random variable $s$ with density $g(s) = 1$. To prepare target vectors to train the model for logistic regression, the CDF $G(s)$ is sampled at $N$ equidistant levels

$$L_j = \frac{j - \frac{1}{2}}{N}, j \in [1, N] \tag{3.5}$$

which yields $N + 1$ intervals. For a given realization $s_{true}$, the binary target vector $\vec{T}$ for logistic regression is defined as:

$$T_i = \begin{cases} -1 & \text{if } g(s_{true}) < L_j \\ 1 & \text{if } g(s_{true}) > L_j \end{cases} \tag{3.6}$$

Since a neural network is used for model fitting, the target vectors use $(-1, 1)$ encoding scheme instead of $(0, 1)$ as opposed to in standard logistic regression. This is due to the bipolar non-linearities used by the network units like *tanh* which range from $[-1, 1]$. Using such bipolar non-linearities is more beneficial for learning in neural networks since it allows neurons to act inhibitory and excitatory. For logistic regression the network outputs $o_j$ have to be shifted and rescaled $\hat{o}_j = \frac{1-o_j}{2}$ to the interval $[0, 1]$ when using bipolar non-linearities. Alternatively one can choose a different non-linearity in the output layer like the softmax function. The estimates $\hat{o}_j$ of the logistic regression model contain for each $L_j$, the probability whether $s$ is smaller or larger than $L_j$. Since the levels $L_j$ are increasing, the $\hat{o}_j$ can be considered as discrete samples of the cumulative distribution function $G(s|x)$ and performing an appropriate curve fitting yields an estimate of the complete CDF.

### 3.2.5 Constrained, smooth curve fitting of $G(s|x)$

In order for the shifted and rescaled neural network outputs $\hat{o}_j$ to be interpreted as the cumulative distribution function $G(s|x)$ conditioned on input $x$, curve fitting has to be performed. However, since the goal is to obtain a smooth density $g(s|x) = \frac{\mathrm{d}}{\mathrm{ds}} G(s|x)$ the fitted curve needs to satisfy important constraints. First, in order to obtain a smooth density $g(s|x)$ as the derivative of the fitted curve, $G(s|x)$ should be continuous and as smooth as possible. Secondly, the derivative should be able to be computed analytically and the fits should be fast. Cubic ($k = 3$) B-Spline curves are smooth and $k - 1$ times continuous differentiable. Furthermore, the derivative of a B-Spline itself is a B-Spline of lower degree and can be computed analytically from it's B-Spline coefficients. In order to yield a valid distribution function, in addition to smoothness some extra constraints are imposed on the B-spline fit, positivity and monotonicity:

**Positivity:** $G(0) = 0$ and $G(1) = 1$

**Monotonicity:** $\frac{\mathrm{d}}{\mathrm{ds}} G(s|x) \geqq 0$

Fitting smooth B-Spline curves is challenging and quickly gets complex since there's always a trade-off between *smoothness-of-fit* and *closeness-of-fit*. In addition to that the choice of optimal parameters strongly depends on the problem domain at hand (e.g. CAD, computer animation, engineering etc) and desired properties of the curve. This can be seen by the vast amount of specialized algorithms for optimal knot selection, positioning and curve fitting. Since B-Spline fitting is not the main focus of this work, a general purpose method for smooth B-Spline fitting is used with automatic knot determination and positioning as described by Dierckx, 1995. Combined with the additional constraints, this method satifies all of the required properties for a smooth $G(s|x)$ fit. A fast, highly optimized implementation of this algorithm is readily available as part of the FITPACK library which is widely available e.g. through the Python library Scipy and its `UnivariateSpline` class.
The constrained curve fitting for $G(s|x)$ through the network outputs $\hat{o}_j$ is performed as follows:

1. First, a regular smooth B-Spline fit is performed to obtain an initial set of knots $\lambda_i$ and coefficients $c_i$. Accepting a squared residual error in the order of $1e^{-6}$ provides a close fit to the network outputs while still maintaining sufficient smoothness of the curve.

2. To satisfy the remaining constraints, a constrained optimization over the initial coefficients $c_i$ is performed where the constraints are:

$$\begin{aligned} G(0) &= 1 \\ G(1) &= 1 \\ \frac{\mathrm{d}}{\mathrm{ds}} G(s|x) &\geqq 0 \end{aligned}$$

If the network is trained well enough, this constraint optimization is fast and requires only very few iterations by an appropriate optimizer (around 10 iterations with LBFGS).

Afterwards, the density $g(s|x)$ can be obtained analytically from the smooth B-spline fit as the first derivative of $\frac{\mathrm{d}}{\mathrm{ds}} G(s|x) = g(s|x)$. To obtain the original $t$ the inverse transformation $F^{-1}(s) = t$ is applied.

### 3.2.6 Computation of probability original density function $f(t|x)$ from $G(s|x)$

The probability density function $f(t|x)$ of the original random variable $t$ can be computed from the density $g(s|x) = \frac{\mathrm{d}}{\mathrm{ds}} G(s|x)$. To show this, the property of inverse function derivatives is used and the theorem of the density of an increasing function. If a function $f:$ x→y is strictly increasing and differentiable, the function $f$ is invertible and the derivative of its inverse $f^{-1}$ is given by:

$$\frac{d}{\mathrm{dy}} f^{-1}(y) = \frac{1}{\frac{\mathrm{d}}{\mathrm{dx}} f(f^{-1}(y))} \tag{3.7}$$

**Theorem 2.** *When $x$ is a continuous random variable with probability density function $f_x$ and $\varphi_x : x \to y$ a strictly increasing, differentiable function of $x$, then also $y$ is a continues random variable and its probability density function is*

$$f_y(y) = \begin{cases} f_x(\varphi_x^{-1}(y)) \frac{d}{dy} \varphi^{-1}(y) & \text{if } y \in \text{support of } \varphi_x \\ 0 & \text{if } y \notin ... \end{cases} \tag{3.8}$$

*Proof.* With equations 3.7 and 3.8 one can obtain the density $f(t|x)$ as follows:

$$\begin{aligned} g(s|x) &= f(F^{-1}(s)|x) \frac{\mathrm{d}}{\mathrm{ds}} F^{-1}(s) \text{ using equation 3.8} \\ &= f(t|x) \frac{\mathrm{d}}{\mathrm{ds}} F^{-1}(s) \\ f(t|x) &= \frac{g(s|x)}{\frac{\mathrm{d}}{\mathrm{ds}} F^{-1}(s)} \\ &= \frac{g(s|x)}{\frac{1}{\frac{\mathrm{d}}{\mathrm{dt}} F(F^{-1}(s))}} \text{ using equation 3.7} \\ &= g(s|x) \frac{\mathrm{d}}{\mathrm{dt}} F(F^{-1}(s)) \\ &= g(s|x) f(t) \end{aligned}$$

This shows, that the probability density function $f(t|x)$ of the original random variable $t$ conditioned on input $x$ can be obtained from the smooth B-spline fit $G(s|x)$ of a neural network's output. □

### 3.2.7 Computation of statistical moments of $f(t|x)$

With the complete conditional probability density function $f(t|x)$ it is possible to compute statistical moments and quantiles of interest.

**Expected value or mean:**

The expected value or mean can be estimated using Monte-Carlo integration:

$$\begin{aligned}\langle t\rangle &= \int t\prime f(t\prime|x)dt\prime \\ &= \int t\prime(s)g(s\prime|x)ds\prime \\ &\approx \sum_j F^{-1}(s_j)g(s_j|x) \\ &\approx \frac{1}{M}\sum_{m=1}^{M} F^{-1}(G^{-1}(\frac{m-0.5}{M}))\end{aligned}$$

Where the latter expression is particularly simple to calculate as only $M$ equidistant points in the interval $[0,1]$ are chosen and the average after applying two inverse transformations is computed. In general the expectation value of any function $a(t)$ of the random variable $t$ can be estimated as follows:

$$\begin{aligned}\langle a(t)\rangle &= \int a(t\prime)f(t\prime|x)dt\prime \\ &\approx \sum_j a(F^{-1}(s_j))g(s_j|x) \\ &\approx \frac{1}{M}\sum_{m=1}^{M} a(F^{-1}(G^{-1}(\frac{m-0.5}{M})))\end{aligned}$$

**Median:**

The median of the conditional distribution can be calculated by:

$$t_{median} = F^{-1}(G^{-1}(0.5))$$

**Left and right error intervals:**

Left and right error intervals can be defined as the limits of the interval that contains 68.26% of the data, similarly to the definition used with Gaussian distributions:

$$\begin{aligned}\sigma_{left} &= F^{-1}(G^{-1}(0.5)) - F^{-1}(G^{-1}(0.8413)) \\ \sigma_{right} &= F^{-1}(G^{-1}(0.1567)) - F^{-1}(G^{-1}(0.5))\end{aligned}$$

## 3.3 Results

The CPESN method is applied to continuous intracranial EEG recordings of dogs with naturally occurring epilepsy (Howbert et al., 2014) and results are summarized in Fig. 3.2. It shows how the method can be used in a generative mode, by feeding back its own predictions as input to evolve in a closed-loop manner. For prediction of the signal's value in the next time step either the mean ($\overline{x}_{CPESN}(t)$) of the predicted probability density is used or a random sample from the density is drawn ($X_{CPESN}(t)$) by inverse sampling from the predicted cumulative distribution function $G(s|x)$. The performance is quantified in terms of similarity of the power spectral density between the model-generated instance and the original EEG time series (Fig. 3.2 B+C).
To act as a detector of rare events like epileptic seizures, for each value at time step $t$ of the original EEG signal the signal value's probability is evaluated using the predicted probability density function of the CPESN model (Fig. 3.2A). This yields a probability trace, where the probability is higher for signal values that follow the regular dynamics (healthy EEG) of the time series the model has been trained on. The probability is low for signal values that do not follow the dynamics of the healthy EEG signal. This can be used to identify anomalous segments of the EEG for example due to a seizure.

