Neural network modelling of the influence of channelopathies on reflex visual attention

Gravier, Alexandre

2014

Gravier, A. (2014). Neural network modelling of the influence of channelopathies on reflex visual attention. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/65485>

<https://doi.org/10.32657/10356/65485>

Neural Network Modelling of the Influence of Channelopathies on Reflex Visual Attention

ALEXANDRE GRAVIER

School of Computer Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2014

À Marcos
À la joie
À la beauté des rêves
À la mélancolie
À l'espoir qui nous tient
À la santé du feu
Et de la flamme
À ton étoile

— Noir Désir

FOREWORD

This thesis is part of the independent research I conducted at Nanyang Technological University (NTU) from August 2008 to December 2013. The research was funded by a ROAR scholarship from NTU until July 2012.

Computational neuroscience is a discipline at the crossroad of neuroscience, mathematics, and computer science. In particular, computational intelligence, computational modelling and artificial neural networks take central roles in serving the primary goal of computational neuroscience: understanding biological neural systems.

As I am foremost a computer scientist and engineer, I take an engineering approach to problem-solving and address research issues for which a computational approach is most suited. My contributions are centred on the development of neural networks models of neurocognitive theories.

The thesis is divided in four parts. Although each part is self-contained, the overall structure from Part i to Part iii forms a narrative arc that follows the progression of my doctoral research work. The first part is introductory, the core contributions are presented in Parts ii and iii, and the last part regroups appendices.

Margin notes accompany the text. They highlight important points of the paragraphs they juxtapose in order to help browsing the dissertation.

There is also an alphabetically sorted list of acronyms in the end of the front matter (pp. 16 – 18). This list contains short reminders of the meaning of most acronyms used in the text. In the original softcopy of the thesis, each acronym in the main body constitutes a direct hyperlink to the corresponding entry in the table. Similarly, references to figures, equations, referenced works and footnotes are hyperlinks to those elements. In printouts of the thesis, it might be useful to keep a bookmark at the beginning of the list of accronyms on p. 16, for quick reference. All acronyms are also defined and often discussed more extensively in the text.

Finally, in complement to the abstract on next page, each part of the thesis is preceded by a short summary of its contents.

ABSTRACT

Research in computational neuroscience has the potential to bridge the gap between the neuroscientific descriptions of neural structures and dynamics, and the psychological models of behaviour and mental states. In particular, simulated neural network models with output or dynamics that are interpretable in terms of behaviour can help to understand neurodynamics and assess the plausibility of vertical theories of cognition. This thesis presents two contributions to the domain: *EVAC*, a model of emergent visual attention in presence of calcium channelopathy, and *PyCogMo*, a model learning framework.

By modelling channelopathy, *EVAC* constitutes an effort towards identifying the possible causes of autism. The network structure embodies the dual pathways model of cortical processing of visual input, with reflex attention as an emergent property of neural interactions. *EVAC* extends a model developed by O'Reilly and Munakata [1] by introducing attention shift in a larger-scale network and applying a phenomenological model of channelopathy. The simulation of *EVAC* results in testable behavioural predictions on reflex attention shift tasks inspired by Posner [2]. In presence of a distractor, the channelopathic network's rate of failure to shift attention is lower than the control network's, but overall, the control network exhibits a lower classification error rate. The simulation results also show differences in task-relative reaction times between control and channelopathic networks. The attention shift timings inferred from the model are consistent with studies of attention shift in autistic children.

Often, generative methods, like the simulation of the model of *EVAC*, require to train the synaptic weights using specialised neural networks learning algorithms. In the *EVAC* model, the weights are learnt with the *Leabra* algorithm [3]. *Emergent* is the only neural simulator implementing the *Leabra* algorithm. Consequently, the *EVAC* network could not be developed in a simulator-independent manner that would allow it to be run on a variety of implementations to improve the reliability of the results. While *PyNN* [4] is a software framework that offers the possibility to specify and run a model on several otherwise incompatible simulators, training those networks remains a simulator-dependent task, for which *PyNN* is not suited. Hence, *PyCogMo*, a novel learning framework based on *PyNN*, was developed to address this limitation. It enables modellers to implement and use learning algorithms with several simulators. *PyCogMo* constitutes a step towards improving the reproducibility of computational experiments across simulators, hence a valuable methodological support for computational neuroscientists.

But if we ask where precisely in the brain that point of view is located, the simple assumptions that work so well on larger scales of space and time break down. It is now quite clear that there is no single point in the brain where all information funnels in, and this fact has some far from obvious consequences.

— Daniel Dennett [5]

ACKNOWLEDGMENTS

It is a difficult exercise in brevity to acknowledge in one page the help and support I have received during more than five years.

First, I would like to express my deep gratitude to my supervisors, Assoc. Prof. Chai Quek, Prof. Włodzisław Duch, and Prof. Abdul Wahab, without whom nothing would have been possible. I am also grateful towards Assoc. Prof. Chen I-Ming for his support, and to Assoc. Prof. Ong Yew Soon who, together with the C2i staff, welcomed me into the pleasant and stimulating research environment of the Centre for Computational Intelligence.

I am very thankful to my loving wife, Dr Joanna Gravier-Rymaszewska, who supported me during my PhD, not only in spirit but also by her perspicacious scientific advice.

I would also not have made it through these studies without my friends and colleagues at NTU: Dr Reza Khosrowabadi, Dr Eng Yeow Cheu, Dr Whye Loon Tung, Dr Richard Oentaryo, Dr Sau Wai Tung, Mr Jair Zhou Weigui, Dr Chan Wai Ting, Dr Stanley Nam Nguyen, Mr Javan Tan, Dr Raj Gupta, Dr Chaoying Tan, Dr Dilip Prasad, Mr Deepak Subramanian, Dr Qin Li, Dr Behrouz Saghafi, Dr Elahe Farahzadeh, Dr Amin Talei, Dr Mohammad Shahjamali, Dr Shukai Li, Dr Ehsan Shahrian, Ms Jingyu Quek, Ms Chunyang Sun, Ms Lili Liu, Mr Bingbing Li, Ms Aye Aye Lwin, and all others whose names I am guilty of not including here.

Finally, I would like to thank my parents and my family for their love and encouragements throughout the years.

CONTENTS

---

i	INTRODUCTION	19
1	BACKGROUND ON THE COMPUTATIONAL STUDY OF THE BRAIN	21
1.1	The last frontier	21
1.1.1	Human behaviour and the brain: an enduring enigma	21
1.1.2	Complexity and the neural basis of behaviour	22
1.2	Medical challenges	23
1.3	Societal aspects	24
1.4	Motivation for the computational modelling of the brain	26
1.5	Dynamical computational models of brain mechanisms	27
2	MOTIVATION AND CONTRIBUTIONS OF THE THESIS	31
2.1	Introduction and motivation	31
2.1.1	Overview of the research approach in Part ii	31
2.1.2	Overview of the research approach in Part iii	34
2.2	Contributions of the thesis	36
2.2.1	A model of visual attention capture in presence of calcium channelopathy	36
2.2.2	PyCogMo: a novel learning framework for computational neuroscience	37
ii	A MODEL OF VISUAL ATTENTION CAPTURE	39
3	MOTIVATION AND BACKGROUND	41
3.1	Objectives and research questions	41
3.2	Autism	44
3.2.1	Neuroscience of autism	44
3.2.2	Structural and developmental abnormalities	45
3.2.3	Complementing psychological theories	47
3.3	Calcium channelopathies and visual attention in autism	48
3.3.1	Calcium homeostasis and signalling in autism	48
3.3.2	Visual pathways in the brain	50
3.3.3	Bottom-up visual attention	55
3.3.4	Models of attention and ASD	65
3.3.5	Summary table of computational levels of study	71
3.4	Potential contributions of research	76
3.4.1	Neuroscientific importance	76
3.4.2	Technological importance	77

4	EXPERIMENT, MODEL ARCHITECTURE AND IMPLEMENTATION	79
4.1	Modelled environment and organism	79
4.2	Task/trial structure	81
4.2.1	The Neutral condition	82
4.2.2	The Gap condition	83
4.2.3	The Overlap condition	84
4.3	Model architecture	86
4.3.1	Model of reflex visual attention	86
4.3.2	Implementation framework	86
4.3.3	Neural unit architecture	87
4.3.4	Layer structure and connectivity	94
4.3.5	Biologically plausible learning	103
4.3.6	Learning and testing implementation	107
5	RESULTS	111
5.1	Unit behaviour	111
5.2	Network learning	115
5.3	Network behaviour	117
5.3.1	Statistical analysis	117
5.3.2	Network performance	130
6	DISCUSSION	133
6.1	Recapitulation of the study	133
6.2	Theories embodied in the model	134
6.3	Results interpretation and model predictions	135
6.4	Research contributions	138
6.5	Limitations of the model	139
6.6	Future experimental research	141
6.6.1	Experimental protocol	141
6.6.2	Related prior study	143
6.7	Future theoretical research	144
6.7.1	Methodological work	145
iii	PYCOGMO: A LEARNING FRAMEWORK FOR COMPUTATIONAL NEUROSCIENCE	147
7	MOTIVATION AND BACKGROUND	149
7.1	Review of related literature and software	149
7.2	Overview of PyCogMo	152
7.3	Potential contributions of research	153
7.3.1	Synaptic plasticity	153
7.3.2	The problem of synaptic consolidation in PyNN	154
7.3.3	Scientific goals of PyCogMo	155
7.3.4	Design goals of PyCogMo	156

8	IMPLEMENTATION AND TESTING	159
8.1	System Design	159
8.2	Components design	160
8.2.1	The learning module	160
8.2.2	The scheduling module	163
8.2.3	The visualisation module	165
8.2.4	The utilities component	168
8.3	Implementation and testing	169
8.3.1	Implementation of the scheduling module	169
8.3.2	Implementation of the learning module	172
8.3.3	Implementation of the visualisation module	174
8.3.4	Implementation of the utilities module	177
8.3.5	Testing	178
9	DISCUSSION	179
9.1	Issues and limitations	179
9.1.1	System design	179
9.2	Research contributions	181
9.3	Future work	181
iv	APPENDICES	185
A	PRINCIPLES OF ASTRUCTURAL NEURON MODELLING	187
A.1	From equivalent circuit models to the Hodgkin-Huxley model	187
A.1.1	Simple electrical circuits	188
A.1.2	Modelling a patch of membrane	188
A.2	From Hudgkin-Huxley to simplified models	196
A.2.1	The Integrate-and-Fire model	197
A.2.2	A more elegant model: The exponential Integrate-and-Fire	198
A.2.3	An most elegant model: the theta neuron	198
A.2.4	An extra dimension	198
A.3	More realistic models	199
B	NEURAL SIMULATION SOFTWARE	201
	CONCLUSION OF THE THESIS	205
	BIBLIOGRAPHY	207

LIST OF FIGURES

Part ii

Figure 3.1	Receptive fields in topographical cortical maps	53
Figure 3.2	Visual pathways in the human brain	54
Figure 4.1	Organisation of the Neutral task condition	82
Figure 4.2	Organisation of the Gap task condition	84
Figure 4.3	Organisation of the Overlap task condition	85
Figure 4.4	Overview of the structure of the visual attention network	96
Figure 4.5	Illustration of the processing performed by the lateral geniculate nucleus (LGN) maps of emergent visual attention under channelopathy (EVAC)	97
Figure 4.6	The six symbols that the network learns	98
Figure 4.7	Abstraction of cortical intra and inter-map connectivity	100
Figure 4.8	Functional overview of the implementation of the experiment in Emergent	109
Figure 5.1	Unit activation trace with and without hysteresis and accommodation	113
Figure 5.2	Control vs. channelopathic unit activation trace	115
Figure 5.3	Average SSE during 700 epochs of learning of the visual attention network	116
Figure 5.4	Histograms of the response times of the control network in repeated overlap, neutral, and gap conditions	121
Figure 5.5	Histograms of the response times of the channelopathic network in repeated overlap, neutral, and gap conditions	122
Figure 6.1	Schema of the experimental setup	142

Part iii

Figure 8.1	Diagram of the top-level system design of PyCogMo	160
Figure 8.2	Diagram of the division of the learning functions by timescale in the nettraining module	162
Figure 8.3	Guideline for the implementation of event processes in the scheduling module	164
Figure 8.4	Class hierarchy and interaction between processes in the visualization module	167
Figure 8.5	Class diagram illustrating the implementation of the events supporting the Leabra algorithm	171
Figure 8.6	Simple interface build using PyCogMo's visualization module	176

Part iv

Figure A.1	Schema of the electrical model of a neuron	187
Figure A.2	Circuit model of a passive cell membrane	189
Figure A.3	Circuit model of an active cell membrane	190
Figure A.4	Circuit model of an active cell membrane with passive and voltage-gated channels	192
Figure A.5	Plot of the subunits time constants steady states in function of membrane voltage	195
Figure A.6	Plot of the subunits steady state activation and inactivation variables in function of membrane voltage	196
Figure A.7	Phase plane diagram of the FitzHugh-Nagumo model	199

LIST OF TABLES

Part ii

Table 3.1	Models of neural mechanisms and applications to modelling brain disorders	72
Table 4.1	Symbols used in the equations describing computations performed by units.	88
Table 5.1	Hysteresis and accommodation parameters in Emergent	112
Table 5.2	Descriptive statistical summary of the timing output of the attention switch network	118
Table 5.3	Summary of the results of two-way analysis of variance (ANOVA) on the attention shift network response times.	123
Table 5.4	Summary of the results of two-way ANOVA on the Box-Cox transformed response times of the attention shift network.	124
Table 5.5	MLE-fitted log-normal distributions and parametric moments of reaction timing data	125
Table 5.6	Classification error rates and activation failure rates of the attention network	130

Part iv

Table A.1	Nernst potential of major ions in the neuron's environment	191
Table B.1	Overview of some available software	201

LISTINGS

Listing 8.1	The <code>InputPresentation</code> event process class and interface method 170
Listing 8.2	The implementation of k-WTA SOM learning in the <code>nettraining</code> module 173
Listing 8.3	Sample program making use of PyCogMo's visualization module 174
Listing 8.4	Definitions of the <code>get_input_layer</code> and <code>get_rate_encoder</code> functions by partial application of <code>population_adapter_provider</code> . 177

ACRONYMS

ACT-R	Adaptive Control of Thought – Rational (see [6])
AGI	artificial general intelligence
AI	artificial intelligence
ANN	artificial neural network
ANOVA	analysis of variance
AP	action potential
API	application programming interface
ASD	autism spectrum disorders
CHL	contrastive Hebbian learning
CNS	central nervous system
CO	central oscillator
CPCA	conditional principal components analysis
DES	discrete event simulator

EEG	electroencephalogram
EOG	electrooculogram
ERP	event-related potential
EVAC	emergent visual attention under channelopathy – the acronym given to the model presented in Part ii of the dissertation
FEF	frontal eye fields
fMRI	functional magnetic resonance imagery – measures changes in blood flow in the brain, correlating with functional activity
GABA	γ -Aminobutyric acid – a prevalent inhibitory neurotransmitter
GeneRec	Generalised Recirculation
GIL	global interpreter lock – mechanism used by the Python interpreter to prevent the concurrent execution of Python in the same environment
GUI	graphical user interface
GWAS	genome-wide association studies
IaF	integrate-and-fire model of neuron dynamics
IPC	inter-process communication
IQ	intelligence quotient measured using a Weshler-like test
IT	inferior temporal cortex
k-WTA	k-winners-take-all
LGN	lateral geniculate nucleus
LISSOM	laterally interconnected, synergetically self-organizing maps (see [7])
LTCC	L-type voltage-gated calcium channel
LTD	long-term depression of the electrical potential of the neuron membrane
LTP	long-term potentiation of the neuron membrane
LTP/D	any long-term change in synaptic strength (either LTP or LTD)
MDL	model description language – a formalism used to describe neurons and networks

MLE	maximum likelihood estimation
MT	middle temporal cortex (same as V ₅)
NMDA	N-methyl-D-aspartate – an excitatory neurotransmitter with a role in LTP
ODE	ordinary differential equation
OSL	object selection layer
PDD	pervasive developmental disorder
PDP	Parallel and Distributed Processing (see [8])
PO	peripheral oscillator
PPT	posterior parietal cortex
RF	receptive field
SMR	standardised mortality ratio
SOAR	State, Operator And Result (see [9])
SOM	self-organised feature map, also called Kohonen map [10]
SSE	sum-of-square error
STDP	spike timing dependent plasticity
TMS	transcranial magnetic stimulation
UML	Unified Modelling Language
V ₁	primary visual cortex
V ₂	secondary visual cortex
V ₄	third extrastriate area in the ventral visual stream
V ₅	middle temporal cortex (same as MT)
VCS	version control system
VGCC	voltage-gated calcium channels
VOR	vestibulo-ocular reflex
VTK	the Visualization Toolkit – scientific visualisation library
WTA	winner-takes-all

Part I

INTRODUCTION

This first part of the dissertation draws an overview of the field of computational neuroscience and of my research. I first expound the reasons and means to model and understand higher cognitive functions, highlighting the exceptional stakes in play at the societal level, and show why computational modelling is a particularly valuable tool for the neural and behavioural sciences. In the second chapter, I place my work in this context by presenting the motivations behind my research and summarising the contributions of this thesis.

BACKGROUND ON THE COMPUTATIONAL STUDY OF THE BRAIN

1.1 THE LAST FRONTIER

1.1.1 *Human behaviour and the brain: an enduring enigma*

Human behaviour has ever been intriguing to curious minds. Albeit among the most accessible objects of study, it has long remained one of the most elusive phenomena under rational scrutiny.

The brain is known as the organ of cognition since the IVth century B.C. at least.¹ The medieval period produced a number of anatomical studies of the peripheral and central nervous systems, and some attempts at assigning functions to the tissues observed. From the high middle ages to the Renaissance, a lot of work was done on the functioning the eye at a macroscopic level. The rate of discoveries gradually increased starting from the XVIIth century, as Robert Hooke's works on microscopy [11] laid down the foundations of what was to become the cell theory, 150 years later.

In contrast, the abstract study of human psychology – that is, the formal description and modelling of behaviour in relative abstraction from cognition as a physical process – flourished more easily. Psychology in that broad sense encompasses formalizations of folk knowledge, appearing as a much more natural discipline than the study of the related biology. Folk psychology is a mostly innate ability [12], making its epistemology and vocabulary immediately accessible and consequently dangerously transparent to critical evaluation.² Like for the philosophical foundations or brain sciences, the foundations of psychology date back to ancient Greece. However, the psyche was then most often studied in a philosophical context, relating it to the metaphysical concepts of soul. Later, medieval physicians would develop and relate psychology to mental disorders, laying the path of

Early brain studies, from ancient Greece to the high middle ages, were mostly anatomical.