## 3.4 Discussion

Although the presented results so far are limited to a single use-case and a single model instance, it can be considered as a successful proof-of-concept of the general approach. Certainly, a more rigorous performance evaluation using cross-validation over several model instances and different data sets needs to be done. Using a random recurrent neural network to expand the dimension of a univariate time-series into a set of random features of different time-scales works but is not optimal. The random RNN is difficult to tune to obtain relevant features and its temporal memory is limited by the number its neurons. Consequently, increasing the number of neurons quickly blows up the computational cost of the overall method. Additionally, the connectivity so far is fixed and no training of the recurrent weights is performed based on the input signal. An alternative could be using a LSTM network (Hochreiter and Schmidhuber, 1997) instead or a feed-forward model using dilated convolutions (Yu and Koltun, 2015; Oord et al., 2016). LSTMs are computationally less expensive, the size of the temporal memory can be learned depending on the time-scale of the input signal and their memory capacity generally exceeds the capacity of RNNs. Dilated convolutions allow the receptive field of the model, in terms of size of the autoregressive history, to grow exponentially as a function of number of neurons in the network. Additionally, both methods would allow the model to be trained end-to-end using backpropagation through time or regular backpropagation.

Another drawback so far is, that the computational overhead of fitting a constrained B-spline at each time-step of the input signal slows down the processing speed of the method. In it's current state it is not applicable for realtime applications, which is a necessary requirement for brain computer interface applications and useful applications within a clinical context. This could be improved, by looking for an analytical, closed-form solution of the constraint B-splines, specific to the context of positive, monotonic and smooth target functions. It would eliminate the additional step of solving a constraint optimization problem for prediction of each time step. Alternatively, it would be interesting to explore if the necessity of B-splines could be eliminated entirely by directly learning the free parameters of the B-splines within the neural network.

Another interesting extension to consider is application of the method to multivariate time-series. For EEG and many other physiological signals, this is the more common case as signals from multiple electrodes or sensors are collected simultaneously. A first experiment towards this direction has been conducted, where for each electrode channel a separate CPESN model has been trained. This can be done very efficiently, as the different model instances can be trained in parallel. Preliminary results indicate, that it is generally working and a viable approach to pursue.

In summary, considering prediction of time-series in the context of estimating conditional probability distributions is an interesting approach and relevant for applications that require more than a single point estimate of a signal's future value.

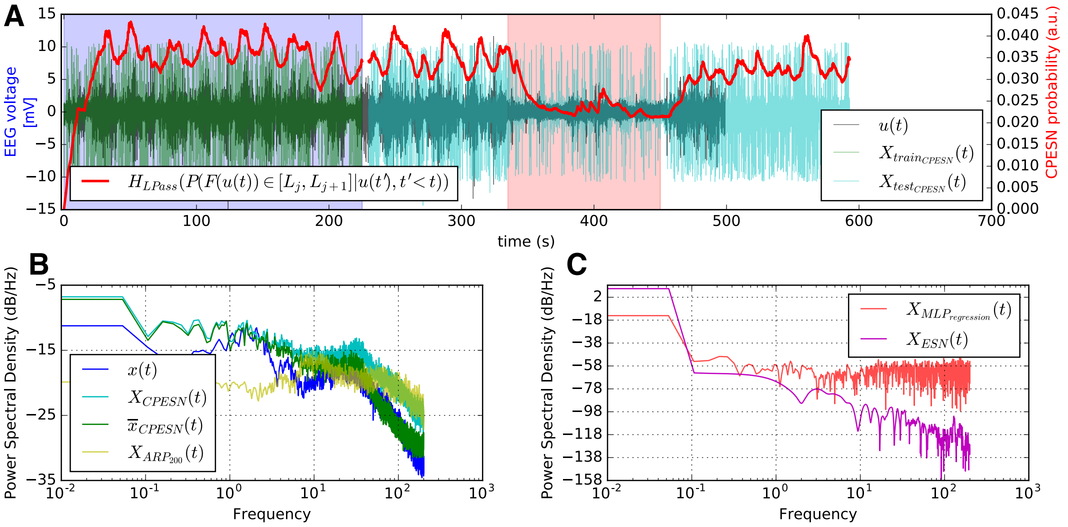


FIGURE 3.2: **A:** Time series $u(t)$ of a single-channel EEG recording (black). The CPESN model is trained on a healthy EEG segment (light blue) between $0 - 220$ seconds ($X_{train_{CPESN}}(t)$). The trained model is then used in a generative mode, to forecast the stochastic signal. This is done by closed-loop simulation (cyan), where a signal value is sampled from the model for time step $t$ and fed back as autoregressive input to predict time step $t + 1$. Overlayed in red is the probability trace (low-pass filtered) as predicted by the model: The conditional probability of $u(t)$, conditioned on the autoregressive history, for each time step. The probability is high, when the signal follows the training distribution and low when not. This is the case during rare events like an epileptic seizure, marked in red between $350 - 450$ seconds. **B+C:** Comparison of the original stochastic process $u(t)$ and instances generated by various models quantified in terms of similarity of the power spectral density. **B:** Closed-loop simulation of the CPESN model using random inverse sampling (cyan) or mean (green) of the predicted conditional probability distribution. For reference a traditional autoregressive model of order 200 (yellow) has been trained. The CPESN is able to generate instances that quantitatively best match the power spectrum of the original stochastic process as compared to the other methods. **C:** Instances generated by an autoregressive multi-layer perceptron model (red) and an Echo-State network model (magenta) are not able to match the power spectrum of the original process (note: different y-axis scales in **B** and **C**)

# Chapter 4

# Discussion

In the following sections higher level concepts are discussed that can contribute towards understanding learning, cognition and behavior in biological and artificial systems. It is considered how methods and ideas of both domains can serve as inspiration to drive progress in each research field. It is discussed how the work of this thesis is embedded within the framework of the discussed broader conceptual ideas and provides suggestions for potential future work along these lines.
For discussions of results specific to the particular research questions of the publications listed under chapter 2 and their implications the reader is kindly referred to the discussion section within the individual paper.

## 4.1 The *Bermuda triangle of intelligence.*

One goal of this thesis is to promote the perspective, that learning, cognition and behavior should be considered simultaneously and studied as an integrative system. This approach has previously been introduced by the analogy of the *Bermuda triangle of intelligence*. The majority of experimental work in neuroscience is conducted and limited to specific sub-systems of the brain, e.g. the retina, visual system, motor systems or memory processing in the hippocampus. For experimental work there are many reasons why this is the correct and most of the time the only way of approaching things, particularly due to technical limitations and controlling experimental variables. However, it can be observed that the same approach is often used in computational neuroscience. Learning rules are developed and applied without considering the different neural codes used for sensory preprocessing or models are built that match statistics of observed activity or some abstract phenomenon of physics. Despite its undeniable existence, often it's not clear how all these abstract methods could be implemented in a real biological system or how it is used to generate real relevant behavior. While this kind of work contributes to a mathematical and statistical understanding of certain phenomenons, it often leads to highly specialized computational models and learning rules, capturing only a very specific aspect. Yet the brain and its sub-systems are highly versatile and one sub-system is able to solve a broad range of different computational, cognitive and behavioral problems. In the end, it often remains elusive how computational models can be used end-to-end to solve real behavioral or cognitive problems. Where *end-to-end* refers to the fact, that a model captures all relevant steps from input to output: sensory input, encoding, processing and generation of (meaningful) output. This integrative approach is captured by the term *Bermuda triangle of intelligence* and is an important gap that needs to be closed towards our understanding of brain function and what makes *intelligence* in general. One possible solution is to build fully functional integrative models.