Early psychology was more prominent but disconnected from biology.

-
- ¹ Alcmaeon of Croton is the first to identify the brain as the seat of understanding and to distinguish understanding from perception. Among other Greek philosophers, Hippocrates (460-379 B.C.) and Plato (427-347 B.C.) accept his view and also consider brain as the seat of intelligence.
  - ² As humans, we attribute mental states, and we reason on beliefs, desires, using these atomic elements and laws of our theory of mind in the scientific study of human behaviour. Besides the dangerous possibility of using intuitively pleasing but false knowledge in a formal context, this conceptually simpler frame of self-reference may not be as practical from a systematic standpoint, when attempting to relate behaviour and neurobiology, as illustrated by the difficulty of finding neural correlates of psychological experience.

clinical psychology. But, for lack of proper observation instruments, the link between psyche and brain function remained tenuous.

*Brain discoveries are
determined by
technical progress.*

Until the XXth century, functional studies of the brain were limited by the serendipity of adequate lesion cases, as technical means of direct observation were insufficient, and consequently, macroscopic anatomical knowledge remained the fastest to progress. Electrical stimulation at first, and then simple recordings of brain activity, let researchers understand that electricity is a key element of brain function. The exploratory research of the brain of earlier eras started benefiting from new advances in physics, (optics, electricity) and biology. This ladder effect is observed all along the history of neuroscience until today, and reveals how complex multidisciplinary fields of study benefit from advances in the many fundamental sciences upon which they rest. Subsequent sections of this part of the thesis highlight that effect for computational neuroscience, with foundations in branches of mathematics³ and physics and engineering.⁴

*The discovery of the
intrinsic complexity
of the brain*

It is in the XIXth and XXth centuries, with the central nervous system clearly established as the main determinant of behaviour, that neuron emerged as its elementary mechanistic units.⁵ Hence, the question of *what* (as in “what is the endogenous cause of behaviour”) quickly shifted to an increasingly puzzling *how*. How indeed does the extremely intricate nervous machinery give rise to coherent behaviour? The brain is very peculiar, compared to the rest of the body, in that finer knowledge of the anatomy of the central nervous system (CNS) only very marginally helps understanding its functioning, while usually revealing more of its essential complexity. Progress in the understanding of other organs had usually been incidental to discoveries of their constituent parts, with lesion studies answering much of the *hows*. But as the brain revealed its structure at the macro- and microscopic levels, and lesion studies laboriously helped assign functional label to some of these areas, it appeared more and more evident that unlike other organs, the key to its understanding did not lie in its individual components, but rather in their staggeringly complex interactions.

1.1.2 Complexity and the neural basis of behaviour

Complexity is a stumbling block common to all brain sciences. There are several reasons for that. First, the network of neurons that constitutes the CNS and which, as a whole, is responsible for individual behaviour, appears as a tightly coupled and integrated system. The *in vitro* study of an individual neuron’s response to artificial inputs

³ including probability, information theory, analysis and chaos theory

⁴ electricity, molecular dynamics, computer science

⁵ The elementarity of brain cells was postulated by Santiago Ramón y Cajal in the late XIXth century

does very few in the way of understanding the functioning of its surrounding tissue *in vivo*. Many of the computationally useful properties of biological neural networks are emergent. Understanding such complex emergent system requires observing its constituent parts at a sufficient resolution in time and in space. This brings the second major obstacle of neurosciences: the technical inability to observe the detailed state of a sufficiently large population of neurons *in vivo* still hinders brain research. Third, understanding biological neural networks is slowed by the comparatively simple mathematical tools at disposition. While algebra, probability theory and statistics are among the more mature mathematical tools used to study and understand highly dimensional connectionist systems, chaos theory, information theory and the study of high-dimensional dynamical systems are relatively new fields of mathematics. However, as often in science, progress in one helps the other, and recent works in computational neuroscience have led to further studies in mathematics.⁶

*Epistemological
obstacles in
neuroscience*

The inadequacy of observation tools has ever been the foremost limiting factor of brain sciences. This is illustrated by the XIXth century's debate between proponents of the reticulum doctrine,⁷ who believed that brain tissue is continuous, and advocates of the neuron doctrine, who thought that it consists of numerous discrete cells [14]. At that time, the limitation was in the magnification power of microscopes. The observation of neurons by Purkinje and Dieters using the newly developed achromatic compound microscope, and Ramón y Cajal's exceptional works on the neuron settled the debate and established the neuron doctrine [15].

*Example: the
aggregate field
theory in the XIXth
century*

Nowadays, the limiting factors are in the observation of the *dynamics* of the brain at the same microscopic level that challenged XIXth century scientists. While the pioneers of brain sciences faced the task of discovering its microstructure, current day challenges lie both in the electrochemical domain and in understanding the differentiated processing mechanisms of seemingly uniform areas of continuous neural tissue.

1.2 MEDICAL CHALLENGES

The massive efforts of the research community to decipher the inner workings of the brain are justified by more than simple scientific curiosity. The medical statistics on brain trauma and illnesses reveal the inadequacy of current knowledge when it comes to developing medical applications, relative to other organs and medical disciplines. For instance, Alzheimer's disease is the sixth leading cause of death in

⁶ See for instance publications by Izhikevich on the dynamics of coupled oscillators in neural network models [13].

⁷ Anecdotically, there is a region in the brain stem called *Formatio reticularis* – the name originates from that time.

The insufficient understanding of the brain comes to light through substandard brain-related morbidity and mortality statistics.

the United States, a situation that has not improved but worsened: the age-adjusted death rate from Alzheimer's disease increased by 39 percent from 2000 through 2010 in the United States [16]. Considering mental health in general, a 2005 study by Kessler et al. estimated that 26.2 percent of adult Americans suffer from a mental disorder each year [17]. Regarding traumatic brain lesions, a 2003 study by Bruns & Hauser [18] revealed that the long-term disabilities (epilepsy) and mortality of traumatic brain injury are major public health problem worldwide. This is well illustrated by a more recent study for Australia [19] that places the standardised mortality ratio (SMR)⁸ of patients released after hospitalisation for traumatic brain injury at 12.3 in the first year, and 3.19 within 15 years. Cancer is another example of the struggle of medical research with the brain, as the one-year survival rate to malignant brain tumour (any type) in adults is only 36%. Finally, other less understood diseases affecting the central nervous system have high prevalence and societal cost. In particular, autism is an often severely disabling cognitive developmental disorder of still unknown etiology affecting 2% of children in the USA in 2012 [20].

All these issues are very disparate, but the poor statistics associated with them have a common factor of insufficient knowledge about the brain. Comparing success rates related to brain procedures to success rates related to other organs may appear dubious, because the brain is a much more complex organ. Yet, this is the point: the brain is a much more ambitious challenge to medical research because of its inherent complexity. Thus, all scientific efforts towards understanding the fundamentals of the nervous system and its relation to psychology are potential improvements of technical and scientific brain-related questions in medicine.

1.3 SOCIETAL ASPECTS

The potential impact of cognitive neuroscience on social structures is tremendous.

Elucidating the link between the brain and the mind would have groundbreaking repercussions on society, because such knowledge would have a deep philosophical impact. With a clear theoretical understanding of the share of environmental determinism in one's actions, the notion of free will would no more be of the realm of metaphysics, but a quantifiable property of the mind. Of course, the consequences at the societal level would be tremendous, as each institution would need to take this new knowledge into practical consideration:⁹ as examples, dispensing justice, governing a society, the balance of democracy, and education would be the most obviously affected.

⁸ The SMR is calculated as the ratio of mortality in the study group to that expected of an age- and sex-matched control group.

⁹ *Neuroethics* is a recently coined term naming the new discipline of the study of individual and societal implications of such knowledge.

The above is unlikely to happen all at once, but new knowledge would gradually settle, first in the scientific community, then in society, as each discovery and technical advance is popularised; but the end-result would still eventually bring radical changes to society; besides, on a historical timescale, modern technological and scientific advancement happens at an accelerating rate, putting higher pressure on society.

On a grander scale still, current brain research will possibly have a decisive role in human history and evolution. There are two distinct claims in that sentence. First, that future human history may be influenced by neuroscientific research, and second, that the branch of the homo genus that currently constitutes homo sapiens may evolve differently in function of current research. The first claim relates to the eventual creation of artificial intelligent entities, while the latter concerns transhumanism. These two claims are grounded in the central and self-determining role of intelligence in human development.

Regarding the first claim, the reason why cognitive neuroscience is potentially among the strongest influencers of long-term human prospects is that the model of intelligence used in researching artificial systems is defined by example, i.e. humans are currently the only known intelligent entities, and human behaviour therefore defines intelligent behaviour [21]. In practice, principles of neural and behavioural sciences are already applied in artificial intelligence (AI) research (e.g. neural networks, reinforcement learning), and this influence will likely strengthen as more of the biological mechanisms of intelligence are uncovered. However, the development of gradually more powerful intelligent systems is not only an important advance for human development, it is also not without risk for the species. Specifically, the unethical or uncontrolled development of artificial general intelligence (AGI) could lead to a mass extinction event [22].

The argument that neuroscience is also a significant potential influencer of human evolution through transhumanism is not as far-fetched as it might first seem. Technological self-improvement as described by this movement is nothing but a projection of the research agenda of contemporary biomedicine. Gene therapy, “performance enhancement” drugs and electronic implants are not fictional anymore, while the reflection on their dangers, on which ethical bounds we should try to enforce clearly lags. Neural and behavioural sciences then have a clear role in these developments: all knowledge of the mechanisms of the brain and their relation with the mind will eventually be used for technological or genetic enhancement, which, if carried out without restraint and reflection, could lead to disastrous developments, including increased inequality, and the resurgence of eugenic practices.

Cognitive neuroscience will impact human development due to its central role in defining intelligence.

The responsibility of cognitive neuroscience in transhumanist prospects and related matters of artificial evolution

1.4 MOTIVATION FOR THE COMPUTATIONAL MODELLING OF THE BRAIN

The study of neural systems using mathematical and computational models has been enabled by the development of sufficiently powerful numerical techniques and computing machinery, which is why it is a new and fast developing discipline. But what needs does it address? This section gives a summary of the main motivations behind computational neuroscience, while Section 1.5 presents an overview of the research approaches.

The brain, as a computational system, is particularly suited to computational modelling.

There are several motivations for the development of computable models of mental and neuronal processes. The most fundamental reason why such processes naturally lend themselves to computational modelling is that they are, in essence, information processors. In addition, they are information processors performing computations that many engineering disciplines aim at reproducing, an observation that has motivated many engineers to take an interest in the brain and to try to develop abstract models of these computational processes. These abstract models, albeit inaccurate in details, seem to capture the essence of the operations performed by the brain.

But the motivation for computational models does not only come from engineering. There are two broad categories of models developed by researchers with a scientific interest in elucidating the brain questions. The category of the model will depend on the approach taken to understanding cognition.

Top-down modelling of abstract cognition is a materialisation tool for cognitive psychologists developing architectural theories of the mind.

- The *top-down approach* to the brain-mind conundrum makes use of more abstract models, qualitative at the low-level but possibly quantitative at high-level. Such computational models may be motivated by the need to make the theories they substantiate (usually in the domain of psychology and philosophy of mind) falsifiable. As an example, a cognitive architecture can be a model of a certain theory in psychology about the high-level interactions between the central executive, long-term and working memories, and sensory subsystems, while keeping a loose correspondence between these elements and cortical and sub-cortical structures. The description of the theory, in its literary form, could be considered quite unscientific as it may not make precise, testable predictions that allow one to refute it. Yet, the computable implementation of such a system might be designed to produce testable quantitative behavioural predictions such as reaction times, accuracy scores, recall rates, or other typical measure of experimental psychology.
- The *bottom-up approach* to understanding cognition is typically undertaken when in possession of descriptive, rather low-level anatomical data about the nervous system. Often, this approach consists in modelling neural networks with sufficient detail to

include the anatomical or biochemical features of interest.¹⁰ It results in the detailed modelling (more so than with the top-down approach) of a specific part and mechanism of the nervous system, on a limited timescale. This task is motivated by the intrinsic complexity of the modelled subsystem. Contrary to abstract generalist top-down computational architectures, bottom-up models often aim at understanding a limited set of biophysical features, that the complexity of the model prevent one to understand without numerical simulation. The resulting quantitative model of low-level mechanisms is often greatly scaled down, for instance modelling the interaction of a few hundred of neurons, in stark contrast with the top-down models of large sub-systems.

Bottom-up modelling of neural mechanisms lets neuroscientists observe emergent phenomena, and create further tests of their hypotheses.

In both cases, the computational simulation provides the modeller with the possibility to make verifiable predictions about the behaviour of the system modelled. This is a very important point, as it allows experimental validation or falsification, followed by another cycle of modelling and prediction, in a virtuous spiral of scientific improvement.

Finally, computational models serve a very useful illustrative role, useful not only in pedagogical contexts, but also for researchers to understand the very intricate behaviour of cortical neural networks. This motivation for computational modelling is important, as understanding the emergent phenomena of the (usually mostly unobservable in vivo) system is a very crucial step in developing a theory of its functioning. Computational models let theorists experiment with emergent phenomena and highly complex and unintuitive information flows in simulated neural networks, letting the experimenter's intuitive insight develop.

Computational models have an important illustrative role.

1.5 DYNAMICAL COMPUTATIONAL MODELS OF BRAIN MECHANISMS

Computational cognitive modelling has been steadily active since the early days of computing and the advent of research in artificial intelligence (AI), with pioneer works before the turn of the XXth century. Early computing resources were in part spent in understanding the essence and mechanisms of human cognition, in its rational and irrational aspects. From the start of mathematical and computational modelling, research on actual mechanisms of cognition progressed alongside with abstract research in AI (see for instance [23] for founding cognitive modelling research by McCulloch & Pitts). The initial works of Anderson, who created with Bower a computable model of *Human Associative Memory* [25], planted the seed of the influential and

Early works on symbolic architecture and AI

¹⁰ The features of interest for the modeller are, for instance, those with a hypothesised influence on network dynamics at a given timescale.

comprehensive *ACT-R* theory of thought. Within this emerging field of cognitive science, different approaches to understanding cognition at various levels were used. In the early days of computational cognitive science, the majority of works on cognitive models was focused on high-level symbolic production systems. This was the natural result of the incorporation of the symbolic baggage of classical AI into psychology.

Connectionism

However, in the 1980s, the resurgence of neural networks naturally brought *connectionism*¹¹ to cognitive modelling. The prototypical example is the Parallel and Distributed Processing (PDP) framework [26]. Nowadays, the same computational foundations are at the basis of all connectionist models of cognitive processes. Hence, current knowledge in the field is in part inherited from several decades of research in connectionism.¹² But more than an application of connectionist AI, computational modelling of human neuro-cognitive functions has been the source of fundamental advances in connectionist computational methods.

The balance between detailed and abstract

At the other end of the research landscape, purely conceptual models of cognition, have the propensity to remain blur in the detailed mechanisms they presuppose, due to their non-computational (verbal) nature. It can prevent latent problems from being discovered until a fully-specified computational or otherwise mathematical model allowing predictions and comparisons with human cognition is developed.¹³ In more general terms, the more detailed and specified a model of a set of cognitive processes, the more refutable and improvable the modelled cognitive theory. Refutability is a necessary attribute of scientific theories. Consequently, researchers building a rational model of cognition ought to model mechanisms only if biologically justified. This ensures that hypothesised functions or interactions do not contradict reality as observed by neuroscientists. Crick [29] underlined this danger of connectionist modelling, in that blind computational explorations of the mind are as uninformative for the understanding of its mechanisms as purely psychological approaches and models. It is essential for any computational model of brain functions to be entirely grounded in current knowledge in neuroscience.

The choice of a connectionist implementation of a cognitive model depends on the level of the model. Naturally, a high-level production system such as the one of Adaptive Control of Thought – Rational (ACT-R) will be prone to a symbolic implementation, while a model of neuronal interaction will be naturally connectionist. Nev-

¹¹ Connectionism is a computing paradigm relying on the computations and interactions of simple individual computational units modelled after the abstract properties of neurons or groups of neurons.

¹² See [27] for a historical review.

¹³ Refer to [28, page 7] for a summary of the arguments against purely conceptual models of cognition.

ertheless, not all are using the connectionist paradigm as a way of representing the interaction of physical brain structures. In fact, psychologists have been using symbolic connectionist models for their computational power without any intent at forming a biological interpretation of the networks used, for instance in cognitive linguistics. However, the first advantage of these symbolic connectionist models is usually to permit quantitative empirical justification thanks to their computational nature. In the second place, it has been argued, for instance by Hintzman [30] that a computational simulation allows for serendipitous discoveries in artificial data. These discoveries, if they are confirmed by clinical trials, may help the understanding of real human cognition. Third, the added advantage of computational models running as simulations is their explanatory power. The complexity of the processes taking place in the brain make it hard to understand the actual effect various elements of the process modelled have on the behaviour of the system studied. Furthermore, the dynamic aspects of cognition are very difficult to grasp from a paper description of its speculated mechanisms. Dynamics are much more easily revealed by simulations where state information is fully available and every aspect of the simulation is controllable, than by clinical observation, as current imagery methods do not allow for high combined spatial and temporal resolution.

The power of simulation in connectionist paradigms

Creating hybrid connectionist-symbolic models of cognition has the advantage of higher expressiveness that can allow for modelling the interaction between several levels of cognition, from the neural substrate level to the level of the cognitive architecture (see [31, 32] for example). However, the added complexity makes that few such integration works have been successfully attempted.

These levels of cognition as typically studied in cognitive science can be classified according to the scale of the phenomenon or object they aim at understanding. Dyan [33] summarised how different types of models at the neural level will respond to different needs for abstraction, since a model rests on reductionist assumptions (themselves considered as models) regarding the underlying details that are its building components. More broadly, Sun et al. [34] argue that cognitive science research needs to emphasise the interplay between levels. In sociology, the link from the highest cognitive model of inter-agent communication (sociological) may be studied in conjunction with the psychological states of individual agents (psychological level). Equivalently, in cognitive science, the “substrate” levels of neuronal computation should be studied in the context of components of the agent’s cognitive architecture. In this light, the recent abundance of modelling work in cognitive neuroscience addresses the missing link between the psychological and the neuronal levels.

The balance between connectionism and symbolism

The study of the interaction between the level of cognition of the model and the cognitive system at higher and lower levels helps to

isolate the model from superfluous details of lower level implementation or higher level mechanisms. By formalising the minimal dependencies on the rest of the system, one builds a more robust and interpretable model, as it no more depends on possibly uncertain or inaccurate knowledge.