In Rapp, Nawrot, and Stern, 2020, despite using only a single neuron, the model is able to generalize to solve the true counting problem and not only the sub-problem of counting items in the range of $0-5$. Additionally, the output of the model can be used to solve more advanced cognitive problems like "greater-than" relation without the need to be retrained on the specific "greater-than" task. The second work in Rapp and Nawrot, 2020 goes one step further. It shows, that the same trained model instance can be used to form associative memories and recall them by generating binary behavior responses. Additionally it can be used to segregate pleasant from background odors and to generate appropriate motor commands during foraging. As such, the models of both studies are versatile to some degree and after being trained once able to solve more than the initial problem which it has been specifically trained on. This is achieved by carefully considering the properties of the sensory input, its neural encoding and the choice of a learning rule that is capable to take full advantage of these specific properties. Additionally, the type of learning problem to be solved is carefully chosen. Instead of learning to solve the entire problem at once, the objective is to learn an appropriate sub-problem which is simple enough to be learned quickly and at the same time can be used to solve the overall problem. The success is achieved by the optimal combination of sensory processing, learning rule and behavioral objective to be solved, taking into account all components of the *Bermuda triangle of intelligence*.

## 4.2 Innate behavior, experience based learning and reasoning.

Innate behavior can be considered as strategies that are either pre-programmed by evolution, due to genetic priors, or emerge during development. It is a fixed set of behavioral and cognitive skills that guarantee survival. The neural wiring and computational mechanisms need to be specialized but still be general enough to successfully deal with complex environments. However, in general their ability to generalize are limited. Complementary to this is the ability to acquire new knowledge and experience during lifetime through continual learning which enables an organism to adapt to new environments and generalize to novel conditions. As neuronal resources are limited, this learning process should not take up all resources or overwrite existing innate behavior. Thus, the learning objective should be the best trade-off between the simplest learnable problem, which at the same time is universal enough to contribute to solutions of as many macroscopic problems as possible.

Conceptually this can be related to a *divide-and-conquer* approach, a versatile and powerful generic principle of problem solving widely used in computer science. It works by (recursively) breaking down a complex problem into two or more sub-problems until they become simple enough to be solved directly. The solutions to the sub-problems are then recombined to solve the original problem. Examples from computer science are algorithms used for sorting (quicksort, merge sort), multiplication of large numbers (Karatsuba algorithm) and many other highly efficient algorithms. Dynamically applying the divide-and-conquer principle can be considered as a form of reasoning. By this compute time and computational capacity is traded off against memory capacity.

Reasoning is the opposite of memorization, where one tries to remember and store all possible solutions to encountered problems. Given the vast amount of possible problems (also see 4.7), this might not be an effective strategy. Ideally, one would like to have causal reasoning, which is considered to be a hallmark of human beings. The objective here is to learn the causal relationship between entities, which allows to reason about underlying rules and regularities (also see 4.9). But already simpler forms of reasoning are superior to pure memorization. In general, reasoning allows to come up with solutions to a problem over time through the dynamical process of thought, which takes more time to compute but requires less memory capacity and is more universal. This could naturally lead to domain-specific specialization of sub-systems within the nervous system, each sub-system developing and using different forms of algorithms for reasoning. The work presented in Rapp and Nawrot, 2020 shows one form of simple but effective reasoning, where single sensory cues are learned and dynamical memory recall of multiple instances can be used to reason about complex composite objects (e.g. odor plumes).

## 4.3 Bio-inspired computing systems of the future.

### 4.3.1 Biological neural systems as substrate for general-purpose computations.

The computational and architectural features of biological neural systems have very appealing properties that are interesting to be considered within the context of general purpose computation. These systems are highly energy-efficient and are able to perform complex computations with very limited power consumption. Given the exponential growth of digital computer systems and their growing energy consumption in combination with prevalent climate issues, it is important to search for alternative energy efficient computation systems of the future. Understanding the computational principles of nervous systems could help to develop general purpose computational systems of the future that still remain small and portable compared to quantum computers.

Additionally, the computational power, in terms of processing speed, of transistor-based digital computer processors are hitting the fabrication limits of physics. The only solution to this problem is by going from single-CPU to multi-CPU (or multicore CPUs) and thus to distributed parallel processing. This type of processing brings its own new challenges to the table that are yet not sufficiently solved. The major problem here is the so-called *Von-Neumann bottleneck*. *Von-Neumann* refers to the specific computing architecture used by modern computers, which consists of the components memory (RAM), processor (CPU) and periphery. In order to perform computations, data constantly needs to be moved back and forth between CPU and memory. This leads to a bottleneck when multiple CPUs or CPU cores are introduced, since the memory bus is shared by all computing units. On the other hand, the computational substrate of biology using neurons and synapses, performs memory-local computations. The memory component are synapses which are local to the computing unit and thus does not have this limitation. There are efforts in the field of neuromorphic computing to develop such in-silico systems. However, it remains elusive how and what could be a suitable programming paradigm to use these systems for general purpose computations. An interesting way to pursue could be to think about how a Turing machine based on neurons, synapses and action potentials could be built. Turing machines have the proofed property being able to compute any arbitrary function, which is a necessary feature to perform general purpose computations.

### 4.3.2 Nervous systems and current distributed computing paradigms.

The most powerful and largest distributed computer system as of today which can be considered to have approximately the size and complexity of a biological brain, is the global internet. A large network of $\sim$ 100 Million [1] interconnected individual servers, performing computations to provide useful services commonly referred to as cloud-computing. Considering that each server an average is equipped with 5 CPU cores, this is approximately the number of neurons of a rat's brain. The development and exponential growth of the internet over the past years has shown, that the scale and type of this highly distributed system requires completely different and new computing paradigms: efficient and low-latency communication to exchange data, consensus to remain consistent, resilience and fault tolerance to name just a few. Although it is a working system, all of the mentioned problems have not been sufficiently solved as of today. Remarkably, all of these properties can be found in biological brains. It's a highly distributed and complex network of computing units (neurons) with low-latency and efficient communication via synapses and action potentials. It is able to maintain consensus across different brain regions, sub-systems and brain hemispheres. Furthermore it is fault tolerant and resilient to a sufficient degree and able to perform self-healing to some extend when lesioned.

Thus, there are a lot of conceptual things that could be learned in terms of distributed computing systems from understanding the computational principles of nervous systems. If we are able to extract the underlying, generic principles we might be able to adapt and transfer them to build systems that share the same properties but are much smaller and energy efficient as today's transistor-based computers. An example of such a principle that has been found so far is sparse coding, which has been found to exist in different sensory systems (Kuffler, 1953; Hartline, Wagner, and Ratliff, 1956; Fuchs and Drown, 1984; Oswald, Schiff, and Reyes, 2006). However, as of today we have not been able to transfer this concept and make use of it to build general purpose computers with this property. Thus, the approach taken in Rapp, Nawrot, and Stern, 2020 and Rapp and Nawrot, 2020 to use biologically realistic models applied to non-biological problems and combining biological realistic models with more machine learning motivated models and learning rules, can serve as a first step towards this line of research.

## 4.4 An algorithmic perspective on behavioral problem-solving strategies.

This section considers the conceptual relationship between algorithms, a core discipline of computer science, and Biology. In particular the duality between algorithms and behavioral strategies to solve problems relevant to biological organisms are considered.

An algorithm is a process or set of rules to be followed in calculations or other problem-solving operations[2]. This can be translated to behavior, which is a dynamical process or sequence of actions like motor commands to be executed by an organism to solve a biologically relevant problem. Thus, observed behavior can be considered as the result of an internal algorithm executed by an organisms nervous system. Behavior in general and the various instances of particular behavioral strategies can

[1] `https://www.racksolutions.com/news/data-center-trends/400-million-new%2Dservers-might-be-needed-by-2020/`

[2] Definition according to the Oxford English Dictionary.

be considered as algorithmic solutions developed by nature. It allows organisms to adapt and survive in changing environments. In this process, over time, evolution has come up with highly optimized systems to execute algorithms under different environmental conditions. There are two major types of algorithms to be considered in this context. The first type can be related to the family of innate behavior which serves as a set of base strategies, which might not be optimal but guarantee survival. The second type is behavior that is learned by experience and executed during lifetime. The first class of behaviors appear to be more deterministic and systematic and are thus easier to understand compared to the second class, which are much more complex. Most of them are thought to emerge from optimizing for a specific objective, which in biological systems likely is a dynamic quantity that depends on environmental conditions, metabolic state, internal states and many other things. Studying and understanding algorithms of both families eventually allows us to come up with equivalent implementations useful for artificial systems.

On the other hand, considering biological systems from the perspective of algorithmic problem solving and execution thereof, can help us to reveal and formalize computational principles of the brain that underlie complex behavior. It can provide a new perspective and access to new methods and concepts to study brain function and computation. Output of algorithms employed by the nervous system can be observed experimentally in terms of actual behavior. The question to be asked is, given the computational substrate (i.e. specific neuronal architectures) what is the required information (i.e. sensory cues) and how does it need to be processed (i.e. representation, integration, learning and recombination) to execute the observed behavioral algorithm. While the internally employed algorithm can be deterministic, the observed output can be stochastic due to the stochasticity of the input from the environment. Algorithms provide a systematic approach by breaking down the overall problem into smaller pieces. This approach further demands going back and forth between experimental and computational work to collect behavioral data and come up with possible implementations in a nervous system. It prevents computational work to go astray towards studying highly abstract problems, it prevents experimental work from conducting too simplistic and unnatural behavioral experiments and generally forces both parties to collaborate and interact.