MOTIVATION AND CONTRIBUTIONS OF THE THESIS

2.1 INTRODUCTION AND MOTIVATION

2.1.1 Overview of the research approach in Part ii

Computational modelling in neural and cognitive sciences

The computational modelling approach is a relative newcomer to neuroscientific enquiry, but can be a valuable tool to complement the observation techniques used in more traditional sciences of behaviour and cognition. Autism is a pervasive developmental disorder of unknown etiology which study could benefit from computational modelling. Autism affects several aspects of the behaviour, and is defined by a set of core symptoms.¹⁴ Because autism is a difficult scientific problem with important human and societal repercussions, it has been studied using a large spectrum of approaches. Autism was studied both from the neurological perspective, and by forming high-level theories of the autistic mind. At the neurological end of the spectrum, neuroimaging¹⁵ and transcranial magnetic stimulation (TMS)¹⁶ both lack in either temporal or spatial resolution. Hence, detailed observations of neural dynamics sufficiently large in scale to unambiguously solve the problem of autism causation can not be obtained. At the other, more theoretical end, psycho-developmental theories of autism, while intellectually attractive and elegantly designed to fit existing frameworks, are not validated, for want of substantiating clinical observations. The current definitions of autism and autism spectrum disorders (ASD)¹⁷ lie on that higher-level side of behavioural sciences: autism is primarily the concern of psychiatrists and clinical psychologists.

Computational techniques are helpful in the development of theories that try to bridge that gap between data from the direct but limited observation of the dynamics and structure of the brain and studies

Computational modelling has the potential to fill the gap between descriptive neuroscience and more abstract psychology.

---

¹⁴ As a syndrome, the symptoms of autism are guidelines to its diagnosis.

¹⁵ Neuroimaging refers to the imaging of the brain, to observe its structure or its dynamics.

¹⁶ TMS is a technique that uses magnetic fields to affect the polarisation of populations of neurons.

¹⁷ Autism is a syndrome in which affected individuals often show a variety of symptoms that may, to a certain degree, correspond to other, related pervasive developmental disorders. Hence, the degree to which patients match the diagnosis criteria gives rise to a graded spectrum of autistic disorders, called the autism spectrum disorders (ASD).

of the behaviour of individuals. The formulation of neurocomputational models helps the understanding of the emergent phenomenon in these sometimes unobservable dynamical systems. It forces the fleshing out of abstract ideas into potentially testable mechanistic hypotheses. The necessity for a specific implementation of each and every mechanism can be felt as a disadvantage of that computational approach. Indeed, it can come at a great initial cost, while still leaving the modeller dissatisfied with the inevitable performance trade-offs and other simplifications.¹⁸ These hurdles are worth being overcome if they can provide significant additional insights into the said mechanisms. This can happen, for instance, by laying out in simulated numbers the existence of a theoretical parameter threshold beyond which new behaviour emerges. Of course, a detailed mechanistic model at the frontier of knowledge in neuroscience is fated to be more often proven wrong than right, but that is a desirable property. Through such iterative refinement, detailed theories that could explain human cognition and its disorders can be invented.

Summary of the motivation and research approach

With these considerations, the work presented in Part ii aims at being both a formulation of a theoretical computational model of visual attention, and the representation of a theory of autism causation in the context of that model of attention. While neither the mechanistic model of bottom-up attention presented, nor the channelopathy hypothesised to be at the origin of certain cases of autism spectrum disorders (ASD) are novel [35, 36], their simulation and integration into the EVAC model are new.

Part ii of the thesis implements a computational model of the theory of autism calcium channelopathy in the context of reflex visual attention.

This research approach is meant to be iterative on the long term, like research on integrative scientific theories ought to be. The iterative nature of computational studies is also a natural consequence of their prospective value: the output of such studies is to be critically confronted with data. Building on the model of bottom-up visual attention by O'Reilly & Munakata [35], Part ii of the dissertation aims at formulating a preliminary model of how autism calcium channelopathy can lead to the disturbed attentional mechanisms observed in some cases of autism spectrum disorders (ASD) [37–39].

Why modelling autism?

Research on the possible root causes of autism has led to very varied hypotheses and to discoveries on the influence of the mutation of numerous genes on developing the syndrome.

Yet, the mechanisms by which autism could occur have not been well determined. Current research directions involve the immune sys-

Autism is a complex behavioural syndrome; its etiology remains mostly unknown.

¹⁸ In particular, it can be disappointing when the said simplifications are to be later revised in light of new experimental data, for instance.

tem, the digestive system, exposure to various chemicals, and more or less influent environmental factors. Yet, none of the current theories of autism has proven itself right or the others wrong. However, as autism clearly affects the functioning of the brain, all theories have to result in an explanation of how the brain is affected.

Since the discovery and definition of autism in the first half of the XXth century, the theories of its causation have been focused on possible high-level dysfunctions of the mind. There was initially less interest in the actual mechanisms by which such dysfunctions could arise.

Whereas these high-level theories stem from psychological observations, there has been an increasing number of attempts to verify them at the physiological level, using imaging experiments. However, the results are still incoherent, with some experiments validating and others refuting these theories. Although such psychological theories can be used as a guideline for computational explorations of the brain-mind relationship, there is more interest in basing biologically realistic modelling on less abstract medical data.

There has been few computational modelling studies of brain disorders, as the understanding of most of neuro-pathologies would not likely be immediately improved by computational modelling work. The pathologies that could benefit from modelling are those that are holistic and hard to outline; in particular, syndromes, by definition. Autism is very appropriate for this type of computational study because it is an increasingly prevalent syndrome, gaining attention from the general public and the medical community.

Obtaining interpretable emergent models of autistic behaviour can also potentially improve the general understanding of the brain-mind relation. Autism is an affect of the mind at high cognitive level, and understanding its neurological roots could shed some lights on the emergence of cognition.

Computational models of visual attention

To study the possible effects of calcium channelopathy on autism with a computational model, it is necessary to choose to model an aspect of the neural basis of cognition that is limited both in temporal scope and in size. This is first necessary for computational tractability. Second, it is useful to have a more limited model because observed phenomena are likely to be more interpretable. For that, it is also preferable to choose to model a better understood mechanism of the brain.

In Part ii of the dissertation, I choose to model reflex visual attention. Visual attention is conceptually divided in top-down and bottom-up components. The top-down component is, in abstract term, an endogenous bias of the “attentional spotlight”; this bias could be brought about by a visual search task, for instance. In con-

Autism is an interesting target for computational studies.

Good candidate mechanisms for neurocomputational studies take place over short timespans and are well understood.

Visual attention consists of top-down visual attention and reflex visual attention

trast, the bottom-up component is an exogenous reflex mechanism that operates on much shorter timescales. The timeline of the orientation of visual attention is theoretically modelled as the result of the interaction of reflex attention with top-down attention.

I choose to model reflex attention because its short timescale allows for the creation of tractable computational models. Several families of models of visual attention exist in the literature. Upon their review, I settle for an emergent model without central saliency map. Additionally, I base the model's principles on the dual-stream theory of visual processing [40].

In a didactic book about the computational foundations of cognition [1], O'Reilly & Munakata presented a model based on the same principles; the EVAC¹⁹ model that I develop in Part ii extends their work.

Reflex visual attention is a good candidate for computational modelling. I create the EVAC model, extending [1].

2.1.2 Overview of the research approach in Part iii

Simulation in computational neuroscience

A large share of studies in computational neuroscience makes use of generative methods. They seek to use simulation to circumvent the various obstacles to the observation and manipulation of *in vivo* neural tissue. The limited temporal and spatial resolutions of current brain imaging and neural activity recording techniques is a major technical obstacle. By simulating populations or neurons and generating the corresponding electrophysiological data at virtually any resolution and any scale, a modeller is thus able to obtain a better understanding of the emergent phenomena implied by a model. In a domain where specific lesion studies are scarce, a simulation allows to easily manipulate a computational model and make verifiable behavioural predictions. Finally, simulation is a useful tool to help get insights from the inherent complexity of biological neural systems.

Hence, the scientific value of the simulation of biological neural networks is clear. However, creating and correctly implementing a computational model of neural mechanisms is as difficult as it is valuable. Aside from the risk that the design of the model is flawed, there is also a significant risk that an implementation error undermines the simulation.

In this regard, the reproducibility principle of the scientific method is a necessary but insufficiently specified condition to uphold the scientific value of a simulation. Reproducing a simulated experiment by running an existing implementation is indeed insufficient to improve confidence in the underling model, because it does not address software implementation flaws. To address the problem of software

Simulations are valuable, but software may be unreliable.

Software faults affect reproducibility.

¹⁹ Standing for emergent visual attention under channelopathy

flaws, which could otherwise be a major cause of hidden defects, two approaches are possible.

The first approach consists in using a programming language that allows the constructive proof of its mathematical specification. For instance, by developing a simulation using the Coq proof management system [41], a modeller could formally exclude a number of software faults from the implementation. However, this approach is not common, and axiomatic verification of programs remains rare. It often requires learning a new software development paradigm, and only guarantee that the programs developed are faultless in restrictive conditions.²⁰ In that sense, this solution would ideally be complemented by the second approach.

The second approach to improving the confidence in a neurocomputational model consists in comparing multiple independent implementations of the same model. In that approach, the scientific principle of reproducibility is extended with a software-specific principle of *variety of implementations*. Developing independent simulations of the same model allows for their comparison. The result of this comparison is informative because the implementations are independent, and thus possible software faults can also be assumed independent. Following that, obtaining concordant results from two independently implemented simulations is a strong indication that both simulations are faithful implementations of the underlying model.

This second approach is implicitly used in the current computational neuroscience research landscape. A number of *simulators* are being developed independently from each other, to the benefit of the scientific method.

However, models developed for these simulators are specified using formal languages that are often not compatible between simulator implementations. Several initiatives aim at addressing this problem by providing common formalisms to develop neurocomputational models that can be simulated on several simulators with few modifications, if any. Among these initiatives, PyNN is a Python framework that allows programming simulations for several simulators under a common API.

Learning synaptic weights in simulated networks

Yet, the software landscape appears incomplete. While frameworks exist to allow creating the structure of networks and running computational experiments without depending on one specific simulator, there is no software that lets the modeller train such network in a

Approaches to improving the reliability of generative scientific methods

PyNN helps unifying incompatible simulators and languages.

Few simulators offer network learning algorithms, and there is no simulator-independent solution.

²⁰ Proof systems such as Coq can assist in positively proving that a program has certain properties if the whole program is developed within the system. However, no such proof system can handle Turing-complete programs (by virtue of Rice's theorem [42]), a limiting factor that may result in only being able to prove that *part* of a program operates as specified.

simulator-agnostic manner. Network training is a crucial part of modelling, because the synaptic weights between neurons can only be set manually in very simple networks of a few neurons.

PyCogMo, presented in Part iii of the thesis, constitutes an attempt at remedying to that situation. It is a software framework for the development and automated training of simulated biological neural networks in computational neuroscience.

PyCogMo is a learning framework for neural networks built with PyNN. PyNN is a versatile simulator abstraction framework build on Python, a general-purpose programming language with powerful scientific computing libraries (NumPy and SciPy). As a result, it is possible to use PyNN as an abstraction mechanism for the learning of simulated biological neural networks.

PyCogMo is a learning framework designed to address this problem.

In that view, PyCogMo is designed first and foremost with extensibility in mind. It aims at providing the fundamental building blocks that allow modellers to implement their own learning algorithms. Besides its role as a network learning tool, PyCogMo also has real-time network visualisation functionalities. This visualisation environment also primarily aims at being extensible. The modular design of PyCogMo supports this intended role as extensible framework.

2.2 CONTRIBUTIONS OF THE THESIS

2.2.1 *A model of visual attention capture in presence of calcium channelopathy*

In Part ii, I develop a model of reflex visual attention that extends the works of O'Reilly & Munakata [1]. This model of emergent visual attention under channelopathy (EVAC) extends their work in scale and in scope. In scale, EVAC is more complex than the model of O'Reilly & Munakata [1], and in scope, EVAC does not only model emergent attention in a dual-pathway model of vision, but also places this model in the context of the calcium channelopathy theory of autism. Thus, the proposed model contributes to the theoretical study of channelopathies. Consequently, clinical studies of autism and of calcium channelopathies can also potentially benefit from the EVAC model.

This computational model of visual attention is based on existing knowledge of the visual streams, and on a timed visual attention shift task inspired by classical experimental psychology. In general terms, my work contributes to the understanding of cortical dynamics by modelling autism as a channelopathy in such a network. Indeed, the EVAC model, by helping to better understand autism calcium channelopathy, helps to improve the understanding of the high-level functional role in cognition of the low-level chemical mechanisms whose malfunctions may contribute to autism.

The simulation of EVAC, presented in Chapter 5, results in a number of direct contributions to the study of the calcium channelopathy theory of autism. The network is run in two configurations: one simulating channelopathy and one corresponding to control subjects. By comparing their performance and dynamics on a variety of attention shift tasks, I can infer possible mechanisms and qualify the effects of channelopathy on the behaviour. The most important contribution of EVAC to the study of channelopathy is thus the set of qualitative predictions that result from these simulations. Importantly, those predictions can potentially guide further clinical explorations of calcium channelopathy.

Finally, the EVAC model presented in this dissertation could be useful not only for the development of the calcium channelopathy theory of autism, but also to a wider audience of cognitive scientists and information engineers. EVAC has the potential to assist cognitive scientists in trying to understand the brain-mind gap. In engineering, EVAC could be useful to information systems engineers by inspiring implementation ideas to reduce the processing load of computer vision applications, in particular with strong real-time processing requirements.

Contributions of the model of emergent visual attention under channelopathy (EVAC) presented in Part ii

2.2.2 *PyCogMo: a novel learning framework for computational neuroscience*

Like a lot of work in computational neuroscience, the model of EVAC is difficult to reproduce. This is primarily due to the state of incompatibility between neural simulation software. EVAC was implemented using the Emergent simulator because it provides the required neuron models and network weights learning algorithm. Yet, EVAC relies on a specific version of the simulator, as Emergent models can not be used in other simulators, or across versions of Emergent. These obstacles to reproducibility affect the scientific value of the EVAC model.

The Python Cognitive Modelling framework (PyCogMo), presented in Part iii of the thesis, was developed to address these limitations. It is a modular and extendible Python library that enables modellers to implement learning algorithms and use them to train models with several popular and otherwise incompatible simulators. It does so by abstracting and unifying the underlying structure of models. PyCogMo also provides real-time visualisation of network structure and activity. It constitutes a step towards improving the reproducibility of computational experiments across simulators, hence a valuable methodological help to computational neuroscientists.

PyCogMo is a new software framework that provides computational neuroscientists with an environment to use and develop artificial neural network (ANN) learning algorithms to train simulated

Contributions of PyCogMo, presented in Part iii

biological neural networks. It fulfils a clear need in the neural modelling software ecosystem by making learning algorithms available for use with several popular neural simulation programs. Currently, learning algorithms are reimplemented for each simulator as the need arises. This negatively affects progress in neurocomputational modelling, reproducibility of research, and the overall availability and use of each algorithm in neuroscience. Thus, by improving reproducibility, PyCogMo fits in an important gap in computational neuroscience modelling software.

PyCogMo offers a novel convenient modular framework for learning in biological neural networks. It is built as a complementary overlay of PyNN, a simulator-independent language for building neural network models. PyCogMo is built on principles of modularity and extensibility, defining abstractions that isolate its operational implementation from changes in PyNN. Modularity isolates implementation from function. The implementation of each module is decoupled from that of all others, thus a change to the implementation of one module only affects the internals of that same module. Several alternative implementations of one module may also exist without increasing the complexity of the overall system. I followed a test-driven development methodology²¹ combined with short release cycles, allowing me to produce a functional framework in spite of being its sole developer.

PyCogMo is, like PyNN, developed entirely in Python following an independent development approach. PyCogMo provides the modeller with Artificial Neural Networks learning algorithms to set the connectivity weights of their networks, and with the necessary abstractions to easily define their own learning algorithms. Thanks to the abstraction offered by PyNN, learning algorithms can be implemented in PyCogMo without depending a particular simulator.

Hence, the major contribution of PyCogMo is to offer a reliable *common learning framework* for several simulators.²² The learning algorithms developed in PyCogMo aim at producing useful approximations of the patterns of aggregated synaptic weights between neural maps observed in the central nervous system.

The second contribution of PyCogMo lies in an extensible module that allows visualising the structure of networks and their parameters and activity in real-time, concurrently to the running of the network. A sample client 3D-rendering visualisation front-end is implemented using the Visualisation ToolKit (VTK) [47]. The use of VTK lets PyCogMo benefit from the advanced algorithms of this high performance scientific visualisation library.

²¹ The implementation of all core functionalities of PyCogMo is systematically tested to minimise the risk of software fault.

²² NEURON [43], NEST [44], PCSIM [45] and Brian [46] are currently supported. As an added benefit of using PyNN, any simulator that will be supported by PyNN will be compatible with PyCogMo.

Part II

A MODEL OF VISUAL ATTENTION CAPTURE

This part of the dissertation looks at emergent attentional effects of specific calcium channels defects hypothesised as a possible root cause for some symptoms observed in cases of autism and autism spectrum disorders. Studies have indicated that calcium channelopathies may be common occurrences in autistic subjects. I have developed a computational model of visual attention based on existing knowledge of the visual streams, and on a timed visual attention shift task inspired by classical experimental psychology. I expound results that corroborate expectations, and open the path for further experimental studies.

MOTIVATION AND BACKGROUND

3.1 OBJECTIVES AND RESEARCH QUESTIONS

There are general dynamical and structural principles that govern information processing throughout the cortex. These properties of the cortical networks are determined by the localised expression of genes, regulated by the cell differentiation, its environment, and its internals. Hence, cortical network dynamics, which could appear to solely depend on static network structure with weights determined by network properties alone, are in fact also dependent on molecular-level details of cell biology and regulated gene expression. The dynamics of cortical networks are not fully understood. My work contributes to the understanding of cortical dynamics by modelling autism as a channelopathy in such a network.

*Molecular
computation*

Autism is interesting to study because it is a pervasive developmental disorder (PDD) that affects many aspects of cognition and behaviour. The cause of autism remains unknown in most cases, and hypotheses have been put forth that the the disturbed expression of certain genes involved in voltage-gated calcium channels (VGCC)²³ may lead to some of the behavioural deficits observed in Autism Spectrum Disorders.²⁴ In particular, it is proposed that it leads to impaired network transitions between stable states, such as delayed shifts of sensory attention.²⁵ However, the gap between the hypothesised low-level disturbance and such macro-level observable behaviour can not currently be bridged by neuroscience and psychology, because the tools to observe the brain over such a wide range of temporal and spatial scales do not exist. Computational neuroscience can help to conceptualise the hypothesised network effects and determine the plausibility of the proposed lesion mechanisms. As Helen Penn states:

*The gap between
molecular and
behavioural levels in
autism research*

There is little research linking neurochemical abnormalities in autism to structural and functional abnormalities. [49, p. 63]

The computational study presented in this dissertation is an attempt towards filling that gap.

---

²³ VGCC are molecular channels on the membrane of neurons that let calcium ions enter the cell in response to changes in electrical potential between the inside of the neuron and the immediate extracellular environment. They are central actor of molecular computation. Affects of calcium channels are called *calcium channelopathies*. More generally, channelopathies are diseases caused by dysfunctional membrane ion channels, e. g. calcium, sodium, potassium, etc.

²⁴ See Section 3.3 for a more extensive review.

²⁵ Attention is the selection of a stimulus or related group of stimuli, with the relative under-activation or deselection of other stimuli [48].

Research questions

The objectives of this part of the dissertation are to develop a plausible model of sequential visual attention shift²⁶ based on current literature, and to use that model to simulate a proposed causation hypothesis of autism. The questions guiding this research effort could be formulated as follows.

1. Is the sequential, short-term bottom-up visual attention²⁷ hypothesis expounded in [35] and in [50] compatible with the dual-pathways model of visual processing in the primate brain? How might the local, emergent hypothesis of visual attention cooperate with the long-range, hierarchical and specialised processing model of primate vision?
2. Is the calcium channelopathy-based theory of autism causation presented in Section 3.3.1 incompatible with this model of sequential bottom-up attention?

*Proposed
computational model
(EVAC)*

*Calcium
channelopathy*

The negative formulation of the second research question is meant to insist on the illustrative aspect of the computational model developed. Regardless of its outcome, a computational analysis alone can in no circumstance *verify* a currently undecided hypothesis in neural science. However, it can help understand a hidden mechanism that appears as the result of the interaction of units, and determine its plausibility within a theoretical framework (*validation*) when the complexity of the system does not allow mathematical derivation. Hence, the model of emergent visual attention under channelopathy (EVAC) developed in this part of the thesis is meant as a computational exploration of the implemented theories, validating their plausibility, and illustrating possible emergent processes. To actually verify the theories (listed in Section 6.2) embodied in the model, it is necessary to compare the predictions (summarised in Section 6.3) made by running the computational simulation to corresponding real-world tasks (proposed in Section 6.6.1). The modelled electrophysiological response of calcium channelopathic neurons is an educated guess based on current knowledge of the properties of affected channels.²⁸ The proposed EVAC model introduces that projected electrophysiological behaviour of calcium channelopathic neurons in a simulated network which embodies a unit accommodation-based theory of short-term reflex visual attention. Accommodation-based reflex attention is an emergent property of competitively inhibited cortical maps whereby

²⁶ The direction of attention to a new stimulus in the visual space.

²⁷ Bottom-up visual attention processes inputs from the retina without selective influence from higher cortical areas. In contrast, top-down attention designates the selection of the attended signal (object or spatial location) with a significant influence from higher cortical areas – often prefrontal, for instance during an image search task. As bottom-up attention is a subconscious phenomenon with comparatively more limited neural circuitry involved (a constraint of its short duration), it is also called reflex attention.

²⁸ See Section 3.3.1 for a review.

neuronal accommodation, a *neuronal fatigue* that occurs in the form of a tendency to hyperpolarise after a short amount of sustained stimulation²⁹ leads to the change of attended location. It was integrated into a model of reflex visual attention by O'Reilly & Munakata [35]. As the projected electrophysiological behaviour of calcium channelopathic neurons consists in higher overall activation rates resulting in earlier, longer hysteresis, and delayed and lengthened accommodation, (see Section 3.3.1), it is reasonable to expect accommodation-based attention to be affected.

In the EVAC model of the effects of calcium channelopathy in reflex visual attention, I keep the principles of visual bottom-up attention modelling introduced by O'Reilly & Munakata [35]. Following these principles, EVAC implements the two-streams theory of visual attention³⁰ [51], with a common visual stream running from a LGN/retina compound input layer to a simulation of the second visual area (V₂). From that map, the dorsal stream comprises of two further cortical layers, and the ventral stream, of three. Still following the principles of reflex visual attention modelling of O'Reilly & Munakata [35], the simulation makes use of phenomenological models of accommodation and hysteresis³¹ in the enhanced Hodgkin-Huxley models of neurons³² described in [1]. In addition, I follow basic structural properties of the cortex: neural connectivity respects the general organisation in terms of layer arrangement and lateral intra-map connectivity (see Sections 4.3.4 and 4.3.4), as well as the general principle of layered cortical maps with receptive field connectivity and systematic feedback (recurrent network).

In the rest of this chapter, I review related literature and value the potential contributions of my research relative to the current scientific, clinical and technological contexts. I detail the model of emergent visual attention under channelopathy (EVAC) and the tasks implementations in Chapter 4. In Chapter 5, I describe network performance during learning, and expose and compare the results of these tasks at unit and network levels, in absence and presence of the modelled channelopathy. Chapter 6 discusses these results, answers the

*Principles of
emergent bottom-up
visual attention*

Overview of Part ii

²⁹ Accommodation is not to be confused with the refractory period that follows an action potential; the two operate on different timescales. Accommodation is more similar to a very short depression; it is a statistical phenomenon triggered after numerous action potential have been fired. In contrast, the action potential's refractory period is a systematic discrete feature of the atomic action potential.

³⁰ The *two-streams hypothesis* is an influential model of neural visual processing in which two distinct subsystems (streams) process visual information along different dimensions. The *ventral stream* is involved in object recognition, while the *dorsal stream* is responsible for the spatial location.

³¹ While accommodation is a short-term neuronal fatigue, hysteresis is an even shorter-term hyperexcitability of neurons in response to continued input.

³² Hodgkin-Huxley models are conductance-based circuit equations describing the initiation and propagation of action potentials in neurons. See Appendix A.

above research questions, and highlights possible developments of the research.

3.2 AUTISM

Prevalence of autism

Autism is a rather common early developmental syndrome, with prevalence figures of the order of 2% of the general population affected [20]. It involves a large, varied, and ill-defined set of symptoms. No consensus has been reached on which combinations of phenotypical symptoms or genotypical abnormalities are specific to autism. Hence, a multitude of definitions of autism as an illness (leading to a different set of core symptoms) has emerged, and the borderline between several PDD has been blurred, if ever defined. The most commonly accepted behavioural symptoms and mental characteristics of autistic people appear during the first few years of their life. They usually consist in difficulties with oral communication, resulting in fewer attempts towards communication with the caregiver. During later development and adult life, the most frequent behavioural symptoms of autism, collectively called *autistic behaviour*, are difficulties with symbolic thought, restricted and repetitive behaviour, associated with or resulting in low interaction with the social environment [52, 53].

Symptoms of autism

3.2.1 *Neuroscience of autism*

Genetic factors

Research on the possible root causes of autism has led to discoveries on the influence of the mutation of numerous genes on the probability to develop autism. Indeed, autism has proven to be highly heritable [53]. It can also be simply shown not to occur from a Mendelian mutation.³³ It is likely to occur as the result of the mutation of multiple genes. This likelihood of several genes being responsible for autism has an interesting implication: the prevalence of bearers of the incriminated alleles³⁴ may be higher than suspected, but the expression of a particular combination may result in high-functioning individuals or in a syndrome that is different from core autism. The concerned genes are encoding proteins related to neural development and the immune system, but these genetic defects seem not to account for all cases of autism. The International Autism Genome Project – a recent research collaboration³⁵ – even seems to indicate that the risk of developing or inheriting autism or ASD increases with the mutation of any of several hundreds loci³⁶ all over the human genome.

³³ A Mendelian mutation is the mutation of one gene.

³⁴ An allele is one among several possible versions of a gene.

³⁵ Read more about the International Autism Genome Project on <http://autismgenome.org/>

³⁶ A locus is a position on a chromosome.

The mechanisms by which autism could occur have not been well determined. Current research paths involve the immune system [54], the digestive system [55], exposure to various chemicals and more or less influent environmental factors [56]; and none of the current theories of autism has proven itself right or the others wrong. However, as autism clearly affects the functioning of the brain, all theories have to result in an explanation of how the brain is affected.

Environmental factors

Neuro-anatomical growth is disturbed as early as during gestation, affecting the development of the whole brain, from the brain stem to the cortex. Thus, it is very likely that there are causal relations between some of the aforementioned hypothetic mechanisms of autism and the disturbed physiology of the autistic brain. In particular, the expression of some corrupt autism-related genes is widely believed [53] to be a basis of most cases of autism. I propose that the development of targeted computational models of biological neural networks could advance the understanding of the influence of low-level mechanisms and gene expression on higher-level functioning of the brain. Such targeted computational models would help to verify the influence of some candidate environmental and genetic factor on the cognition of autistic persons. Additionally, linking gene expression and the influence of suspected chemicals with brain development and neurons behaviour could also be of interest to finding therapies to autism.

Anatomy and physiology of the autistic brain

Of course, the proposed model of channelopathy as a possible factor of autism only covers one aspect of the neurobiology of the syndrome. Studies in neurology have revealed a number of other structural and developmental factors that may have an influence on the autistic behaviour. They are empirical observations that are beyond the scope of EVAC, but they are helpful to contextualise my contributions. The following will present these possible etiologies of autism.³⁷

3.2.2 *Structural and developmental abnormalities*

Some structural and developmental abnormalities have been statistically linked to autism. These include cerebrum overgrowth, disturbed cortical minicolumn structure, and problems in the cerebellum.

At the macro level, a general brain overgrowth is often noted, with macroencephaly between age 2 and 12 [57, 58]. It is likely that this structural change reflects large-scale disturbances in the evolution of the distribution and density of neurons. Research indicates that an increased neuronal growth or a decreased neuronal pruning³⁸ underlie this overgrowth [59, 60].

Brain overgrowth

³⁷ For a more in-depth review of the current state of the research in neurobiology of autism, please refer to [49].

³⁸ In other words, too many neurons are created compared to the amount being naturally pruned during development.

*Disturbed cortical
minicolumn
structure*

Cell migration errors [61] and disturbed cortical minicolumn structure [62] have also been detected. Cortical minicolumns are a fundamental element in the microstructure of the cortex. They consist in the physical grouping of a few hundreds neurons that share their receptive fields.^{39 40}

Recent findings of abnormal levels of several brain growth-related proteins corroborate the observation of brain overgrowth and abnormal minicolumn structure:

- First, Persico et al. [64], Skaar et al. [65], and Serajee et al. [66] found a link between problems with the expression of a gene regulating neuronal migration and autism.
- Second, Fatemi et al. [67–69] found a similar link with a protein regulating the inhibition of apoptosis, a form of programmed cell death.
- Third, various aspects of neural synapses development regulation are examined in [59, 70], along with other proteins potentially linked to secondary symptoms of autism. The same publication also examines the involvement in autism of genes linked to programmed cell death prevention and growth regulation.

One may refer to [49] for a more comprehensive discussion about dysregulated neuronal growth proteins and neuropeptides in autistic individuals.

Cerebellar problems

It is to be noted that there also are developmental anomalies in non-cortical regions of the brain associated with autism. Prominently, the microstructure of the cerebellum is often affected in autistic individuals. First, the loss of Purkinje cells⁴¹ is typical, supporting the argument that autism has a consistent prenatal component, as this anomaly is strongly suspected to happen prior to the 30th week of gestation [71]. Second, other studies of the cerebellum of autistic individuals indicate the loss of granular cells, and suggest that the cell loss may be preceded by an enlargement of the neurons during childhood [72]. It is also worth noting that parts of the cerebellum⁴² show under-development in four fifth of cases, and hyperplasia in the remaining cases [73]. More cerebellar disturbances have been noted and linked with mental deficits in autism [49].

Overall, the scientific study of the possible genetic and neurological causes of autism does not lag by lack of evidence, but by its abund-

³⁹ See 3.3.2 for an explanation of the role of receptive fields in EVAC.

⁴⁰ The dynamic effects of such structural abnormalities are possible to computationally model. The Blue Brain project [63] is an extensive attempt to perform the computational modelling of a neocortical minicolumn. Recent news and results from the Blue Brain project are available on bluebrain.epfl.ch

⁴¹ Purkinje cells are very large neurons with a unique and central role in the microcircuitry of the cerebellum.

⁴² These parts are the lobules of the cerebellar vermis.

ance and complexity. On the other side of the neurocognitive spectrum, psychology and related disciplines have developed high-level cognitive theories of autism that may prove useful and complement neurological observations.

3.2.3 *Complementing psychological theories*

Since the discovery and definition of autism in the first half of the XXth century by Hans Asperger and Leo Kanner,⁴³ the theories of its causation have been focused on possible high-level dysfunctions of the mind, without as much interest in the actual mechanisms by which such dysfunctions could arise. The *weak central coherence* [75], and the *executive dysfunction* [76] theories of autism are prominently cited, along with the *mind-blindness* [77] theory. These theories are related and not completely mutually exclusive, as expounded by Frith [78]. *Weak central coherence* is a phrase describing the inability to gather separate pieces of information into a more global and possibly more abstract thought. The term “executive dysfunction” denotes the observable behavioural consequences of possible frontal lobe dysfunctions in autism. Finally, the lack of a theory of mind in autism refers to the difficulties autistic patients have in empathic thinking, in internalizing other people’s state of mind. These high-level theories stem from psychological observations, and attempts are being made to verify them at the physiological level, using imagery experiments. The results in all cases are incoherent, with some experiments validating and others refuting these theories.

*Psychological
theories of autism*

These mind-level theories based solely on psychological observations may not carry as much explanatory power as etiological theories at the neuroanatomical level. Contrary to what is expected of a theory in hard scientific fields, such conceptual cognitive scientific theories of the mind may suffer from a lack of detailed biological explanatory mechanisms to be tested, proven or disproved. This state is attributable to the rooting of these theories in psychology; in fact, these theories are not meant to explain the brain dysfunctions, but to describe how the autistic mind functions. Although psychological theories can be useful as guidelines for computational explorations of the brain-mind relationship, there is more scientific interest in basing biologically realistic modelling on low-level medical data.

*Limitations of
high-level theories*

It is important to underline that any validation of a theory of autism does not necessarily invalidate other possibilities. It may either account for a fraction of all cases, or may take a partial role in the expression of the syndrome,⁴⁴ or may only be a partial determinant of the final autistic phenotype. As they are specified based on a continuous scale of behavioural symptoms, autism and ASD may be viewed

⁴³ See [74] for a historical review.

⁴⁴ Population studies often reveal numerous potential factors of autism.

*Autism may have
several possible
etiologies.*

as the same syndrome expressed at different degrees of severity, or could be assumed to be different disorders with some common symptoms. This debated point of nomenclature has led to studies being conducted under both assumption (see for instance [79]). To avoid confusion and keep a coherent picture, studies targeting ASD rather than core autism only are systematically mentioned as such in the following literature review. The same caution has to be given to the unknown origin of autism: I also carefully keep in mind that autism may be symptomatic of several completely different genetic disorders with distinct environmental triggers.

The weak central coherence theory of autism can be viewed as an overarching principle or guide of my computational study, as it is related to unbalanced neurodynamics, in a similar manner as the autism calcium channelopathy theory that I am modelling.

3.3 CALCIUM CHANNELOPATHIES AND VISUAL ATTENTION IN AUTISM

This section introduces and connects studies that underlie my work on the preliminary synthetic model of the emergent effect of calcium channelopathies linked to autism on the dynamics of short-term visual attention (EVAC).

There are two important mechanisms that need to be reviewed. First, at the unit level, calcium channelopathy is assumed by the EVAC model to have behavioural consequences in autism. To justify this, I discuss current literature on calcium homeostasis and signalling in autism in the next section. Second, the reflex visual attention processes are implemented both structurally, as the networks replicates visual streams, and at neuronal level, using unit accommodation as the main driver of reflex attention shift. Subsequent sections justify both choices: Section 3.3.2 discusses visual pathways in the brain, and Section 3.3.3 explores current theories and models of visual attention.

3.3.1 *Calcium homeostasis and signalling in autism*

*The genetic link
between calcium
channels and autism*

Autism and ASD have highly complex and heterogeneous genetics. Several recent genome-wide association studies (GWAS) have linked autism and a group of genes related to the functioning of some of the calcium channels ubiquitously found on the neuron membrane⁴⁵ [80]. The finding of calcium channel disturbances corroborates some primary and secondary symptoms of autism.

⁴⁵ These calcium channels have fundamental roles in the response of neurons to inputs. For detailed information on the basic electrophysiology of neurons and the roles and models of particular ion channels, please refer to Appendix A. The exact model of ion channel dynamics used in my study, described in Section 4.3.3, is based off these foundations.

First, there is evidence that disturbed immune activity during neurodevelopment may be part of the causal factors of autism, as the symptoms of autism have repeatedly led researchers to believe in a poorly regulated immune response of autistic children [54, 81]. Such dysregulated immunity could impair the development of the neural system, as the latter depends on a balanced immune response [54]. Furthermore, the maintenance of calcium homeostasis is crucially important in the functioning of immune cells,⁴⁶ and recent studies have revealed evidence of altered Ca^{2+} homeostasis in ASD [86, 87]. Together, these studies point towards a causal link from channel-mediated intracellular calcium imbalance in autism and the disturbed neurodevelopment of autistic subjects.

The immunity link

Second, there is direct evidence of both offset calcium homeostasis and disturbed calcium signalling pathways in the autistic brain. Dysregulated calcium homeostasis has been observed at the macro level by Palmieri et al. [86]. This finding of dysregulated homeostasis was later linked with calcium signalling defects by Napolioni et al. [87]. It is known that voltage-gated calcium channels are essential to neuronal maturation and differentiation, in addition to their central function in neural information processing [88]. The body of evidence on these calcium signalling pathways anomalies in the neurons of autistic subjects consists primarily in a growing corpus of genetic studies. Indeed, the genetic evidence linking autism and calcium channelopathies is clear, as exposed by Gargus [89]. However, the exact nature of that link is not yet understood. Among recent works, [80] uses GWAS data to demonstrate that the calcium channel genes contributing to neuronal function have a definitive role in some cases of ASD.