This approach has been taken in Rapp and Nawrot, 2020 and proved successful. The studied behavior is the *cast&surge* strategy executed by many flying insects to solve the foraging problem. In *Drosophila melanogaster* two distinct sensori-motor reflexes have been identified (Breugel and Dickinson, 2014) which requires generation of motor commands to execute U-turn and upwind surge. Similar strategies have been found in many other flying insects. Furthermore, the neural architecture and computational features of the involved primary sensory system of olfaction have been identified by experimental studies. Considering the sensory environment of turbulent odor plumes, the problem turns out to be quite complex and impossible to be solved directly by the olfactory system. In Rapp and Nawrot, 2020 the problem is broken down to the simpler problem of learning to process atomic sensory cues, which relate to the encounter of a single filament of an odor plume. This simple problem can be learned rapidly and efficiently by the olfactory model system. The solution to this problem, in terms of a trained model, is then used to estimate the spatial extent and center line of the entire cone-shaped odor plume. This is achieved by dynamical memory recall of many instances of the simpler problem of processing atomic sensory cues. In other words, the odor plume can be considered as a composite object of simpler atomic sensory cues. It turns out, that this approach yields a neural signal that can directly be used to derive optimal motor commands to execute a *cast&surge* strategy. This signal can be interpreted as an unnormalized likelihood function of atomic sensory cues in time. It's important to mention, that this approach is very general and works independently of the plume's actual statistical structure. The only two assumptions made by this approach are, that plume's constitute of filaments and that these filaments are present within a plume's boundary and absent otherwise. The results of using this approach in Rapp and Nawrot, 2020 suggest, that it can serve as a general principle of problem solving in neural systems and its potential to generalize to other sensory systems and behavioral problems.

## 4.5 Learning in biological systems and machine learning: statistical vs. representation learning

In this section, the similarities and differences between machine learning, statistical learning and learning in biological systems are discussed. This is an important aspect to consider, as both disciplines are concerned with very similar problems of object detection, classification, prediction and feature extraction. There is great potential, that insights from one discipline can advance progress in the other field and vice versa.

Machine learning and statistical learning make heavy use of symbolic math, probabilities and advanced mathematical concepts. These concepts are generally not available to the nervous system and it's often unclear how such methods could be implemented in biological neural systems. Particularly, probabilities are used in artificial neural networks as a unified representation within these models. They are continuous and thus mathematically very convenient and also easy to interpret by humans. The latter is important when analyzing such models and interpreting their output.

### 4.5.1 Beyond unified statistical representations.

Although, there is statistical evidence that the brain acts within the framework of some of such advanced mathematical concepts, in order to build functional models of the brain the actual implementation matters. Additionally, it seems that the brain does not maintain a unified representation of stimuli (Abbott, DePasquale, and Memmesheimer, 2016) across different sensory systems. In fact, it's an important feature of sensory systems, to process and transform initial representations at the periphery into more efficient neural codes for further processing in higher brain areas. For example, the olfactory system used in Rapp and Nawrot, 2020. In contrast to machine learning methods, the type of preprocessing has strong implications on learning in biological neural systems, as there is no unified representation that can be optimized for, particularly not probabilities. Representation of stimuli in the brain are heterogeneous due to different types of neural coding schemes, for example temporal, latency and rate coding. Thus, learning in biological models with spiking neurons can be considered to be closer to representation learning. The output of a spiking neuron is correlated to some external sensory or internal memory representation. In some cases this can be a representation of the probability of sensory input, but in general it is not and can code for arbitrary sensory features, numerosity or other higher cognitive concepts (Rapp, Nawrot, and Stern, 2020). To build functional models, the neural activity of spikes or spike-trains should not only be considered statistically, but as quantities carrying semantic information, for example associations with sensory cues or memories. As a consequence, overall it's unlikely that there is a single global objective function and learning rule to be optimized for in the brain, as suggested by machine learning methods minimizing the empirical risk. Furthermore it suggests, that different types of learning rules are at play in the brain exploiting specific computational properties and neural codes within different sub-systems.

### 4.5.2 Systematic limitations of pure statistical learning.

Very recently, the field of machine learning has seen a shift from building purely perceptual systems (classifiers, pattern recognition and predictive models) towards agent-based systems and reinforcement learning. The key difference is, that agents require interaction with their environment, which is similar to execution of behavior by a biological organism. This introduces additional problems to learning systems that have not been studied intensively within the machine learning community, namely generalization and coping with non-stationarities due to continuously changing environments. Contrary, biological systems are very good at dealing with non-stationarities and changing environments.

Machine learning is strongly rooted in statistical learning and thus is based on methods and assumptions of statistical learning. The major principles are empirical risk minimization and maximum likelihood estimation (MLE) which can be considered as learning probability distributions and function approximators. To make these optimization problems mathematically tractable, the fundamental assumption of **i**ndependent and **i**dentically **d**istributed (iid) data is introduced. This property can artificially be enforced by shuffling the data.

*"Nature does not shuffle the data, so we shouldn't either"*

Leon Bottou (Facebook AI Research, ICML 2019 Keynote)

However, this assumption introduces a strong systematic limitation, where any learning algorithm based on this assumption cannot generalize to data samples that have zero probability under the training distribution. Overcoming this issue is called out-of-distribution (OOD) learning and is related to the capability of a learning algorithms to generalize. As an example, the method presented in chapter 3 makes this problem explicit. Any sample that is not captured by the marginal density function $f(t)$ by definition has zero probability under the model. However, in the context of the application of this method this property is explicitly used to perform detection of anomalous events like epileptic seizures, which have low or close to zero probability of occurring. This allows the model to be trained on a large corpus of regular data which is cheap to collect and identify rare events that do not fall into the training distribution.

A potential solution to go from iid to OOD and deal with non-stationarities for both, machine learning and functional models of biological systems, is the concept of compositionality which is discussed in 4.7.

## 4.6 Biological plausablity of gradient-based plasticity rules.

In Rapp, Nawrot, and Stern, 2020 and in Rapp and Nawrot, 2020 the same type of neuron model and learning rule has been used, the Multispike Tempotron (Gütig, 2016). This learning rule is formulated in the framework of gradient-descent to adjust a spiking neuron's synaptic weights to elicit a specific number of output spikes.

This requires differentiation of the neuron's membrane potential, which in general is considered to be biologically implausible. It is not clear how this operation could be implemented in a real biological neuron or linked to known biochemical processes. However, due to the success of this method in the field of Deep Learning it has recently become popular to consider this method in the context of plasticity rules for spiking neurons and networks thereof. Several methods (Lee, Delbruck, and Pfeiffer, 2016; Huh and Sejnowski, 2017; Nicola and Clopath, 2017; Bohte, Kok, and Poutré, 2000; Zenke and Ganguli, 2018; Neftci, Mostafa, and Zenke, 2019; Taherkhani et al., 2019) have been proposed along this line also in the search of explanations how this could be implemented by biochemical processes that exist in nervous systems. In Samadi, Lillicrap, and Tweed, 2017 feedback weights are used to backpropagate global error signals along a different set of synapses as used during the forward pass. An alternative way is using eligability traces (Gerstner et al., 2018) which has been used in Bellec et al., 2019 to develop a biologically plausible backpropagation through time algorithm to train spiking recurrent neural networks.

In contrast to the backpropagation algorithm used to train artificial neural networks and its equivalents for training spiking neural networks, the Multispike Tempotron learning rule is local, even when building networks of multiple neurons. It does not require backpropagation of global error gradients. The gradient-based learning rule can be approximated on the basis of the correlation between presynaptic activity and postsynaptic voltage. An eligability coefficient can be derived from this for each synapse and a threshold value is used to select a small subset of eligible synapses that undergo potentiation or suppression. In biological neurons, this form of elegibilities could be realized on the basis of intracellular calcium signals. These signals are sensitive to coordinated pre- and post-synaptic activity through the voltage dependence of NMDA receptors. There is documented evidence that the induction of long-term changes of synapses require specific threshold levels of these calcium signals (Artola, Bröcher, and Singer, 1990; Cummings et al., 1996; Malenka and Nicoll, 1999). As such, the approximated Multispike Tempotron learning rule can be linked to known biological processes and thus be considered to be biologically plausible despite its theoretical nature. Although, for computer simulations the gradient-based rule is more convenient to implement and faster to run. While the Multispike Tempotron is rather abstract, it can serve as a biologically plausible abstraction in a setting when the precise biological implementations and biochemical mechanisms are not fully known yet or when the objective is to abstract from different implementations across several species, that share the same computational feature. Having learning rules that have two equivalent formulations in two domains, one of which optimized for numerical computations in digital computers and one optimized for computations (or plausability) in biological systems, are an appealing direction of research. It allows to systematically study and quantify the difference in learning efficiency and performance and can provide insights why biological brains use a different implementation of the same plasticity rule which might be specific to their computational substrate.

## 4.7 Compositionality as a generic principle of computation in higher order brain areas.

This section discusses compositionality as a generic conceptual framework for learning and computation in higher brain areas and across species.

Compositionality (or Frege principle) is a principle rooted in mathematics, logic and linguistics. Very generally, this principle states, that the meaning of a complex expression is a function of the meanings of its constituent expressions (Hintikka, 1984). As such it is related the divide-and-conquer algorithm introduced previously which can be seen as one algorithmic implementation of this principle applicable to a family of problems. In the context of sensory processing, the constituent expressions can be considered to be locally available sensory cues of the environment and a complex expression can be thought of as a macroscopic problem or objective of interest, for example an odor plume or a complex visual scene. Recombination of the constituent expressions allows to reason about the complex macroscopic problem. Different recombinations of the same cues may refer to different macroscopic problems which means compositionality can also be context sensitive.

Compositionality can be achieved in different forms. For example the layered hierarchical structure of neural systems can be considered as spatial or structural compositionality, as it is the case for the processing layers of the olfactory system used in Rapp and Nawrot, 2020. It is also the major form of compositionality used in Deep Learning. Another form is on the level of representations, for example in the sequential inspection strategy studied in Rapp, Nawrot, and Stern, 2020. Here the overall image representation is transformed into a sequence of representations of smaller images.

However, the most natural form is compositionality in time, which might be one principle used by nervous systems. For example, the simultaneous movement of both eyes between phases of fixation points (saccades), can be considered as temporal compositionality (of focus) to perceive visual scenes (Gegenfurtner, 2016). The sequential inspection strategy used by insects and in Rapp, Nawrot, and Stern, 2020 to solve numerical cognition tasks, can also be seen as temporal compositionality as well as the sensory experience of odor filaments to reason about odor plumes as used in Rapp and Nawrot, 2020. In general, by introducing the additional dimension of time any (static) input can be transformed into a sequence of smaller inputs or expressions. Thus, there are many ways how to define what makes a constitutent expression within the framework of compositionality. However, the challenge remains to determine what is a *suitable* constitutent expression, such that the amount of information related to the original problem increases when recombining them. This raises the hypothesis if biological systems successfully use and combine multiple forms of compositionality, in particular temporal compositionality as time is inherent to biological organisms, for example by development and experience based learning throughout life-time.

Additionally, this principle is one possible solution to overcome the generalization issue introduced by the iid assumption in statistical learning as discussed in 4.5. The counting MNIST task introduced in Rapp, Nawrot, and Stern, 2020 shows, that by learning the concept of a single digit one, instead of the distribution of possible occurrences, allows to generalize counting to images that have not been included in the training data. In general, it allows to generalize from iid to OOD, as new images are just a composition of previously learned individual instances. The same principle can be applied to more abstract concepts. For example a face is composed of eyes, mouth, nose and ears. By learning the individual concepts of eye, mouth, nose, ear and how to recombine them, allows to generalize to faces (human and animal) without explicitly learning a distribution over all possible faces.

In summary, the principle of compositionality is general and applicable to many computational aspects of biological systems, including sensation, learning, memory and cognition. Revealing the underlying computational mechanisms and neural

| olfactory system | Drosophila | Honeybee | Mouse |
|---|---|---|---|
| **# of receptor types** | $\sim 52$ | $\sim 160$ | $\sim 1800$ |
| **# of stimuli combinations (cues)** | $2^{52}$ | $2^{160}$ | $2^{1800}$ |
| simple visual system | Drosophila | Honeybee | Mouse |
| **# of object detectors** | 2<br>bright & dark objects | 5<br>bright & dark objects<br>+ 3 geometric shapes | ?? |
| **# of stimuli combinations (cues)** | $2^{2}$ | $2^{5}$ | $2^{??}$ |
| size of *intrinsic universe* | $2^{54}$ | $2^{165}$ | $>> 2^{1800}$ |

TABLE 4.1: Concept of an organisms *intrinsic universe* equipped with two sensory systems, exemplary for *Drosohila melanogaster*, Honeybee and mouse. An olfactory system that can detect a set of odorants and combinations thereof (based on by the number of glomeruli found in Drosophila (Vosshall and Stocker, 2007), honeybee (Galizia and Menzel, 2001) and mouse (Potter et al., 2001)) and a simple (hypothetical) visual system that can sense the presence or absence of a fixed number of objects. The expressive power in terms of total possible stimuli combinations that can be sensed by each sensory system is given as powers of two when considering the the binary case where a sensory cue can only be present or absent. The *intrinsic universe* is defined as the combined expressive power of all sensory systems. In the binary case the size of the *intrinsic universe* is given by summation of the exponents of the two sensory systems. The size of an organisms *intrinsic universe* follows a combinatorial explosion with each additional sense.

implementations that could allow this principle to emerge, can provide fundamental insights about higher order brain computations and has the potential to explain many aspects of intelligence, in biological and artificial systems.

## 4.8 Concept of the intrinsic universe.

In the context of sensory processing the principle of compositionality can be used as an effective way to deal with combinatorial explosions of sensory cues, limited memory capacity and multi-modal sensory integration across species. To this end the concept of an organisms *intrinsic universe* is introduced in table 4.1. It refers to the total number of possible stimuli combinations, or sensory cues, that can be detected by an organisms sensory systems. The size of this intrinsic universe follows a combinatorial explosion of possible sensory cues. Learning a distribution over all of them would require a large memory capacity and collection of enough samples is not feasible, even if only a single instance gets collected for each possible sensory cue. This is very costly in terms of neuronal resources as well as energy and timewise. However, many of the possible sensory cues might not be relevant to the organism or do not occur within its environment. Thus, the learning problem can be reduced to only learn the relevant sensory cues and store them in memory. This reduces both, the learning problem and the size of required memory. For example, despite learning only a small subset of 10 sensory cues and allocating memory for them, compositionality allows to increase the effective perceptual power to $2^{10} = 2024$ unique composite cues at each time step. Where perceptual power refers to the number of cues and unique combinations thereof that the system is able to recognize.

| **sensory cue** | $t_{-2}$ | $t_{-1}$ | $t$ |
|---|---|---|---|
| Odor 1 | 1 | 1 | 0 |
| Odor 2 | 0 | 0 | 0 |
| Odor 3 | 0 | 1 | 0 |
| ... | ... | ... | ... |
| Odor 52 | 0 | 0 | 1 |
| Bright object | 1 | 0 | 0 |
| Dark object | 0 | 1 | 1 |

TABLE 4.2: Applying the concept of compositionality to sensory processing of sensory cues in time can be considered as a language processing problem. At each time step $t$ an organisms sensory systems (here olfaction and vision) detect sensory cues. Presence of a cue is indicated by a 1 and absence by 0. The set of sensed cues at each time step is considered as a sensory *token* or *word*. An example token for time step $t_{-2}$ is indicated by the red column, represented as a binary vector. A sequence of sensory tokens over time constitute a *sentence* of sensory information. The problem of sensory processing can then be translated into a language processing problem.

## 4.9 Sensory processing as a language processing problem.

Applying compositionality in time can further increases the effective perceptual power while maintaining moderate memory capacity requirements. This can be considered by translating sensory processing into a language processing problem. Where at any time step $t$ a sensory cue *token* (or word) is sensed which in time leads to a stream of tokens that constitute *sentences* (also see table 4.2). Different temporal sequences of the same sensory tokens can relate to different objects in the environment. Additionally, the meaning of a specific sensory token can be different depending on its contextual position in relation to its surrounding sensory tokens. For example the same sensory token could refer to either a predator or a mate depending on its context. This perspective on sensory processing also covers symbol manipulation, a concept of cognitive science (Gärdenfors, 2005; Cangelosi, 2005), and a neural implementation which includes plasticity could deal with grounded and ungrounded symbols.

On the abstract level of language processing, the problem of sensory processing translates to the question: Is sentence $\mathtt{S}$, composed of tokens $\mathtt{s}_i$, a valid sentence of language $\mathtt{L}$ defined by a (formal) grammar $P$. For example, such a grammar can be considered as a set of rules, that need to be satisfied by a stream of sensory tokens, that characterize the sensory experience related to a predator or food. This framework allows to use methods and theories of Formal Systems (a sub-field of theoretical computer science) to study and model sensory processing as a stream of discrete events. Most of these methods could allow for a direct implementation using spiking neurons and networks thereof, e.g. pushdown automaton by using population codes and working memory. The problem of learning can then be studied on two different levels: learning of single sensory cues (associative learning and memory) and learning of a valid grammar based on experienced sequences of sensory tokens. The later can be related to reinforcement-like learning and allows to extract underlying causes, regularities and rules. Additionally, it can also be used to study the phenomenon of imagination by generating new token sequences internally for example from an internal random recurrent neural network or another internal world model.