Evidence of disturbed calcium homeostasis

The model of emergent visual attention under channelopathy (EVAC) expounded in this part of the dissertation focuses on these potentially disturbed transmembrane calcium channels. One type of calcium channel, called voltage-gated calcium ion (Ca^{2+}) channel (VGCC), is of particular interest in neuronal dynamics. VGCC are very important actors in action potential (AP) propagation, in addition to being involved in muscular contraction, and endocrine and hormonal releases [90]. Their role in AP generation and propagation is cyclical, as it consists in further depolarisation of the membrane in response to an initial decrease of electrical potential.⁴⁷ The modelled calcium channelopathy, which results in increased overall neural activation

⁴⁶ The importance of calcium homeostasis in the immune response is evidenced by the cytopathic effects of the Ca^{2+} homeostatic imbalance triggered by several viral infections, see for instance [82–84]. Misra et al. [85] also showed that beryllium toxicity is in part the result of altered Ca^{2+} metabolism in mononuclear phagocytes consequent to reversible opening of plasma membrane channels, which not only reveals the central role of calcium homeostasis in the immune system, but also that of membrane calcium channels in that process.

⁴⁷ See Appendix A for more explanations.

*L-type voltage-gated
calcium channels
and autism*

but longer accommodation, is hypothesised to model the effect on neural activity of the VGCC anomalies comorbid with autism. Specifically, my working hypothesis of channelopathy is primarily justified by the recent linkage of defects in a subtype of VGCC called L-type voltage-gated calcium channel (LTCC) with a well-known channelopathy called the Timothy syndrome. The Timothy syndrome is notable for being one of the rare monogenic channelopathy paradigms, and furthermore presents core autism as a major symptom [89]. Notably, the Timothy syndrome is a condition proven to be caused by the mutation of a gene (*CACNA1C*) encoding a sub-unit of LTCC [91], and as such is proven to be the root cause of an autism-causing voltage-gated calcium channelopathy. Furthermore, LTCC are of particular importance in a number of the biological systems and processes whose disruptions produce significant symptoms of autism. With respect to the disturbed neurodevelopment and more frequent immune and gut problems of autistic children, this hypothesis that calcium channelopathy is an important component of autism has led to the formulation of a new and interesting partial theory of the etiology of the syndrome [92].⁴⁸

3.3.2 *Visual pathways in the brain*

The previous section gave an overview of the latest discoveries and theories of autism that concern the model at the level of individual neurons (calcium channelopathies). This section discusses current knowledge and theories of visual pathways and reflex visual attention, upon which the structure of the neural network of EVAC is designed.

*Criteria for the
choice of modelled
network*

The hypothesis that channelopathic neurons lead to cortical networks with dynamic properties that are symptomatic of ASD, or at least compatible with ASD, only constrains the properties of the neuron model, and not of the network. The choice of which cortical network to model remains up to me, as modeller. It is reasonable to choose to model a subsystem for which existing computational models are known to replicate the neural implementation with reasonable fidelity. Choosing a neural subsystem that is sufficiently well-understood is advantageous to build a simplified model with some understanding of the consequences of the simplifications made. In my case, I also want to have easily understood and experimentally accessible correspondences between the inputs and outputs of the model of emergent visual attention under channelopathy (EVAC) and those observable in a similar psychological experiment on human subjects.

⁴⁸ Lozac's calcium channelopathy theory proposes dysfunctional calcium homeostasis and disturbed functioning of *chemokine receptors* and VGCC during development as an etiology of autism. Chemokine receptors are agents in the migration of neurons during development. This theory is coherent with recent studies [93] on Ca^{2+} signalling pathways of autistic individuals.

To directly map model inputs to non-invasive experimental stimuli, the input layer of the model must correspond to a sensory area. Similarly, if one is to easily interpret the model response in terms of externally measurable behaviour, the output of EVAC should readily translate into expected muscular, physiological, or event-related potential (ERP)⁴⁹ recordings.

Considering the above factors, I found that the reflex orientation of visual attention⁵⁰ seems an appropriate mechanism to model. Reflex attention designates the involuntary and fast focus of the abstract “attentional spotlight”⁵¹ towards a stimulus. In the case of visual reflex attention, the focus is physiologically measurable by the orientation of the gaze, which occurs either by the saccadic rotation of the eyeballs alone or by the rotation of the head together with the compensated eye saccade. Hence, the output of a model of visual attention should translate into a physically measurable event. Input to the model is also sensory and well-understood. Furthermore, the mammalian visual systems are among the least misunderstood central nervous subsystems [94], and numerous computational models exist that accurately predict properties of the human visual system. Finally, several studies have revealed abnormal reflex orientation of visual attention in ASD subjects (see for instance [38, 39, 95–97]). Altogether, these properties make the study of the reflex orientation of attention in the visual system a prime candidate for the projection of the effects of calcium channelopathy on behaviour. In addition, there is a potential for generalisation of some results to other modalities. First, the cortical networks are rather uniform in their microstructure, as was expounded earlier. Second, primary sensory cortices share organisational features. So results obtained with a model of low-level visual attention may guide the explorations of aural, tactile or other modalities.

The next subsection will present the structure and principles of the human visual system, which are reflected in the EVAC model.

Overview of the visual system

In the human brain, visual information from one side of the visual field is processed by the contralateral hemisphere. Schematically, data are transmitted to successive cortical maps,⁵² in a serial flow with modulatory recurrent connections. The processing of information in

Reflex visual attention is straightforward to measure.

⁴⁹ Event-related potentials are EEG-recordable correlates of motor or cognitive events.

⁵⁰ Throughout this part of the dissertation, reflex visual attention is also called bottom-up visual attention, visual attention capture, or, where there is no possible ambiguity, visual attention.

⁵¹ The term “attentional spotlight” is mostly used to illustrate the selectivity of attention. It may not be a good illustration of the neurological processes underlying attention.

⁵² As a reminder, cortical map is the term used to name a sheet of cortical neurons with similar functions.

one map is a parallel process that follows the principles exposed in Chapter 4.

Receptive fields

The concept of *receptive field* of a neuron on a cortical map is central to the whole chain of visual processing. For a given neuron i receiving an input projection from an input map M , a receptive field of i on M is a set of topographically adjacent neurons of M that project their outputs on i . Figure 3.1 illustrates how each unit receives its input from a patch of topographically adjacent units of the preceding map. This organisation results in an increasingly broader indirect coverage of the visual field by units farther along the processing chain. In Figure 3.1, units are grouped into minicolumns represented by the background light grey circles, and sections of the input and post-synaptic maps with adjacent minicolumns are respectively denoted by A and B . Neuron j_1 in neural map B receives inputs from several adjacent units of the preceding layer A . Its neighbour, neuron j_2 , is part of the same cortical minicolumn, and as such receives inputs from a similar set of neurons in A . All units in the minicolumn j have the same receptive field, and units from adjacent minicolumns share a portion of the receptive fields of their neighbours. For instance, the receptive field of unit k_1 partially overlaps with that of units j_1 and j_2 .

The visual system as an information processing pipeline

On top of these general principles, the successive neural maps that form visual system are arranged in a specific processing “pipeline” that starts with the dual inputs of the retinas and can be argued to end in two of a number of higher cortical areas. The selection of two regions to be considered as the last members of the visual pathway is mostly an arbitrary decision that will depend on the phenomenon studied.

Organisation of the visual system

Figure 3.2 illustrates the organisation of the visual system. In this diagram, the eyes are in their approximate location, under and slightly in front of the prefrontal cortex. The signal first propagates from the left of the diagram to the right, that is from the retinas in the eyes, to the extrastriate cortices (V_1 and V_2), following the path of the optic nerves. These nerves start at the retina and first transmit visual information to the optic chiasm, in which the nerve bundles are rearranged by hemisphere.⁵³ Following that, the signals reach the lateral geniculate nucleus (LGN), a sub-cortical structure located in the thalamus. The LGN processes the signal to extract some microfeatures in both spatial and temporal domains. In the spatial domain, the processing performed by the LGN is a continuation of that done by the retinas: it extracts local differentials from its receptive fields, varying in scale. The processed signal is then transmitted to the primary visual cortex (V_1), which performs further microfeatures extraction from several components of the visual signal using its receptive fields on the LGN. The same principles underlie the functioning of the secondary

⁵³ This *decussation* of the optic nerves does not affect the information processing in EVAC.

Figure 3.1: Diagram illustrating the organisation of receptive fields (RF) in topographical cortical maps

visual cortex (V_2), which takes its inputs from the first visual cortex (V_1), extracting slightly more complex features based on the output of primary visual cortex (V_1) units. Thus, as the signal progresses into increasingly deeper neural maps, it is transformed and activates units representing increasingly complex features of the signal. However, this visual pathway diverges into two separated streams after passing through area secondary visual cortex (V_2) [40], leading up to the splitting of the processing pathways into dorsal and ventral streams, as shown in Figure 3.2. The dorsal stream runs from V_2 to the middle temporal cortex (V_5 or MT) and ending in the posterior parietal cortex (PPT). The ventral stream goes to third extrastriate area in the ventral visual stream (V_4) and to the inferior temporal cortex (IT). As represented in the figure, the ventral visual stream is considered to end in the IT, and the dorsal stream, in the PPT [40]. The EVAC model follows that convention.

Figure 3.2: Visual pathways in the human brain

*The where stream
and the what stream*

Overall, there is evidence that the dorsal stream units are more sensitive to location information (position and motion in the visual space), and less to object identity than units of the ventral stream [40]. In contrast, as one goes deeper down the stream, the ventral stream units are shown to activate in response to the presence of geo-

metric features of increasing complexity and with decreasing spatial location sensitivity. This *where-what duality* makes these two streams complementary [40]. It is also likely that the clear separation of pathways represented in Figure 3.2 is not a reality but a schematisation, as sub-cortical and cortical structures may interact in ways that are not shown in this model, although to a smaller extent [98]. The areas that are later in the processing pathway represent many complex features of the transformed input signal. However, in its output layers, the EVAC model of reflex visual attention only represents the spatial location of the stimulus (dorsal pathway output) and its identity (ventral pathway output).

Each neural map of the visual system contains neurons that are each sensitive to one feature, but covering altogether the entire visual field with a wide variety of features. Most features are not absolute, but rather consist in local spatial or temporal differentials. For instance, a V_1 neuron can be a filter of either oriented contrast edges, oriented colour edges or motion direction, and all these possible features are parameterised by the frequency (spatial or temporal frequency), orientation, polarity in the relevant cases, and ocular dominance (the eye from which the signal comes). It only get more complex in higher areas.

Neurons are feature detectors

In EVAC, I only tune V_1 neurons to the contrast orientation feature at a specific spatial frequency, because the input samples are unmoving black-and-white line drawings.⁵⁴

The structure described above is the basis of all of the visual information processing happening in the brain. Early sensory transforms make direct use of this structure, while the more complex and often multimodal processing that happens deeper in the cortical networks relies on these early sensory networks, and often feeds back to them. Bottom-up visual attention is no exception, and as an early visual process, it necessarily takes place in the early visual maps discussed. But albeit reflex visual attention is based on the neural substrate of the visual system described above, its exact mechanisms are less well-understood. The next section summarises current knowledge about the biological basis of reflex visual attention and describes current models.

3.3.3 Bottom-up visual attention

Visual attention is a process defined in cognitive psychological terms. It is the *cognitive selection* of an entity in the visual space. The subject may be aware of this selection, a phenomenon called conscious attention, or not, in unconscious attention. Attention may lead to a redirection of the eyesight in overt attention, or only have neurodynamical

Types of attention

⁵⁴ To read more about the implementation of receptive fields in EVAC, please refer to Section 4.3.4 page 94.

consequences, a process called covert attention. Finally, the orienting itself:

- may seem initiated by the self, which may be called voluntary, endogenous or top-down attention;
- or it may be a reflex-like process driven by the appearance of a stimulus in the visual field, a mechanism known as involuntary, exogenous, reflex or bottom-up attention [99].

Through all these processes, attention can be oriented from one target to another, a process named *attention shift*.

*Posner's phases of
attention shift*

In the 1980s, studies of spatial attention by Posner et al. (see for instance [100]) were influential in describing these different kinds of attention. Posner et al. [100] introduced the three phases of attention shift to a new target as: *disengagement of attention* from its current focus, *moving attention* to the target, and *engagement* of the target. This study linked parietal lobe injury to difficulties to disengage attention.

In [101], Knudsen consolidated the psychological model of attention into a 4-tier framework comprising of working memory, top-down sensitivity control, competitive selection, and automatic bottom-up filtering for salient stimuli. In this framework, the bottom-up filtering mechanism corresponds to the reflex visual attention mechanisms being modelled in my work. Thus, it is necessary that I review models of visual attention in order to choose the most fitting framework for my work on the effects of calcium channelopathy on reflex attention.

*The choice of a
model of visual
attention*

There is no verified model of top-down or bottom-up attention, but several theories co-exist, each having valuable explanatory power over aspects of attention. The choice of model of attention to use in my work is based on arguments about the relative biological plausibility of each of these theories. For that purpose, it is important to note that many theories have been conceived with less attention given to the plausibility of a biological implementation, and more to the absolute performance, or to fit a psychological theory, or to express a mathematical principle that may or may not relate to the neural implementation of attention. For instance, control theoretical approaches to attention have led to more high-level frameworks such as CODAM (Corollary Discharge of Attention Movement) [102]. This high-level engineering approach let the authors extend the architecture to propose a cognitive theory of consciousness [103]. Simulations that stem from this higher-level approach are more fuzzy on the biological details that underlie the cognitive architecture. The papers that describe the framework and simulations do not give details on the functioning of each high-level module, or any way to access their implementation⁵⁵ [103–105]. Nevertheless, the authors claimed quantitat-

⁵⁵ This practice of omitting to fully document and detail the model from which results are published appears common in computational neuroscience. It prevents from re-

ative results that match the results of psychological and brain imaging experiments.

But the distinction between the conceptual models that are less grounded in biology and those that place more emphasis on neural plausibility is not as clear-cut as it may seem from the above. Often, conceptual models describe algorithms, and as such will inspire new hypotheses mapping the abstract components of the algorithm to physiological candidates.

In the rest of this section, I present a representative selection of computational models of reflex visual attention. Computational models of visual attention may be categorised by their mechanism of action into four groups, each pertaining to the idea behind the hypothesis central to the model [106]. Each of the next four sections concentrates on one such category, and gives an overview of relevant models. Although some models may fit in several categories, I try to group them under the most fitting conceptual tag. All of these theories seek to understand how visual attention functions, with at least some parallels with the known neural pathways. They all include a bottom-up component and explain how these attentional processes may contribute to the human selective perceptual capability.

The saliency map hypothesis

The saliency map hypothesis postulates the existence of one or several maps that extract stimulus features in parallel, and of a saliency map that combines their outputs into a measure of topographic prominence. A winner-takes-all (WTA) mechanism selects the most prominent topographic location. Attention shifts happen by inhibition of the current winner. The salience measured by each feature map reflects the difference with the surrounding locations.

*Saliency maps are
WTA mechanisms*

The work of Koch & Ullman [107] is foundational in establishing the saliency map theory. But in another important publication, Desimone & Duncan [108] showed that no single explicit saliency map is required for the effect of saliency detection; and neuroimaging studies seem to favour this decentralised explanation of saliency encoding [109]. These observations are compatible with the emergent hypothesis of visual attention, which can be seen as a decentralised reformulation of the saliency map hypothesis. The emergent hypothesis is presented on page 62. Itti & Koch [109] presented a good review of salience-based focal bottom-up attention modelling. They summarise evidence of early modulation (at least down to V_1) of neurons according to attention, and propose that models of low-level attention should take that activation of a per-hypercolumn WTA into account. It is also shown that there is few interaction between modalities (between features) at low level, but a feature (e.g. oriented

*From saliency map
to emergent
attention*

*Neurological
validity*

producing the simulations, and makes the produced work less scientifically valuable, as reproducibility is a foundation of the scientific method.

contrast boundaries) can be biased by top-down attention across the whole visual field. Neurons are more sensitive to contrast and context of activation than to absolute feature value. They are activated when their selective stimulus is presented to them, but this activation is inhibited when the stimulus extends beyond their receptive fields. As a notable exception, when an orientation-selective V_1 neuron is activated by a contrast line forming a contour (thus extending to the neighbouring orientation selective neurons' receptive fields), its activation is increased. *Perceptual grouping*⁵⁶ is the result of such low-level details; this example shows how primordial the detailed functioning of the neural substrate is to the functionality of higher levels. It illustrates why the computational study of the influence of minute neuronal dynamics on network behaviour is useful, which is important as the EVAC model essentially amounts to modelling the influence of molecular dysfunctions on neurodynamics in the context of reflex attention.

The saliency map and the top-down biases can be integrated to form coherent eye movements by modelling the frontal eye fields (FEF), a dorsal region of the prefrontal cortex that is activated early upon visual or auditory stimulus onset [110]. In [111], Hamker & Zirnsak proposed a model of bottom-up visual attention that departs from traditional winner-takes-all (WTA) saliency maps in that a map modelling the FEF complements early saliency computation. Three elements in the model influence the location of attention. The salience, computed in this model in a bottom-up fashion, provides a stimulus-driven component of attention. The target, encoded in the topmost "prefrontal layer", provides, through an inferior temporal cortex (IT) layer, a top-down propagated task-based component of attention. The FEF layer, horizontally connected to the ventral visual pathway, integrates saliency and task-relevance to slowly build-up activity at the top layer of FEF. A threshold mechanism in the FEF layer indicates when an eye movement would be triggered. Additionally, a certain amount of plasticity is introduced in the topographical location of the receptive fields of V_4 . The fields adapt to the locus of attention by excluding non-attended stimuli, or even completely shifting towards the location of the attended stimulus. This matches the behaviour of V_4 neurons of primates in studies cited in the paper. Although under-specified in the cited paper,⁵⁷ the model seems a robust proposal of the integration of bottom-up and top-down attention into a motor decision.

*Possible integration
of saliency-based
reflex attention and
top-down attention*

⁵⁶ Perceptual grouping here refers the prominent idea behind Gestalt principles: the similarity, proximity, symmetry, periodicity and good continuation of individual stimuli form the whole.