## 4.10 Applied machine learning for statistical and functional understanding of large scale brain structures.

This section discusses how applied machine learning can be used to aid the understanding of brain function of larger scale brain systems, where it's not possible to build detailed functional models. This leads to a statistical understanding of the brain and can help to identify computational principals, function of certain subsystems and diseases related to their dysfunction.

Traditionally, experimental neuroscience uses statistical methods and handcrafted neural signals derived from recorded activity to study brain function. As the amount of data is rapidly increasing due to technological advances, this rather manual process becomes unfeasible. Additionally, the dimensionality of the data often is much larger compared to the number of collected trials (# neurons >> N of samples), which poses an additional problem to most statistical methods. This trend is often referred to as data-driven neuroscience.

As such, data-driven approaches to automate the extraction of relevant correlations and reducing the dimensionality (Byron et al., 2009) of the data using advanced statistical and machine learning methods are becoming increasingly popular (Barrett, Morcos, and Macke, 2019; Sussillo et al., 2016; Berens et al., 2018; Panzeri et al., 2015; Pillow, Ahmadian, and Paninski, 2011). However, a major problem with these automated methods remains the interpretability of the results and the features learned by the methods. Relating them back to the original data and experimental conditions is key to answer and understand the underlying biological research question. An additional caveat is, that these self-learning methods always produce some output, as they will always learn *something* by definition. However, the question whether the output is sane and correct often is not straight forward and needs careful investigation.

In Nashef et al., 2017 statistical as well as machine learning methods have been used to study the functional impact of the CTC system on motor control in primates. Principal component analysis has been used to identify area specific response patterns of MUA signals. Additionally, a random forest classifier has been trained to decode cortical sites based on single MUA and LFP responses.

Apart from understanding the brain, machine learning methods have also been successfully used to build brain machine interfaces (BMIs) and neural prosthetics (Pandarinath et al., 2018; Pandarinath et al., 2017). The CPESN method proposed in chapter 3 follows a related line of research where machine learning methods can also be used as clinically relevant diagnostic tools. Here the objective is not to control prosthetic devices but detect disease related rare events in noisy physiological signals, for example EEG and ECoG.

# Bibliography

Abbott, L F, Brian DePasquale, and Raoul-Martin Memmesheimer (2016). "Building functional networks of spiking model neurons". In: *Nature Neuroscience* 19.3, pp. 350–355. ISSN: 1546-1726. DOI: 10.1038/nn.4241. URL: http://dx.doi.org/10.1038/nn.4241.

Artola, A., S. Bröcher, and W. Singer (1990). "Different voltage-dependent thresholds for inducing long-term depression and long-term potentiation in slices of rat visual cortex". In: *Nature* 347.6288, pp. 69–72. ISSN: 1476-4687. DOI: 10.1038/347069a0. URL: http://dx.doi.org/10.1038/347069a0.

Aso, Yoshinori et al. (2014). "The neuronal architecture of the mushroom body provides a logic for associative learning". In: *Elife* 3, e04577.

Barrett, David GT, Ari S Morcos, and Jakob H Macke (2019). "Analyzing biological and artificial neural networks: challenges with opportunities for synergy?" In: *Current Opinion in Neurobiology* 55. Machine Learning, Big Data, and Neuroscience, pp. 55 –64. ISSN: 0959-4388. DOI: https://doi.org/10.1016/j.conb.2019.01.007. URL: http://www.sciencedirect.com/science/article/pii/S0959438818301569.

Bellec, Guillaume et al. (2019). *Biologically inspired alternatives to backpropagation through time for learning in recurrent neural nets*. arXiv: 1901.09049 [cs.NE].

Berens, Philipp et al. (2018). "Community-based benchmarking improves spike rate inference from two-photon calcium imaging data". In: *PLOS Computational Biology* 14.5. Ed. by DanielEditor Bush, e1006157. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1006157. URL: http://dx.doi.org/10.1371/journal.pcbi.1006157.

Bohte, Sander M., Joost N. Kok, and Han La Poutré (2000). "SpikeProp: backpropagation for networks of spiking neurons". In: *ESANN*.

Breugel, Floris van and Michael H. Dickinson (2014). "Plume-Tracking Behavior of Flying Drosophila Emerges from a Set of Distinct Sensory-Motor Reflexes". In: *Current Biology* 24.3, pp. 274–286. DOI: 10.1016/j.cub.2013.12.023. URL: https://doi.org/10.1016/j.cub.2013.12.023.

Byron, M Yu et al. (2009). "Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity". In: *Advances in neural information processing systems*, pp. 1881–1888.

Cangelosi, Angelo (2005). "Chapter 32 - APPROACHES TO GROUNDING SYMBOLS IN PERCEPTUAL AND SENSORIMOTOR CATEGORIES". In: *Handbook of Categorization in Cognitive Science*. Ed. by Henri Cohen and Claire Lefebvre. Oxford: Elsevier Science Ltd, pp. 719 –737. ISBN: 978-0-08-044612-7. DOI: https://doi.org/10.1016/B978-008044612-7/50087-1. URL: http://www.sciencedirect.com/science/article/pii/B9780080446127500871.

Caron, Sophie JC et al. (2013). "Random convergence of olfactory inputs in the Drosophila mushroom body". In: *Nature* 497.7447, p. 113.

Cummings, Jennifer A et al. (1996). "Ca2+ Signaling Requirements for Long-Term Depression in the Hippocampus". In: *Neuron* 16.4, pp. 825–833. ISSN: 0896-6273.

DOI: 10.1016/s0896-6273(00)80102-6. URL: http://dx.doi.org/10.1016/S0896-6273(00)80102-6.

Demmer, Heike and Peter Kloppenburg (2009). “Intrinsic membrane properties and inhibitory synaptic input of kenyon cells as mechanisms for sparse coding?” In: *Journal of neurophysiology* 102.3, pp. 1538–1550.

Dierckx, Paul (Apr. 1995). *Curve and Surface fitting with Splines*. Clarendon Press.

Egea-Weiss, Alexander et al. (2018). “High precision of spike timing across olfactory receptor neurons allows rapid odor coding in Drosophila”. In: *iScience* 4, pp. 76–83.

Eichler, Katharina et al. (Aug. 2017). “The complete connectome of a learning and memory centre in an insect brain”. In: *Nature* 548.7666, pp. 175–182. ISSN: 1476-4687. DOI: 10.1038/nature23455. URL: http://dx.doi.org/10.1038/nature23455.

Feindt, Michael (2008). “A Neural Bayesian Estimator for Conditional Probability Densities”. URL: https://arxiv.org/abs/physics/0402093.

Fuchs, Jannon L and Paul B Drown (1984). “Two-point discriminability: relation to properties of the somatosensory system”. In: *Somatosensory research* 2.2, pp. 163–169.

Galizia, C.Giovanni and Randolf Menzel (2001). “The role of glomeruli in the neural representation of odours: results from optical recording studies”. In: *Journal of Insect Physiology* 47.2, pp. 115 –130. ISSN: 0022-1910. DOI: https://doi.org/10.1016/S0022-1910(00)00106-2. URL: http://www.sciencedirect.com/science/article/pii/S0022191000001062.

Gärdenfors, Peter (2005). “Chapter 37 - CONCEPT LEARNING AND NONMONOTONIC REASONING11This chapter is an expanded and revised version of Gärdenfors (2001)”. In: *Handbook of Categorization in Cognitive Science*. Ed. by Henri Cohen and Claire Lefebvre. Oxford: Elsevier Science Ltd, pp. 823 –843. ISBN: 978-0-08-044612-7. DOI: https://doi.org/10.1016/B978-008044612-7/50092-5. URL: http://www.sciencedirect.com/science/article/pii/B9780080446127500925.

Gegenfurtner, Karl R. (2016). “The Interaction Between Vision and Eye Movements”. In: *Perception* 45.12. PMID: 27383394, pp. 1333–1357. DOI: 10.1177/0301006616657097. eprint: https://doi.org/10.1177/0301006616657097. URL: https://doi.org/10.1177/0301006616657097.

Gerstner, Wulfram et al. (2018). “Eligibility Traces and Plasticity on Behavioral Time Scales: Experimental Support of NeoHebbian Three-Factor Learning Rules”. In: *Front. Neural Circuits*.

Gidon, Albert et al. (2020). “Dendritic action potentials and computation in human layer 2/3 cortical neurons”. In: *Science* 367.6473, pp. 83–87.

Gütig, Robert (2016). “Spiking neurons can discover predictive features by aggregate-label learning”. In: *Science* 351.6277.

Hartline, H K, Henry G Wagner, and Floyd Ratliff (1956). “Inhibition in the eye of Limulus”. In: *The Journal of general physiology* 39.5, pp. 651–673.

Hige, Toshihide et al. (Sept. 2015). “Plasticity-driven individualization of olfactory coding in mushroom body output neurons”. In: *Nature* 526, 258 EP –. URL: http://dx.doi.org/10.1038/nature15396.