⁵⁷ The connectivity and network dynamics allowing that are not specified by the authors. This is regrettable, because this voids the power of the model in proposing a theoretical explanation to an observed behaviour. The subject of the value of a computational model is further discussed in [112]

The selective routing hypotheses

The selective routing hypotheses explain how visual information is selected and transmitted across the cortex in a manner that preserves identity and location information. The receptive fields connectivity pattern with backwards projections would lead to a *pathway search problem* if they were naively implemented: with many layers between input and output with very fanned-out connectivity, the route from one particular input and its correct output is easily scrambled. More specifically, there are three sub-problems to these general routing difficulties [113]. First, the fan-out connectivity of each unit results in the topographical dilution of the activity of a single input in the output representation. This first problem is called the *blurring problem*. Second, a single output unit typically covers – through the transitivity of receptive fields – a large part of the input map, if not all of it; this is the *context problem*. Third, there is a *cross-talk problem* in that some units later in the visual pathway will be common to the representation of distinct input locations. In the case of simultaneous inputs, the resulting output activation is not clearly representative of either input signal. A selective routing hypothesis is a theoretical source selection mechanism that is able to address the pathway search problem. Such source selection mechanisms were reviewed by Korsten et al. in [114]. The author identified two approaches to input segregation in broad receptive fields of higher visual areas. On one hand, the *gating neurons* based models suppose explicit control signals and units that seem unrealistic with regard to biological mechanisms. On the other hand, the *population synchronisation* approach proposes that, in the absence of explicit control signal, source selection is performed by the repeated simultaneous firing of units. This second principle has better experimental support, and should be favoured.

*Attention and
routing problems*

The EVAC model uses recurrent excitatory connectivity between maps, and there are no gating neurons, so it would tend to fall in the latter category. However, there is no population synchronisation either, and the model does not completely address the pathway search problem, because the cross-talk and blurring problems are not solved by the proposed model of emergent visual attention under channelopathy (EVAC). The computational tasks are developed using inputs with little enough scale and intensity variations that the output map activity does not run out of control, and there is no input discrimination task, in that for most of the duration of experiments, only one object is present in the visual field. Finally, intra-map inhibition is stronger when the total net input is larger, which further mitigates possible activity explosions. So, thanks to careful experiment design coupled with simple inhibitory control, I do not have to explicitly implement a selective routing mechanism.

*Information routing
in EVAC*

The temporal tagging hypothesis

The temporal tagging model uses the synchronisation of the firing of individual neurons to select information about the features of the attended object. This hypothesis is more rooted in dynamical systems theory, with several sound theoretical models of attention, including [115, 116]. Niebur et al. [115] proposed a model of visual attention that shows the plausibility of their prior work on the “frequency tagging” of attended stimuli [117]. The model is very simplified, as they neglect any influence V_2 may have, modelling only the LGN, V_1 (sensitive to a bipolar red-green hue feature), and V_4 . The model is abstract enough that the authors did not consider which type of attention it embodies – bottom-up or top-down. They rather assumed an external selection, hinting that the model is more directly relevant to top-down.

The temporal tagging theory: a theoretical model of attention based on phase synchronisation

The more theoretical approach of Borisjuk & Kazanovich is more relevant to bottom-up attention because it is driven by a very general mechanism of phase synchronisation of oscillatory cortical activity between modules [118]. Each module, or *peripheral oscillator (PO)*, represents a cortical unit (whose scale is undetermined), and produces a signal characterised by its frequency and phase. The study specifies the manner in which such POs can, under the influence of a central oscillator (CO), exhibit regimes of synchronisation that reflect the dynamics of focal attention. Specifically, it is hypothesised that the focus of attention is formed by the synchronicity of a set of POs with the CO. Nevertheless, the paper does not indicate the mechanism by which synchronicity entails attention. From that respect, the biological support for this model is sparse. This theoretical work has the merit of modelling a possible complementary mechanism of attention at high level, where:

- the hippocampus, due to its central connectivity with many cortical areas, could take the role of CO, and
- neurons in the ventral and dorsal visual pathways would be POs.

The model was extended in further studies [119, 120], this time in an attempt to integrate the role of low-level selective attention (for feature binding) and memory (for novelty detection) into an oscillatory model of attention and visual scanpath formation. Computational units are generalised as their output varies in phase, amplitude and frequency. The amplitude of synchronised oscillators increase by resonance, marking which elements of the partially synchronised system are related to the currently attended object in the visual scene. In this paper, the POs represent cortical columns, whereas the CO is more broadly hypothesised to correspond to one of: the prefrontal cortex, some posterior parietal area, the posterior lobe, the hippocampus, or

a combination thereof. Thence, the biological support for the mechanism modelled there appears thin, but the simulation shows that, assuming a CO, the role of phase synchronisation in selective visual attention and novelty detection is computationally plausible. The authors presented the hippocampus as the most likely candidate for the role of CO. Although assumed connected to all POs, the model is simplified by having the CO connected to the first hidden layer, named object selection layer (OSL). The OSL and higher level layers represent incrementally higher functionalities of the visual pathway, whereas the output layer, performing novelty detection, is assumed to functionally represent the hippocampus. The visual pathway is forward-connected, as it does not model realistic cortical maps and, the CO being connected to the first hidden layer, attentional selection does not need to propagate backwards through the network. Synchronicity between the CO and a set S of oscillators of this layer indicates that the object attended to is represented by S . Minicolumns (oscillators) of the OSL that receive input corresponding to an object are in an active state until they enter in resonance with the oscillator (when the object is attended). They remain in that high amplitude synchronous state until the output layer has memorised or identified the attended object. At this point, these units come to a forced passive state, which may correspond to a refractory period at the neurological level. At the psychological level, that models the fact that attention is biased against returning to previous spatial locations.

*Hypothesised links
with biology*

Oscillatory neurodynamics of the brain are often studied in purely mathematical frameworks, whereas the oscillatory models of Borisjuk & Kazanovich described above are not taking the neural substrate as continuous, but blending computational and connectionist aspects into the analysis framework. [121] is another paper by Mishra et al., that showcases a computational study of phase synchronisation, this time using the compartmental simulation of an individual neuron. Two excitatory V_2 bundles, one with more synaptic connections representing the preferred stimulus of the V_4 unit, project on the dendrites of the modelled neuron. They also project to a common inhibitory interneurons pool. The unit is also subject to background inhibitory and excitatory noise. It is shown that the modelled unit responds non-linearly to the input bundles' activity rates, with a maximal response for an input frequency that depends on the number of excitatory connections. Furthermore, the frequency of the V_4 unit response is a weighted compound of its response to each individual bundle alone. Therefore, this study shows the plausibility of the frequency tagging hypothesis at neuronal level. Crick & Koch [122] explained the possible links of the temporal tagging hypothesis with biology. It presumes the existence of a spotlight of attention operating on V_4 or higher, and of certain oscillatory properties of participating neurons. [123] critically assesses the biological plausibility

*Neural plausibility
of the frequency
tagging hypothesis*

of temporal coding as the central mechanism of attention, of brought forward by the temporal tagging hypothesis, concluding with a justified negative outlook. Nevertheless, temporal tagging could play a complimentary role in a future iteration of the emergent model of reflex attention shift presented in this thesis.

The emergent attention hypothesis

In the last theory, used in EVAC, attention is an emergent property of the neural population, and not directed by a CO or saliency map. Lateral inhibition gives the population its ability to select amongst inputs, while the top-down influence of higher areas guides attention. This type of *biased competition* is conceptually simple and robust to noise. It does not require ad-hoc neural circuitry, as it emerges from the existing map structure and dynamics. The emergent attention hypothesis differs from the saliency map theory in that there is no explicit central topographic saliency map that computes the most conspicuous location.

*Emergent attention:
a decentralised
saliency map*

There are strong indicators that intra-map competition is a major component of the neurological implementation of attention. In a study of the macaque visual system, Reynolds et al. [124] showed that in areas V₂ and V₄, attention biases competitive interactions among neurons, causing them to respond primarily to the attended stimulus. In addition, [125] provided evidence that the inferotemporal cortex can take the role of the top-down selector that is hypothesised as attention guide by the emergent attention theory. These findings ground the emergent hypothesis of reflex attention in biology, as they hint at an explicit mechanism for its integration with top-down attention. The biological plausibility makes the emergent attention theory more likely than centralised saliency or pure temporal tagging, which is the foremost reason to the modelling of bottom-up attention using emergent mechanisms.

Links with biology

In related works [126–130], Deco, Rolls et al. studied the dynamics of visual attention using modular neurodynamical models of the cortical visual processing paths. The model of reflex attention developed in the thesis uses the same principles as in these works, but both implementations differ significantly in their focus. In the works of Deco et al., the computing unit is not a neuron, but a pool of neurons. Hence, these models use equations that describe the activity of a group of neurons. They make use of further simplifications, including the use of Gabor wavelet transformations to model the role of the primary visual cortex, and a Hebbian learning rule taking past activations of the post-synaptic unit into account. This framework is described in [129]. Besides abstracting away individual neurons into models of minicolumns based on Gabor filters, the implementation of Corchs & Deco further compresses layers V₂ and V₄ into one layer. It approximates the operations of both layers by using more

*Emergent attention
modelled with
minicolumn as units*

complex minicolumn models, sensitive to several spatial frequencies. In contrast, this thesis models individual neurons, and scales down the number of neurons per minicolumn to keep the computational cost manageable. This difference between the two models stems from their different goals. The model of Corchs & Deco aims at validating this particular algorithmic theory of emergent attention based on lateral competition and parietal top-down bias, and comparing the behaviour of the network in competitive attentional tasks to relevant clinical results (fMRI study). They presented a computational model of attention based on emergent selection. In their model, the *what* and *where* visual pathways are explicitly modelled as successive retinotopic neural maps with reciprocal backwards connections, and units take their inputs over receptive fields throughout the network. The architecture in [126] and related works makes it possible to simulate spatially-guided visual search by providing a high-level input that propagates backwards from the dorsal stream, or to simulate the visual search for an object by biasing the topmost ventral module. This allows simulating visual search for an object or voluntary visual attention to a specific region in space. Such tasks are shown to activate the prefrontal cortex in a way that biases activation of the corresponding ventral or dorsal pathway. In an object-based top-down attention task, the activation of the lower-level features that compound the object is positively biased. This is due to the back-propagation of the object bias. Following that bias mediated by the object pathway, the activity of the spatial pathway reflects the spatial location of the activated target object. In the spatial attention task, the same mechanism operates, however initiated this time from the biased dorsal stream, with the object pathway running free. This model is used to make qualitative predictions, for instance on the effect of eccentricity to the fovea on the firing rate of the infra-temporal cortex unit associated (by learning) with the object presented [129]. The overarching goal is to guide the understanding of the neuronal dynamics of the visual network, encompassing attention mechanisms.

*Overview of the
model of Corchs &
Deco*

Their simulation is presented as a support for the scientific viability of the underlying inter-modular model of attention (between the visual cortices taking part in the dorsal and ventral streams). The authors described the ways by which the simulation either quantitatively or qualitatively conforms with experimental observations, showing that apparent serial processing (at the psychological level) can come from emerging dynamics of parallel mechanisms at the neurodynamical level.

The model of attention expounded in this dissertation differs from the related works of Deco et al. in several important ways:

- First, it is not possible to abstract away units and use the same minicolumn models as in [126], because I am studying the dynamics of networks in response to subunit-level changes (cal-

*Differences with the
EVAC model of
emergent visual
attention*

cium channelopathy model). For instance, abstracting units into Gabor-filtering minicolumn models would require to already have a specification of the exact influence of channelopathy on the minicolumn model, which has not been researched.

- Second, the model of Corchs & Deco [126] incorporates independent top-down influence, and the computational tasks and their real-world equivalents actively seek to manipulate and qualify this influence over periods of time of more than 2 minutes. This is very different from EVAC, which focuses on non-biased competition. Although the top-down pathways are modelled, there is no other activation coming from high areas than the normal recurrent activation following feed-forward signal progression. In other words, there is no top-down attention. This is an explicit design decision, as the goal of EVAC is to only explore bottom-up emergent visual attentive processes in channelopathic vs. control networks, with attention shifts mediated by neuronal accommodation. For that, the computational task and its real-world equivalent are designed to minimise top-down bias: each task occurs over a few hundred milliseconds, and the subject is in a relaxed state of mind, not searching any visual stimulus.

As the EVAC model of purely reflex visual attention accounts for attention shifts, it is important to examine the mechanism of *inhibition of return* in the emergent model of attention, and how it may be computationally modelled. Inhibition of return is the relative suppression of the neural representation of a recently attended stimulus. This suppression results in the change of saliency from the currently active stimulus to stimuli that have not been recently attended (novelty preference) [131].

*Mechanism of
attention shift*

As a result, attention shifts are accounted for by neural accommodation – the short-term post-activation depression mentioned earlier – or other inhibitory mechanisms implementing inhibition of return on the saliency map or its possibly distributed equivalent. However, the psycho-physical phenomenon of inhibition of return has been shown to interact with higher levels of cognition in a more intricate way than what has ever been modelled, as it allows for instance for a moving observation frame, or for tracking mobile objects. Even without accounting for such interactions, a biologically accurate model of inhibition of return poses several modelling challenges. First, it likely involves a significant amount of interaction, both between several processing pathways at different levels of cognition, and between bottom-up and feedback streams. Second, an accurate model of inhibition can not solely be modelled by the neuron-local phenomenon of accommodation [1, chap. 8, p. 271]. The same problem of interaction of streams emerges when one tries to model *attentional scanpaths*

on longer time-spans than a few hundred milliseconds. The interaction of top-down attention and reflex attention needs to be accurately modelled when considering longer timescales, as top-down attention becomes increasingly influential. In abstract terms, this means modelling the interaction of saliency maps with top-down semantic attention (object recognition and identification). Olshausen et al. [132] were among the first to propose an extensive computational model of such interactions. However, neuroscience and psychology have yet to provide precise models of the interaction of the top-down and bottom-up streams and of the dorsal and ventral paths when considering attention control. Hence, no model has been crafted yet with extensive experimental support of the mechanisms used.

This emergent model of reflex visual attention in channelopathy ought to be put in the context of models of attention in autism, as channelopathy is hypothesised here to be a core component of autism. The next section presents existing computational works on attention in autism and related disorders.

3.3.4 *Models of attention and ASD*

There have been few computational studies on brain disorders, as the understanding of most neuro-pathologies would not necessarily be immediately improved by efforts in computational modelling. The pathologies that could benefit from modelling are those that are holistic and hard to outline; in particular, syndromes, by definition. Autism is very appropriate for this type of computational study because it is a prevalent syndrome with increasing prevalence rates, and is given a lot of attention from the general public and medical community. In the following, I review several computational models of autism, and place them in relation to the work reported in this thesis.

Why model autism?

Qualitative computational models

Recent studies have tried to model autistic traits with simplified connectionist models, because very abstract models are useful to illustrate possible conceptual relationships between properties of artificial neural networks and the idiosyncratic behaviour of autistic persons.

In [133], McClelland paralleled the bias/variance trade-off as perceived in the context of connectionist model building to the *hyperspecificity*⁵⁸ that autistic people tend to exhibit. The goal of the work is not to go into biological details, but rather to parallel artificial neural networks and biological neural networks; for instance, the author compared the over-specificity of autistic learning to a high-threshold conjunctive coding. This type of very abstract model is not so much

*High-level
connectionist models*

⁵⁸ Hyperspecificity is defined as the tendency to learn overly specific representations of information.

computational as conceptual. The computational implementation is meant to be mostly illustrative rather than predictive. Nevertheless, the claim of autistic hyperspecificity is worth bearing in mind, as it has neurodynamical implications.

Cohen [134] presented another general and exploratory model of the effect of variations in the number of synaptic connections on the generalisation and discrimination (linked to attention) of a simple 3-layers feed-forward artificial neural network with back-propagation learning rule. The paper shows that it is in principle possible that some of the autistic symptoms observed are a direct consequence of unusual synaptic density. As with [133], the simulations do not lead to quantitative behavioural predictions on a specific task, but serve as an illustrative purpose. They also show the value of simplified models of neural networks.

Self-organising maps-based models

Self-organising feature maps (SOM), or Kohonen networks, are artificial neural network that were introduced in a seminal paper by Kohonen [10]. A self-organised feature map (SOM) is a rectilinear grid of feature detecting neurons interacting via a winner-takes-all (WTA) function, together with a learning algorithm which adjusts the sensitivity of the winner neuron and a limited set of its neighbours towards the input feature. They are very popular base tools to build models of cortical networks. This is in part due to their simplicity. SOM are also popular due to the approximate similarity of cortical maps development or long-term cortical learning with SOM self-organisation. This is particularly applicable to early sensory areas. Elements of this self-organisation by the winner-takes-all⁵⁹ process are used in all network-level connectionist models of the cortex. A simple Kohonen SOM taking its input in one invariant location of an input map will permit the modelling of a continuous feature map, roughly corresponding to one cortical minicolumn. With distant lateral interactions, other more elaborate types of self-organising feature maps will allow for more realistic modelling of cortical maps.⁶⁰

The model of emergent visual attention under channelopathy (EVAC) presented in the next chapter employs SOM-inspired neural networks learning algorithms, combined with error-driven learning. Several studies have shown that the application of SOM to the modelling of biological networks is a valid approach.

A good example of the application of SOM to attention modelling is [136]. In this study, Gustafsson & Papliński created a general model of sensory cortical feature maps. They used self-organising maps to

Kohonen maps are simple tools that may mimic cortical connectivity.

⁵⁹ It may be a winner-takes-all or a k-winners-take-all process: the latter follows the same principles as WTA, but allows for $k > 1$ winners to be selected at each epoch.

⁶⁰ See for instance the laterally interconnected, synergetically self-organizing maps (LISSOM) used to model V_1 in the Topographica modelling software [135].

model the development of cortical maps and assess whether attentional impairments in autism are due to difficulties to shift attention, to higher familiarity preference or to a more negative response to novel stimuli (novelty avoidance). These terms relate to psychological concepts, and may seem to paraphrase or imply each other, but the authors clarified this. Without arguing that autistic subjects have attention-related impairments, they presented two existing positions in the literature regarding the nature of these impairments.

The first idea is that autistic individuals have difficulties to shift attention, partly because of cerebellar damage (loss of Purkinje cells). This line of thought is primarily followed by Courchesne et al. [137]. A related possibility is that the impairment to shift attention is only expressed when in connection with specific types of stimuli or mental process [136, page 190]. This would physically locate the deficiency in higher-level processing or even control areas of the cortex rather in the cerebellum.