Hintikka, Jaakko (1984). “A hundred years later: The rise and fall of Frege’s influence in language theory”. In: *Synthese*, pp. 27–49.

Hochreiter, Sepp and Jürgen Schmidhuber (Nov. 1997). “Long Short-Term Memory”. In: *Neural Comput.* 9.8, pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. URL: http://dx.doi.org/10.1162/neco.1997.9.8.1735.

Howard, +Scarlett+R. et al. (2018). “Numerical ordering of zero in honey bees”. In: *Science* 360.6393, pp. 1124–1126.

Howbert, J. Jeffry et al. (2014). “Forecasting Seizures in Dogs with Naturally Occurring Epilepsy”. In: *PLoS ONE* 9.1. Ed. by MaximEditor Bazhenov, e81920. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0081920. URL: http://dx.doi.org/10.1371/journal.pone.0081920.

Huh, Dongsung and Terrence J. Sejnowski (June 2017). “Gradient Descent for Spiking Neural Networks”. In: eprint: 1706.04698. URL: https://arxiv.org/abs/1706.04698.

Inada, Kengo, Yoshiko Tsuchimoto, and Hokto Kazama (July 2017). “Origins of Cell-Type-Specific Olfactory Processing in the Drosophila Mushroom Body Circuit”. In: *Neuron* 95.2, 357–367.e4. ISSN: 0896-6273. DOI: 10.1016/j.neuron.2017.06.039. URL: http://dx.doi.org/10.1016/j.neuron.2017.06.039.

Ito, Iori et al. (Sept. 2008). “Sparse odor representation and olfactory learning”. In: *Nature Neuroscience* 11, 1177 EP –. URL: http://dx.doi.org/10.1038/nn.2192.

Jaeger, Herbert (2001). “The ”echo state” approach to analysing and training recurrent neural networks”. In: *GMD Report, German National Research Center for Information Technology* 148.

Kazama, Hokto and Rachel I Wilson (2009). “Origins of correlated activity in an olfactory circuit”. In: *Nature Neuroscience* 12.9, pp. 1136–1144. ISSN: 1546-1726. DOI: 10.1038/nn.2376. URL: http://dx.doi.org/10.1038/nn.2376.

Kloppenburg, Peter and Martin Paul Nawrot (Dec. 2017). “Neural Coding: Sparse but On Time”. In: *Current Biology* 24.19, R957–R959. DOI: 10.1016/j.cub.2014.08.041. URL: http://dx.doi.org/10.1016/j.cub.2014.08.041.

Kuffler, Stephen W (1953). “Discharge patterns and functional organization of mammalian retina”. In: *Journal of neurophysiology* 16.1, pp. 37–68.

LeCun, Yann and Corinna Cortes (2010). “MNIST handwritten digit database”. In: URL: http://yann.lecun.com/exdb/mnist/.

Lee, Jun Haeng, Tobi Delbruck, and Michael Pfeiffer (Aug. 2016). “Training Deep Spiking Neural Networks using Backpropagation”. In: eprint: 1608.08782. URL: https://arxiv.org/abs/1608.08782.

Malenka, Robert C. and Roger A. Nicoll (1999). “Long-Term Potentiation–A Decade of Progress?” In: *Science* 285.5435, pp. 1870–1874.

Nagel, Katherine I and Rachel I Wilson (2011). “Biophysical mechanisms underlying olfactory receptor neuron dynamics”. In: *Nature neuroscience* 14.2, p. 208.

Nashef, Abdulraheem et al. (Nov. 2017). “Area-specific processing of cerebellar-thalamo-cortical information in primates”. In: *Biological Cybernetics* 112.1-2, pp. 141–152. ISSN: 1432-0770. DOI: 10.1007/s00422-017-0738-6. URL: http://dx.doi.org/10.1007/s00422-017-0738-6.

Neftci, Emre O., Hesham Mostafa, and Friedemann Zenke (2019). *Surrogate Gradient Learning in Spiking Neural Networks*. arXiv: 1901.09948 [cs.NE].

Nicola, Wilten and Claudia Clopath (2017). “Supervised learning in spiking neural networks with FORCE training”. In: *Nature Communications* 8.1, p. 2208. DOI: 10.1038/s41467-017-01827-3. URL: https://doi.org/10.1038/s41467-017-01827-3.

Oord, Aaron van den et al. (2016). “Wavenet: A generative model for raw audio”. In: *arXiv preprint arXiv:1609.03499*.

Oswald, Anne-Marie M, Max L Schiff, and Alex D Reyes (2006). “Synaptic mechanisms underlying auditory processing”. In: *Current opinion in neurobiology* 16.4, pp. 371–376.

Pamir, Evren et al. (Sept. 2014). "Rapid learning dynamics in individual honeybees during classical conditioning". In: *Frontiers in Behavioral Neuroscience* 8. ISSN: 1662-5153. DOI: 10.3389/fnbeh.2014.00313. URL: http://dx.doi.org/10.3389/fnbeh.2014.00313.

Pandarinath, Chethan et al. (2017). "High performance communication by people with paralysis using an intracortical brain-computer interface". In: *eLife* 6. ISSN: 2050-084X. DOI: 10.7554/elife.18554. URL: http://dx.doi.org/10.7554/eLife.18554.

Pandarinath, Chethan et al. (2018). "Inferring single-trial neural population dynamics using sequential auto-encoders". In: *Nature Methods* 15.10, pp. 805–815. ISSN: 1548-7105. DOI: 10.1038/s41592-018-0109-9. URL: http://dx.doi.org/10.1038/s41592-018-0109-9.

Panzeri, Stefano et al. (2015). "Neural population coding: combining insights from microscopic and mass signals". In: *Trends in Cognitive Sciences* 19.3, pp. 162 –172. ISSN: 1364-6613. DOI: https://doi.org/10.1016/j.tics.2015.01.002. URL: http://www.sciencedirect.com/science/article/pii/S1364661315000030.

Pillow, Jonathan W., Yashar Ahmadian, and Liam Paninski (2011). "Model-Based Decoding, Information Estimation, and Change-Point Detection Techniques for Multineuron Spike Trains". In: *Neural Computation* 23.1, pp. 1–45. ISSN: 1530-888X. DOI: 10.1162/neco_a_00058. URL: http://dx.doi.org/10.1162/neco_a_00058.

Potter, Steve M. et al. (2001). "Structure and Emergence of Specific Olfactory Glomeruli in the Mouse". In: *J. Neurosci.; Journal of Neuroscience* 21.24, pp. 9713–9723.

Rapp, Hannes and Martin P. Nawrot (2020). "A spiking neural program for sensorymotor control during foraging in flying insects."

Rapp, Hannes, Martin Paul Nawrot, and Merav Stern (Jan. 2020). "Numerical cognition based on precise counting with a single spiking neuron." In: *iScience*, p. 100852. ISSN: 2589-0042. DOI: 10.1016/j.isci.2020.100852. URL: http://dx.doi.org/10.1016/j.isci.2020.100852.

Samadi, Arash, Timothy P Lillicrap, and Douglas Blair Tweed (2017). "Deep Learning with Dynamic Spiking Neurons and Fixed Feedback Weights". In: *Neural Computation* 29, pp. 578–602.

Shiozaki, Hiroshi M, Kazumi Ohta, and Hokto Kazama (2020). "A Multi-regional Network Encoding Heading and Steering Maneuvers in Drosophila". In: *Neuron*.

Sussillo, David et al. (2016). *LFADS - Latent Factor Analysis via Dynamical Systems*. arXiv: 1608.06315 [cs.LG].

Szyszka, Paul et al. (2014). "High-speed odor transduction and pulse tracking by insect olfactory receptor neurons". In: *Proceedings of the National Academy of Sciences* 111.47, pp. 16925–16930. DOI: 10.1073/pnas.1412051111. eprint: http://www.pnas.org/content/111/47/16925.full.pdf. URL: http://www.pnas.org/content/111/47/16925.abstract.

Taherkhani, Aboozar et al. (2019). "A review of learning in biologically plausible spiking neural networks". In: *Neural networks : the official journal of the International Neural Network Society* 122, pp. 253–272.

Takemura, Shin-ya et al. (2017). "A connectome of a learning and memory center in the adult Drosophila brain". In: *eLife* 6. ISSN: 2050-084X. DOI: 10.7554/elife.26975. URL: http://dx.doi.org/10.7554/eLife.26975.

Toyoizumi, T. and L.F. Abbott (2011). "Beyond the edge of chaos: Amplification and temporal integration by recurrent networks in the chaotic regime". In: *Physical Review E* 84.5. DOI: 10.1103/PhysRevE.84.051908.

Vosshall, Leslie B. and Reinhard F. Stocker (2007). “Molecular Architecture of Smell and Taste inDrosophila”. In: *Annual Review of Neuroscience* 30.1, pp. 505–533. ISSN: 1545-4126. DOI: 10.1146/annurev.neuro.30.051606.094306. URL: http://dx.doi.org/10.1146/annurev.neuro.30.051606.094306.

Wilson, Rachel I (2013). “Early olfactory processing in Drosophila: mechanisms and principles”. In: *Annual review of neuroscience* 36, pp. 217–241.

Wulsin, D F et al. (2011). “Modeling electroencephalography waveforms with semi-supervised deep belief nets: fast classification and anomaly measurement”. In: *J Neural Eng* 8.3, p. 036015. DOI: 10.1088/1741-2560/8/3/036015.

Xu, C. Shan et al. (2020). “A Connectome of the Adult Drosophila Central Brain”. In: *bioRxiv*. DOI: 10.1101/2020.01.21.911859. eprint: https://www.biorxiv.org/content/early/2020/01/21/2020.01.21.911859.full.pdf. URL: https://www.biorxiv.org/content/early/2020/01/21/2020.01.21.911859.

Yu, Fisher and Vladlen Koltun (2015). *Multi-Scale Context Aggregation by Dilated Convolutions*. arXiv: 1511.07122 [cs.CV].

Zenke, Friedemann and Surya Ganguli (2018). “SuperSpike: Supervised Learning in Multilayer Spiking Neural Networks”. In: *Neural Computation* 30.6. PMID: 29652587, pp. 1514–1541. DOI: 10.1162/neco_a_01086. eprint: https://doi.org/10.1162/neco_a_01086. URL: https://doi.org/10.1162/neco_a_01086.

UNIVERSITÄT ZU KÖLN

# *Zusammenfassung*

Faculty of Mathematics and Natural Sciences
Institut für Zoologie

Doctor of Philosophy

**Spiking neural models & machine learning for systems neuroscience**

von Hannes Andreas RAPP

Die Fähigkeiten des Lernens, der Wahrnehmung sowie in der Umwelt zu navigieren, mit ihr zu interagieren und durch unser Verhalten zu verändern, sind charakteristische Eigenschaften der künstlichen Intelligenz und von intelligenten biologischen Systemen. Ziel ist es zu verstehen, wie aus einem Nervensystems intelligentes Verhalten und Handeln entstehen kann. Die Arbeit im Rahmen dieser Dissertation schlägt hierzu vor, Lernen, Kognition und Verhalten gleichzeitig als einheitliches System zu betrachten um zu einem ganzheitlichen Verständnis zu gelangen. Der Ansatz in dieser Arbeit basiert auf dem Erstellen von detailierten funktionalen Computermodellen von biologischen neuronalen Systemen. Diese Systeme sollen hierbei gleichzeitig in der Lage sein sensorische Informationen zu verarbeiten, zu Lernen, zu speichern und diese in Steuerungssignale zur Ausübung des gewünschten Verhaltens zu übersetzen. Desweiteren untersucht die Arbeit, inwiefern biologische neuronale Systeme, basierend auf Neuronen, Dendriten und Aktionspotentialen, als Alternative zu aktuellen Methoden des maschinellen Lernens herangezogen werden können, um relevante Probleme im Feld der künstlichen Intelligenz (KI) zu lösen. Es wird hinterfragt, ob die aktuell verwendeten künstlichen Systeme aus raten-basierten Neuronen eine zu einfache Abstraktion darstellen und wichtige Rechenleistung zu Gunsten von mathematisch einfacher zu handhabenden Modellen geopfert wird. Hierzu wird die Rechenleistung von einzelnen, spikenden Neuronen untersucht und auf ihre Leistungsfähigkeit zur Lösung von biologisch motivierten kognitiven und KI relevanten Problemen geprüft. Weiterhin werden detaillierte, funktionale Netzwerke aus solchen Neuronen erstellt, basierend auf der Physiologie sensorischer Systeme in Insekten. Die Modelle werden anschließend verwendet um das biologisch relevante Verhalten der Nahrungssuche von fliegenden Insekten zu untersuchen und nachzubilden. Die zusammengefassten Ergebnisse und gesammelten Erkenntnisse dieser Arbeit sind neu und von hoher Relevanz für die Forschungsgebiete des maschinellen Lernens, Neurowissenschaften und Computational Systems Neuroscience. Die Arbeit schließt mit einen Ausblick, wie Methoden des maschinellen Lernens verwendet werden können, um ein statistisches Verständnis über Systeme in höher entwickelteren Gehirnen zu erlangen, für die aktuell keine funktionalen und detailierten Computermodelle erstellt werden können. Im konkreten Fall wird untersucht, welche Funktion dem *cerebellar-thalamo-cortical* System bei der Steuerung und Koordination von Bewegung zukommt.

UNIVERSITÄT ZU KÖLN

# *Abstract*

Faculty of Mathematics and Natural Sciences
Institut für Zoologie

Doctor of Philosophy

**Spiking neural models & machine learning for systems neuroscience**

by Hannes Andreas RAPP

Learning, cognition and the ability to navigate, interact and manipulate the world around us by performing appropriate behavior are hallmarks of artificial as well as biological intelligence. In order to understand how intelligent behavior can emerge from computations of neural systems, this thesis suggests to consider and study learning, cognition and behavior simultaneously to obtain an integrative understanding. This involves building detailed functional computational models of nervous systems that can cope with sensory processing, learning, memory and motor control to drive appropriate behavior. The work further considers how the biological computational substrate of neurons, dendrites and action potentials can be successfully used as an alternative to current artificial systems to solve machine learning problems. It challenges the simplification of currently used rate-based artificial neurons, where computational power is sacrificed by mathematical convenience and statistical learning. To this end, the thesis explores single spiking neuron computations for cognition and machine learning problems as well as detailed functional networks thereof that can solve the biologically relevant foraging behavior in flying insects. The obtained results and insights are new and relevant for machine learning, neuroscience and computational systems neuroscience. The thesis concludes by providing an outlook how application of current machine learning methods can be used to obtain a statistical understanding of larger scale brain systems. In particular, by investigating the functional role of the cerebellar-thalamo-cortical system for motor control in primates.

# Appendix A

# Own contributions to publications

## A.1 Publication #1: Rapp, Nawrot, and Stern, 2020

I conducted and contributed the following work and materials to this publication:

- 80% of original research idea and conceptualization of it
- improvement of the used learning rule
- conceptualization of machine learning task and the biologically motivated cognition task
- 95% of results and computational experiments
- 90% of results figures and additional artwork
- 80% of the first draft of the manuscript
- 50% of revised manuscript

## A.2 Publication #2: Rapp and Nawrot, 2020

I conducted and contributed the following work and materials to this publication:

- original research idea and conceptualization of it
- all methods and computational experiments
- all results
- all results figures and additional artwork
- 100% of the first draft of the manuscript

## A.3 Publication #3: Nashef et al., 2017

I conducted and contributed the following work and materials to this publication:

- statistical analysis (PCA) of MUA data and time-frequency analysis of LFP data
- choice of machine learning method and model tuning/training for decoding of cortical sites based on MUA nad LFP signal
- results figures 2,3,4,5,6

- contributed data to results figures 7 and 8 (50% each)
- writing of the methods section for the above mentioned methods
- contributed to the writing of the results section

## A.4 Contribution to work listed under chapter 3

I conducted and contributed the following work and materials to this publication:

- 50% of original research idea and conceptualization of it
- all results and computational experiments
- all results figures
- 100% writing of current draft of manuscript (shown in chapter 3)

# Eidesstattliche Erklärung

Ich versichere, dass ich die von mir vorgelegte Dissertation selbständig angefertigt, die benutzten Quellen und Hilfsmittel vollständig angegeben und die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken im Wortlaut oder dem Sinn nach entnommen sind, in jedem Einzelfall als Entlehnung kenntlich gemacht habe; dass diese Dissertation noch keiner anderen Fakultät oder Universität zur Prüfung vorgelegen hat; dass sie - abgesehen von unten angegebenen Teilpublikationen - noch nicht veröffentlicht worden ist, sowie, dass ich eine solche Veröffentlichung vor Abschluss des Promotionsverfahrens nicht vornehmen werde.
Die Bestimmungen der Promotionsordnung sind mir bekannt. Die von mir vorgelegte Dissertation ist von Dr. Martin Paul NAWROT betreut worden.

Unterschrift: H. Rapp

Datum: 25.06.2020

## SUMMARY

Learning, cognition and the ability to navigate, interact and manipulate the world around us by performing appropriate behavior are hallmarks of artificial as well as biological intelligence.

In order to understand how intelligent behavior can emerge from computations of neural systems, this thesis suggests to consider and study learning, cognition and behavior simultaneously to obtain an integrative understanding. This involves building detailed functional computational models of nervous systems that can cope with sensory processing, learning, memory and motor control to drive appropriate behavior.

This work further considers how the biological computational substrate of neurons, dendrites and action potentials can be successfully used as an alternative to current artificial systems to solve machine learning problems.

**Dissertation of Hannes Rapp, 2020**
**Computational Biology**
**University of Cologne**
**Supervisor:** Prof. Dr. Martin Paul Nawrot