Alternately, the second idea considered by Gustafsson & Papliński in their model is that attention impairments are due to a preference for sameness, as such a preference describes a cardinal autistic trait. In this light, *novelty avoidance* would mean that, in presence of familiar stimuli, a novel stimulus is actively avoided, while *familiarity preference* would imply that a new stimulus is just ignored. The authors used a SOM model because SOM properties make them adequate for modelling the learning process in cortical maps. However, the SOM used by Gustafsson & Papliński is not an in-depth simulation of biological feature maps, but a standard SOM using a 1024-epoch simulated annealing learning.⁶¹ They assumed that their model captures the minimal learning mechanisms to adequately model network learning difficulties in autistic children. Abstraction capabilities in affected individuals may be hindered by deficient learning of their cortical maps. These models test the different effects on learning of general attention shift difficulties and familiarity preference. Attention-shift impairment is modelled by lowering the probability for the network to take new inputs in consideration, while familiarity preference means that the attention shift probability is conditional on the map's familiarity with the symbol presented, where familiarity is represented by distance from the stimulus to the learnt cluster centre. The results show that learning is much more affected by familiarity preference than by attention-shift impairments: typically, the stimuli with smaller variance will be learnt at the expense of the other. In a later work [138], the authors followed up on that familiarity preference model by presenting a model of learning therapy that modifies the probabilities of the stimulus class presented according to the current preference bias of the network. This type of study on familiar-

Using SOM to model familiarity preference and novelty avoidance as factors of autism development

⁶¹ Simulated annealing is a artificial neural network learning rate adaptation scheme which I also use in learning the weights of EVAC.

ity preference could be extended by modelling the influence of channelopathy on learning. The network presented in this thesis does not model the influence of channelopathy on learning, but its effects on an already formed network. Hence, it may be worthwhile to investigate the influence of channelopathy on learning in terms of familiarity preference.

Much like in the works of Gustafsson & Papliński, Noriega et al. used SOM to study the effect of abstract affects on networks on point neurons. The hypothesis modelled is that of a connectivity issue in autism, suggested by abnormal neuronal growth and synaptic densities. It is proposed that, during infancy, such defects have an effect on learning that corresponds to the symptoms of autism. The authors of [139–141] proposed a rough model of neuronal growth. They defined a measure of map unfolding and examine the effect of temporarily increasing the number of units. They observed that the final degree of unfolding does not depend on the epoch in which the temporary increase of map size happens, but on the intensity of overgrowth modelled. Additionally, the effects of hypersensitivity and hyposensitivity are modelled by the use of “attention functions” that modulate the lateral interactions of units. This particular way of modelling unusual sensitivities to certain input modalities does not appear biologically justified. The author expanded that particular model in [140] by modelling the effect of a feedback mechanism on the SOM. This paper makes the parallel with the weak central coherence theory of autism, but does not to bring the model closer to the biological reality. Similarly, in [139], the particular attention to details by autistic children is simulated by rejecting distant neighbours, which is however not justified by clinical observations.

Overall, most literature on modelling neurological disorders extensively uses SOM. The rationale is that the learning of self-organising maps approximates the development of the cortical maps. In the context of the study of developmental disorders such as ASD, SOM appears as an appropriate primary approach tool. In the case of autism, the development of brain structures appears to be hindered from very early age (actually prenatally, see for instance [142] for neurological evidence). The actual short-term, low-level learning mechanisms themselves do not appear affected, as once an autistic child’s interest in a task, game or object is locked, he will learn the task without issue, if not hindered by specific aspects of the task that make it less accessible to him. Discounting for the frequent co-occurrence of mental retardation and poor motor control, abstraction and ambiguity are characteristics that make a task less accessible for autistic children. This is due to the cognitive trait that underlies autistic cognition, and in fact leads to some of the typical problems observed with social interactions and language. A task that requires learning and avoids such ambiguities of social contact and language can be learnt as well

as or better than controls.⁶² Psychological evidence is reviewed in [143]. In view of this evidence, in the case of autism, the SOM approach does in fact aim at modelling the brain development from early age and on long time scales. However, neural development is poorly modelled by SOM. While SOM-based simulation helps to generate similar patterns of units sensitivity in isolated maps [135], several aspects of SOM learning make it insufficient for modelling the actual mechanisms that drive brain development:

SOM as a model of long-term brain development

- Self-organised Hebbian learning, as modelled by SOM, does play a role in brain development, as for instance exposure to a given language during growth forges the phonetic discrimination capabilities of children. However, the influence of learning on development is not limited to the localised, horizontal learning modelled by SOM. Deeper than the sensory input layers (e. g. V_1), vertical learning based on rewards and expectations is dominant. This aspect is further discussed on page 103.
- It appears that the SOM algorithm can result in topographical maps with units that have similar reception patterns to those observed in some input maps of the cortex. Still, this result does not imply that the SOM algorithm is sufficiently biologically realistic to predict the outcome of disturbed brain development. If the SOM algorithm is functionally similar to the effect of lateral interactions in cortical maps, the implementation of the algorithm is centralised and does not involve inhibitory interneurons,⁶³ two properties completely incompatible with the biological reality.

In addition, the brain development process is far more involved than SOM learning; there are unique mechanisms guiding the migration of neurons, controlling their production, the growth of various structures, the timing of events, and current knowledge on the matter is very incomplete.

In that light, the prospect of modelling the development of autistic brain stands out as overly difficult. A more realistic study would be to examine the immediate dynamical effects of anomalies on cognition in an already affected network, rather than their long-term anomalies on development. To make the implementation of a computational model helpful, a middle point has to be taken between complexity and explanatory power. It seems that the current computational frameworks and knowledge about the central nervous system (CNS)

⁶² For instance, autistic subjects perform normally or better than average on experiments that involve memorising large quantities of specific information, music imitation tasks, or learning technically precise knowledge such as grammatical rules are all

⁶³ SOM use direct lateral inhibitory interactions, which does not occur in the brain as a neuron produces only one transmitter, with most often a consistent (either inhibitory or excitatory) post-synaptic effect among all receiving neurons.

place this point at the level of short-term network dynamics, with simplified units with parameters that replicate selected biological factors of units of the right scale; namely neurons, cortical minicolumns or even functional areas.

In EVAC, self-organised feature maps (SOMs) still have a central role, but the self-organised component of the learning algorithm is not meant as a model of cortical development. Rather, SOM learning is used in conjunction with error-driven learning for the cortical-like organisation of the resulting network.

Non SOM-based models

Oliver et al. [144] took a novel approach to modelling developmental disorders, departing from the traditional computational lesion studies which are less adapted to seemingly pervasive affects of the central nervous system. In contrast, the authors promoted the study of the emergence of non-neurotypical dynamical properties of networks in face of atypical characteristics of lower-level units, and made the effort to develop tools to understand neurodynamics, rather than take a pure black box analysis approach.

Oliver et al. [144] performed a study of the development of neural networks, on the reasonable premises that developmental disorders are:

- not prototypically due to localised, discrete brain lesions, but to more subtle neuron and network properties,
- better understood in terms of development, network formation and emergence of representations (referred to as *developmental trajectory* in the cited paper),
- dependent on the environment – the behaviour of a developed network depends on *the structure of the information to be processed*,
- syndromes with graded symptoms.

A model of cortical development

On these bases, the authors modelled a general cortical map, with randomly distributed and laterally interconnected excitatory and inhibitory point units, the excitatory units receiving direct signals from an input map. Learning is the central interest of the model, as it is supposed to simulate the process of brain development. Indeed, the model learns using by a modified Hebbian rule that prunes or stabilises synaptic connections, a phenomenon proven to take a major part in infant brain development. In this framework, the study examines the influence on the excitatory units' threshold of the distribution of lateral and afferent projections, by changing the synapse pruning rate and the correlation of inputs.

At regular intervals during learning, the response of each cortical node to each stimulus is characterised and compared to that of its

neighbours to qualify the unit as Clustered (C) with its neighbours, Topographically ordered (T), Inert (I) or Uncommitted (U). This reduces the state space of each unit to four binary variables. By counting and normalising the number of units in each state, the whole map's state can be projected as a point on a simple hyperplane.⁶⁴ This interesting approach to visualising the network⁶⁵ lets one interpret trajectories in the projected space as the evolution of a cortical network during brain development in terms of overall clusteredness, topographical ordering, unresponsiveness, and maturity.

*Dimensionality
reduction and a
measure of cortical
development*

However, while [144] can clearly describe low-level effects of parameter changes, it appears at best exploratory when suggesting how high-level cognitive faculties that define such syndromes as autism are affected by these changes. This is in part due to the fact that the study means to model a general cortical map, instead of a specific subsystem. But it also highlights that the main obstacle of computational research in neuroscience remains the complexity of the brain. The study presented in the remainder of this thesis does not face the first obstacle, as it is specifically targeted at the visual system. However, it does suffer from the second issue; namely, the simplifications made limit the accuracy of the model prediction made by the simulation.

3.3.5 *Summary table of computational levels of study*

To give an overview of existing models of brain mechanisms and the level of abstraction that they imply, I have organised them into a table.

Table 3.1 synthesises approaches to computational modelling of neural mechanisms, in approximately decreasing order of abstraction. It also gives examples of applications related to the modelling of cognitive and/or neural disorders.

The model category that corresponds to the one presented in this part of the dissertation is named "Hybrid deep feed-forward (task learning) and SOM (associative learning) networks", page 73.

⁶⁴ The constraint that $C + T + I + U = 1$ restricts the state projection to the plane described by that equation in the bounded continuous four-dimensional space $\{C, T, I, U\} \in [0, 1]^4$.

⁶⁵ A cortical network is a high-dimensional dynamical system which would be more difficult to grasp without such dimensionality reduction.

Table 3.1: Taxonomic table of computational models of brain mechanisms and their applications to brain disorders modelling

Model	Model description	Ref. publi.	Applications to CNS disorders
Production rules based cognitive architectures (ACT-R, SOAR) and other abstract cognitive theories	These models usually encompass most major function of the brain: sensory, executive, motor, and memory. They are relatively detached from biological realism other than behavioural observations.	[24]	Due to the lack of biological details, disorders that result from specific mechanisms (expression of deficient genotype, brain lesions, specific chemical influence of the environment) can not be directly modelled.
Hybrid abstract and connectionist models	These models keep the scope of their cognitivist counterparts, but attempt to implement their mechanisms with biologically more plausible units of computation	[31] [32]	Hybrid architectures have tried to increase the flexibility of abstract models, but have not solved the conceptual discrepancy between the theoretical modules of cognition hypothesised by those and the cortical areas organisation, and remain difficult to use for cognitive disorders study.
Connectionist networks of concepts (semantic network)	Units have a definite meaning or label. For instance, in a study of language impairments, units may be words.	[145]	[133] takes a high-level approach to the understanding of hyper-specific learning in autism, using ontological networks.

more abstract /
↑
higher-level models

continued on next page

continued from previous page

Model	Model description	Ref. publi.	Applications to CNS disorders
Pure feed-forward networks with back-propagation learning rule	Task learning is not biologically realistically modelled by the back-propagation algorithm. However, the end result may be used to simulate cortical pathways.	[26]	[134] uses such a model to study variations in synaptic density linked to autism. The lack of realism only allows for general qualitative observations.
Pure SOM-based models of cortical maps	A unit represents a group of biological neurons, possibly a full minicolumn. Units are very detached from biological reality; this model's interest is that it is the simplest representation of the SOM-like feature learning taking place in cortical maps.	[136]	At the primary processing stages of sensory data, unsupervised learning creates topographical cortical feature maps; Hence, simplified SOM-based models can be used to assess problems in such first processing stages. [136]
Hybrid deep feed-forward (task learning) and SOM (associative learning) networks	Biological networks are known to perform both input pattern and task learning. While still using simplified (point) units, it is possible to create models that mimic biological learning. The challenge is to avoid the biologically unrealistic back-propagation.	[3] [1]	[35] in which the Leabra algorithm that performs local, error-driven and associative, biologically realistic learning, is used to quantitatively study the effect of specific cortical lesions on vision.
Modular neurodynamical models of visual paths	In such models, units are pools of several neurons. Modelling the activity of such a group of neurons necessitates greatly simplifying its dynamics, but allows for networks whose scale is a compromise between abstract area-level models that use the simplest form of SOM and unit-level networks of smaller cortical patches.	[129]	No lesion studies, but the results of these models are experimentally tested [130].

more detailed /

smaller-scale models

visual attention models

continued on next page

continued from previous page

Model	Model description	Ref. publi.	Applications to CNS disorders
Classical saliency map based multi-layer models of bottom-up visual attention	Point units are arranged in two-dimensional maps representing cortical maps. One of the maps is central to bottom-up attention because it is sensitive to visual salience. It is used to bias the other cortical maps and hence guide bottom-up attention.	[109]	In [146], a saliency map based model of visual attention was used to explain the fixation of autistic children on the mouth rather than the eyes. However, in this model of autism, it is the top-down bias onto this map which is hypothesised to be affected, rather than the bottom-up attention mechanism.
Oscillatory models of bottom-up visual attention	Specific firing frequencies of groups of neurons are effectively "tagging" a specific location/object in the visual space. Possibly, a central oscillator (CO) is coupled with peripheral oscillators (POs) and determines the currently attended object.	[115] [118] [119] [120]	Although [147] clearly explains the interest of temporal synchrony studies in the context of the weak central coherence theory of autism, I could not find implemented computational oscillatory models of attention in a PDD.
Soliton models of nerves	A very recent alternative model of information propagation within neurons that is based on the propagation of solitary compression waves in the lipophosphate membrane of neurons. This theory permits the explanation of some phenomena that are unaccounted for by the Hodgkin-Huxley model of AP propagation (isothermic AP).	[148] [149]	The soliton propagation model is very recent, but has been used to propose an explanation of the previously unexplained action of certain anaesthetics [150]. Therefore, it potentially holds explanatory power to link molecular neurodynamics and behaviour, an interesting prospect for computational autism research.

continued on next page

visual attention models

more abstract /
higher-level models

continued from previous page

Model	Model description	Ref. publi.	Applications to CNS disorders
Studies of biological neurons or networks using compartmental neuron models	Many sub-cellular (morphological or biochemical) variations are difficult to model using the aggregate expression of their influence on membrane potential that constitutes the point neuron equations. Compartmental models express the channel flows in a piece-wise manner between sections (compartments) of the dendrites. Compartments assumed internally homogeneous.	[121]	Several works use compartmental modelling to study sub-cellular factors of neurological disorders. [151] stands out in its methodology of iterative refining of models of epileptiform activity in single neurons.
Molecular-level simulations of membrane dynamics	Molecular dynamics simulations are used to study the interaction between individual molecules or the dynamics of small populations of molecules. In particular, the precise timeline of the passage of ions through membrane channels is commonly studied using such models. However, current techniques are difficult to scale to the whole cell.	[152]	Variants of cell structures such as malformed ion channels can have pathogenic effects on the phenotype. Simulation of molecular interactions can help in the design of drugs that may for instance bind to the protein and beneficially alter its dynamics.

more detailed /

smaller-scale models

3.4 POTENTIAL CONTRIBUTIONS OF RESEARCH

The EVAC model presented in this dissertation could be useful not only for the development of the calcium channelopathy of autism, but also:

- to a wider audience of cognitive scientists, by providing them with insights in the brain-mind gap, and
- to information systems engineers, by inspiring implementation ideas to reduce the processing load of computer vision applications, in particular with strong real-time processing requirements and/or limited computational resources.

3.4.1 *Neuroscientific importance*

Understanding the possible role of low-level brain mechanisms in autism

As outlined in the previous sections, autism is a developmental disorder with a vast and unexplored etiology. The more research goes into exploring autistic biochemistry, genetics and physiology, the more factors and intricate influences are discovered. Without doubt, a better understanding of autism means a better understanding of the high-level functional role of these low-level chemical mechanisms whose malfunctions contribute to autism. However, it most often appears that higher-level functioning of the brain can not be unambiguously deduced from experiments and observations. Social intelligence and the autistic way of thinking⁶⁶ are, in terms of cognitive processes, *mare incognitum*. For instance, savant feats of autistic persons are often suspected to be associated with a greater capacity or different use of memory, but this characteristic, as well as all other traits specific to autism, are neither explained in terms of neural connectivity nor shown to be due to different activation of brain regions. For the cognitive neuroscience researcher, autism research aims at understanding the high-level modifications of brain functioning that arise from the phenotypical expression of the autism-prone genetic information and its interaction with the environment. Because of the particular extent of the cognitive symptoms of autism, the interest of fully understanding its mechanisms is not only to indicate the ways to possible treatments, but also to acquire the understanding of how social intelligence and some aspects of cognition and human behaviour emerge from biochemistry, brain development, and neural activity.

Bridging the brain-behaviour gap

Full-fledged computational simulations of the brain are currently impossible in practice, due to problems of scale. Hence, models of certain aspects of cognition, of cognitive mechanisms have to be limited in scale and in comprehensiveness.

⁶⁶ Mental retardation is prevalent in most of the cases, but even more prevalent is a different approach to problem solving and interaction, which famously leads to very specific and technical savants skills in many high-functioning individuals.

However, this limitation only prevents large-scale integrative simulations of the brain. Specific aspects of cognition, limited scale models can still be constructed. O'Reilly & Munakata [1] argued that being constrained in terms of complexity helps to create a minimal model of the system studied, without the clutter of irrelevant or excessively detailed mechanisms. Such minimalistic computational models of aspects of the central nervous system have indeed been constructed and incrementally enriched, as has been exposed in the previous sections.

In this light, the EVAC model of channelopathies potentially stands out as a useful contribution to research on autism, as calcium channelopathies are new but central etiological elements of autism and related PDD. Although many details are omitted in the model, it provides some initial insights into the neurological-to-behavioural link of channelopathies.

*Better
understanding
channelopathies*

In general, EVAC has the potential to offer insights about the the dynamics of disturbances of the accommodation and hysteresis of cortical neurons. That includes channelopathies, but could potentially be applied to environmental factors, or to study the neural basis of cognition. In particular, the autistic cognitive style, characterised by hyperspecific cognition [133], could very well be an extreme occurrence in a spectrum of neurocognitive styles. In that case, EVAC could be a starting point for the study of such a spectrum.

*Potential
generalisation of
EVAC*

3.4.2 Technological importance

Attention is not an exclusive interest of researchers who study the brain and the behaviour. In computer vision, attention plays a crucial role in the engineering of automated data processing systems. Indeed, it has always been important to prioritise or reduce the amount of visual information processed by the computer systems. Often, machine vision algorithms implementations will run within strongly constrained environments in terms of time, energy, memory and processing power. But even with relaxed constraints, the nature of visual information demands attentional processes, because of the combinatorial explosion of visual search combined with the problems of perceptual organisation that underlie the object and scene recognition.

*Attention is used in
computer vision to
reduce information
load.*

For that, computer vision researchers have taken a lot of inspiration from knowledge and theories of biological visual attention. For instance, [153] makes use of the saliency map theory of visual attention for rapid scene analysis.

Hence, in the context of the neural network based implementation of bottom-up attention as used in computer vision, the proposed model of the influence of accommodation and hysteresis on attention could be useful in several ways. First, the development of point neuron models that parallel the dynamics of real neurons is helpful to create artificial systems with human-like attentional capabilities.

*Biological
inspiration in
computer vision*

ies. Second, if the dynamics of accommodation and hysteresis indeed turn out to truly be central elements in the hyperspecificity of autistic cognition, the model of proposed emergent visual attention under channelopathy (EVAC) would provide a direct way to adjust the dynamics of an artificial neural network-based attentional filter in terms of selection specificity.

However, the thesis does not explore possible engineering applications of the model developed. It only presents its application to the field of computational neuroscience.

EXPERIMENT, MODEL ARCHITECTURE AND IMPLEMENTATION

The previous chapter justified the implementation of an accommodation-based emergent model of bottom-up visual attention, in view of studying the possible effects of calcium channelopathies, hypothesised as a root factor of autistic behaviour. This chapter details the structure and algorithms of the model on this basis.

The model uses eight neural maps to implement the dual-pathways visual system. The Hodgkin-Huxley type point neurons that form these populations are forward- and backward-connected by between-maps excitatory projections, while within-map inhibition ensures that activity is regulated within and between cortical minicolumns.

The synaptic weights are learnt using a hybrid self-organised and error-driven algorithm called Leabra, introduced by O'Reilly in [3]. This algorithm combines Kohonen maps principles [10] with the error-driven learning principles or the Generalised Recirculation algorithm [154], in order to achieve learning compatible with the possibilities of biological neural networks.

Before going into the details of the model of emergent visual attention under channelopathy (EVAC), it is important to unambiguously specify the parts of the central nervous system being modelled and the tasks the model will accomplish.

Summary

4.1 MODELLED ENVIRONMENT AND ORGANISM

The EVAC model targets the reflex orientation of visual attention. In humans, several types of eye movements (reflex or intentional, of varying angle, speed and frequency) respond to the need of the central nervous system to compensate for

- the physiological characteristics of the retina,
- the processing limitations of cortical maps and sub-cortical structures, or
- the absolute movements of the head.

In particular, the brain compensates for absolute head movements in a reflex loop called the *vestibulo-ocular reflex* (VOR). This reflex is independent of visual stimulus, and uses the inner ear to detect head translation and rotation. It occurs in less than 10 milliseconds. The reflex seems unaffected in high-functioning autistic children [155].

Mechanics of eye movement

Hence, I can model reflex attention shift and meaningfully compare simulated experiment results to the results of clinical experiments without having to account for the vestibulo-ocular reflex (VOR) in EVAC.

*Density of retinal
photoreceptors*

*Saccades and head
movements*

To decide the size of the section of retinal input map being modelled, it is important to understand its functioning, and the relation with the reflex orienting of attention. On the retina, the resolution distribution is not uniform: the fovea is a central patch of the retina where the receptive fields are the smallest and most dense, resulting in a much higher input resolution. Perceptual processes require the fovea to be aligned with the region of interest to build the detailed representation of the object. However, the human fovea only covers about 2° of visual angle. This area is insufficient to build a locally detailed map of a visual scene. This property of the retina is compensated by *saccades*. They are fast eye movements that align the fovea with the parts of the visual scene that present a greater interest or salience (memory or visually guided saccade and predictive saccades), or to come back from an off-centre gaze to a preceding position in the optokinetic reflex. As such, saccades towards salient stimuli are the primary behavioural manifestation of bottom-up visual attention. Contrarily to the VOR and to microsaccades, conscious control can greatly influence the frequency and direction of saccades. The angle of saccades can vary from a few degrees to a full right angle, but an angular shift of more than 20° will typically be accompanied by a head movement. In the presence of head movement, saccades are more sophisticated in that Listing's law⁶⁷ is no more held valid, and the VOR interacts with the saccade to keep the fovea on target at all times. Saccades to an unexpected stimulus take about 200ms to initiate in normally developing children. The angular speed of the eye varies on the angular distance to cover, but varies linearly ($300^\circ/\text{s}$ for an amplitude of 10° , 30° covered at $500^\circ/\text{s}$) up to 60° of rotation, and remains under a maximal rotation speed above that.

*The input map
models a few degrees
of visual field.*

These considerations reveal that modelling larger attention shifts gives more challenges than modelling horizontal shifts of less than 20° . Moreover, due to the heterogenous density of photoreceptors on the retina and variations in receptive fields sizes, it is preferable to limit the model to a small area close to fovea, so as to keep EVAC simple. Most models of retinal input maps are rectilinear grids of sensors, which is much more computationally tractable, but as the density of receptors on the retina decreases the farther from the fovea, it is more accurate to consider that the model is only meant to capture attention shifts within a few horizontal degree of the fovea. The model of emergent visual attention under channelopathy (EVAC)

⁶⁷ Listing's law states that there exists a reference line of sight (defined about an axis in a reference plane, unique per reference line of sight) from which any other line of sight can be reached solely by rotations about axes in the reference plane (*id est* without torsion).

follows that common practice, using a rectilinear input map to cover a few degrees around the fovea.

4.2 TASK/TRIAL STRUCTURE

The aim is to model human visual attention capture and measure its timing in a neuron-level simulation. For that, I design several input/output tasks⁶⁸ that the network then performs repeatedly to collect timing and performance statistics. These tasks are inspired by existing studies in psychology and computational cognitive neuroscience. Below, I review the most relevant works.

Both the network model and the task conditions designed are inspired by the computational studies of O'Reilly & Munakata. In [1], O'Reilly & Munakata implemented another, simpler computational experiment inspired by the Posner Spatial Cueing Task [2]. Computational experiments inspired by the Posner task usually follow its principles. The Posner paradigm originally aimed at assessing the influence of the covert orienting of attention on reaction time in a simple task: the subject is first shown a visual cue, followed by an off-centre stimulus. The cue may indicate where to expect the stimulus to appear (this is called a valid trial), or may be misleading (in an invalid trial). Posner considers variations of in the timing between the appearance of the second stimulus and the saccade towards it in terms of three phases: *disengaging* from the current focus of attention, *moving* attention to the location of the target, and *engaging* the target. The attention experiment in [1, p. 261] uses the simplest possible model of visual processing⁶⁹ to qualitatively illustrate that the timing differences between valid and invalid trials are compatible with bottom-up, emergent visual attention.

*The Posner
paradigm*

More in line with the work presented here, the same authors attempted to implement a spatial attention shift model using the PDP++ simulator, without success.⁷⁰ [1] describes the model used in this attempt. The goal of that model is to implement accommodation-based sequential attention. This model conforms with the emergent attention hypothesis, and uses unit accommodation as the basis of sequential processing, as described in Section 3.3.3. Structurally, the EVAC model of reflex visual attention is closer to this second model aimed at replicating attention shift by O'Reilly & Munakata [1]. However, the tasks are inspired by the Posner paradigm.

*Models of O'Reilly
& Munakata [1]*

⁶⁸ These tasks are designed to have parallels in the physical world; hence the computational experiment designed here can be viewed as simulating the clinical experiment described in Section 6.6.1 page 141.

⁶⁹ The model in [1] uses a one-dimensional input space comprising of 7 discrete locations, 2 input categories input on a 14-units map diverging into a dorsal and a ventral pathway, each comprising of one hidden and one output layer.

⁷⁰ See authors' note at grey.colourado.edu/CompCogNeuro.

The computational experiment comprises of three simple task conditions: Neutral, Gap, and Overlap. Like in the Posner paradigm, these tasks allow the exploration of the way interaction between the ventral and dorsal processing streams enable the reflex orientation of attention. Each task corresponds to one condition, and is either performed by a network with units simulating channelopathy, or by a control network, with nominal unit parameters. The next section describes the three task conditions in details.

4.2.1 The Neutral condition

The Neutral condition is meant to assess the time to fix attention in the absence of any distractor, from a neutral state of activity.

Description of the Neutral condition

In this condition, the network is at rest, without any input stimulus. After a short period of time, a stimulus appears at a random location on the input map. The time is measured from the appearance of that only stimulus to the settling of the dorsal and ventral maps on the correct location and identity. Figure 4.1 is a time diagram of the Neutral task condition, showing its two phases N(1) and N(2).

Figure 4.1: Organisation of the Neutral task condition in the simulated reflex attention shift experiment

The Neutral condition is a control condition, as it is not an attention shift task, but meant to be compared against the other conditions that are directly related to attention shift. In terms of the three phases of attention shift that were posited by Posner, the Neutral condition should assess the time to *move* attention to the location of the target, and to *engage* the target. The *disengagement* from the current focus of

Rationale of the Neutral condition

attention does not exist in the Neutral condition, because the network is not presented with any input.

At the start of the task, the network activity is null, and all neurons' membrane potentials are at rest. Hence, it is expected that, during the first phase of the Neutral task condition N(1), in which there is no input, the output layers of the control network remain inactive. The channelopathic network should also remain inactive, as accommodation and hysteresis, the two parameters affected by channelopathy, only affect the activity of neurons beyond a short period of activity. During the second phase of the Neural task condition N(2), the timing engagement of attention to the emerging stimulus could be affected by channelopathy. The task finishes when both output maps have settled on the correct stimulus location and identity, or failed to do so after a timeout.

The movement of attention is a psychological concept that is expected not to have a great significance at the neural level in the Neutral task, as attention is not shifted from one location to another, or from one object identity to another. Instead, activity is expected to emerge at the right spatial location on the dorsal output map, and with the correct output category, thanks to the interaction of all maps' k-winners-take-all (k-WTA) inhibition and bidirectional excitation.

*Expected
neurodynamics of
the Neutral
condition*

4.2.2 The Gap condition

The Gap condition is the first of the attention shift tasks. It is meant to study attention shift when the network has recently been (but is not anymore) presented with a distractor.

The four phases of the timeline of the Gap condition are represented in Figure 4.2 on page 84. During phase G(1), the network is first run without any stimulus for a small duration. Afterwards, the input sequence is divided in three phases. First, during phase G(2), a stimulus is presented to the input map in a random location A. After a while, this stimulus disappears, and the input map remains turned off for a fixed duration (phase G(3)). Finally, during phase G(4), a new stimulus appears at a location $B \neq A$ and remains displayed on the input map until both output maps show the correct stimulus location and identity, or a timeout is reached. The duration of phase G(4) during a successful trial determines the speed of attention shift to the stimulus at location B.

*Description of the
Gap condition*

In this condition, neurons start at rest, and should remain so during phase G(1), for the same reasons as in the Neutral condition. Similarly, during phase G(2), output maps are expected to converge to the correct spatial and categorical activations to reflect the input presented at A, and sustain that activity until phase G(3). Once input activity ceases at the start of phase G(3), the network's overall activity is expected to gradually decrease, until phase G(4). During phase G(4),

*Discussion of the
Gap condition*

Figure 4.2: Organisation of the Gap task condition in the simulated reflex attention shift experiment

input is expected to drive the emergence of a different pair of outputs than during G(2). However, the remaining activity from past input A may temporarily hinder convergence to category and spatial output representations of input B. Furthermore, the accommodation and hysteresis changes driven by channelopathy should affect that convergence delay. This effect of the model of channelopathy are of primary interest.

4.2.3 The Overlap condition

The Overlap condition, shown in Figure 4.3 is the second task meant to capture an attention shift process. In this task, the target appears while the distractor is still present.

Phases O(1) and O(2) of the Overlap condition are the same as G(1) and G(2) in the Gap condition. However, phase O(3) consists in the simultaneous display on the input map of the first stimulus at location A, and of a second stimulus at a different location B. After a short while, in the last phase of the Overlap condition, the initial stimulus disappears, and the second stimulus remains displayed at location B (phase O(4)). This last phase is finished when the output maps display the correct location and identity of the second stimulus. Like with other conditions, invalid outputs are those that have not converged to the correct identity and location within the imparted time. The total duration of phases O(3) and O(4) determines the duration of reflex attention shift in the overlap condition, as the second stimulus appears at the start of O(3).

*Description of the
Overlap condition*

Figure 4.3: Organisation of the Overlap task condition in the simulated reflex attention shift experiment

The two first stages of the Overlap condition are the same as in the Gap condition; the same behaviour is expected. However, O(3), in which two stimuli are simultaneously active at A and at B, is very different from G(3), without input. Whereas overall activity should gradually decrease during G(3), it should slightly increase in O(3). Indeed, the goal of the Overlap condition is to examine reflex attention shift towards the new stimulus B in the presence of the existing detractor A. As the stimulus at A has already been displayed for the duration of O(2) when O(3) starts, the units that collectively represent it in each map of the spatial and object streams should be accommodated, allowing for the emergence of the representation of the new stimulus at B and the relative fading of that at A. Hence, during phase O(4) with only the newer stimulus displayed, the attention shift may already be underway. As the effects of autism channelopathy on the tradeoff between focusing attention in one location and shifting attention to a new one are the main focus of the experiment, the effect of channelopathy on total attention shift time is measured. The emergent attention hypothesis implemented by unit accommodation and the dual-pathways vision model are the implementation context of this study of the focus/shift tradeoff; they are assumed sufficiently representative of the biological reality.

*Discussion of the
Overlap condition*

4.3 MODEL ARCHITECTURE

4.3.1 *Model of reflex visual attention*

*The emergent model
(without explicit
saliency map) is
chosen for its
biological
plausibility.*

In Chapter 3, I review models of bottom-up visual attention (pp. 55–78). As was then noted, the saliency map hypothesis (p. 57) is a robust proposal that has the advantage of giving a clear picture of how bottom-up and top-down attention can interact. The emergent hypothesis of visual attention presented on page 62 is a more biologically plausible model of bottom-up attention, although without a clear mechanism for the integration of top-down attention. But, as I only take bottom-up attention into consideration, the use of the emergent model is more justified than that of the model of the explicit saliency map hypothesis.

4.3.2 *Implementation framework*

There are numerous software environment used in computational neuroscience for the simulation of biological neural networks. Table B.1 of Appendix B presents a summary review of some of the most popular freely available software environments. I reviewed them individually and tried implementing the model with some of the packages listed in the table.

I decided to implement and run EVAC using the Emergent environment.⁷¹ Emergent is the successor of PDP++, a piece of neural simulation software originally developed by McClelland and Rumelhart [156]. As such, the Emergent reference book is the latest handbook for PDP++ [1].

*Related models
implemented using
Emergent*

One reason why Emergent is more appropriate for the implementation of the proposed model is that the reference book [1] contains simplified models of visual attention. Although these models are much smaller in scale and do not feature accommodation-based reflex attention shifts, they provide a good basis for the incremental development of the larger-scale and more complex EVAC model.

Note that, on pages 269–272 of [1], the authors also discussed a more complex model of visual attention that appear much more similar to the EVAC model presented in this dissertation. The first difference is that the model of O'Reilly & Munakata, simulated using PDP++, often fails to perform attention shifts from one object to the other, in what would be similar in their tasks models to the Overlap condition of EVAC.⁷² The second difference lies in the structure of their network, which is simpler, with a 2-layer spatial pathway connected laterally to V_1 and V_2 . Their model is implemented in PDP++, and has not been successfully ported to Emergent.

⁷¹ Emergent version 5.0.2, 32 bit, available freely on grey.colourado.edu

⁷² See authors' notes at grey.colourado.edu/CompCogNeuro

Emergent offers the possibility to work at the appropriate level of abstraction for the EVAC model of reflex attention. The neuron models are Hodgkin-Huxley point units and Emergent gives the possibility to add additional channels that model accommodation and hysteresis, mechanisms central to the EVAC model of attention shift.

*Advantages of
Emergent for the
simulation of EVAC*

The major drawback of using Emergent is its relative isolation from other modelling software. First, model description files are not implemented in or exportable to an open standard such as NeuroML or NineML. Second, an Emergent computational experiment is scripted in an esoteric language called C-Super-Script, used only by Emergent.

*Limitations of
Emergent*

Despite these limitations, I have found that alternative software at the same level of abstraction do not cover my modelling needs, often by lack of appropriate neuron models to simulate accommodation and hysteresis. Thus, Emergent stands out as the most appropriate modelling and simulation environment.

4.3.3 Neural unit architecture

The point neuron is a conductance-based model of biological neurons that abstracts the complex electrical state of the neuron into a vector of numbers aggregating the values of its ionic concentrations and trans-membrane ion flows.⁷³ I choose this type of neuron model because it entails a level of abstraction that suits the goals of the model: the Hodgkin-Huxley equations used provide sufficient modelling power to represent channelopathy, while keeping the computational cost of the simulation manageable. The neuron model I use comprises of five types of channels. In equations, literal symbols that relate to a particular channel are subscripted by a letter that identifies the channel, or by α for a generic expression applicable to several channels.

In this section, the computations performed by the point units of the model of emergent visual attention under channelopathy (EVAC) are thoroughly explained in their mathematical details. The functioning of ionic membrane channels approximation is shown for each type of channel (Equations 4.2 to 4.5). The resulting equations constitute the basis to build the model of ionic current (Equation 4.1), and the dynamics of trans-membrane voltage. Once the equations defining membrane voltage are fully specified (Equation 4.6), the output of the unit can be directly computed 4.7, as in this model, membrane voltage is the only factor of neuron output. Units in the simulation are arbitrary, though they conserve a linear relationship to physical quantities, which allows for relative quantities to remain comparable to experimental clinical results.

*Summary of the
section*

⁷³ Please refer to Section A.1 in Appendix A for a detailed introduction to the principles of astructural electrical models.

All symbols are described and defined as they appear. They often correspond to the symbols used in the reference book [1]. To further help readability, Table 4.1 is provided as an additional reference.

Table 4.1: Symbols used in the equations describing computations performed by units.

x	Vector input to unit (before applying weights)
β/N	Unit bias weight
y^t	Scalar rate-coded unit output
α_k	Expected activity of projection k
n_p	Total number of projections to a unit
dt_{net}	Time-integration constant for the excitatory conductance update
I_{net}^t	Net current for all channels
V_m^t	Membrane potential of the unit
Θ	Voltage threshold to generate an action potential (AP)
E_a	Reversal potential for accommodation channels
E_e	Reversal potential for excitatory channels
E_i	Reversal potential for inhibitory channels
E_h	Reversal potential for hysteresis channels
E_l	Reversal potential for leak channels
g_a^t	Net input through accommodation channel
g_e^t	Net input through excitatory channel
g_e^*	the excitatory input deprived from the bias input β
g_{ek}^t	Excitatory conductance of projection k
g_i^t	Net input through inhibitory channel
$g_i^\Theta(l)$	inhibitory conductance of unit l at the threshold Θ
g_h^t	Net input through hysteresis channel
g_l^t	Net input through leak channel
$\bar{g}_a$	Maximal conductance of accommodation channel
$\bar{g}_e$	Maximal conductance of excitatory channel
$\bar{g}_i$	Maximal conductance of inhibitory channel
$\bar{g}_h$	Maximal conductance of hysteresis channel
$\bar{g}_l$	Maximal conductance of leak channel

continued on next page

continued from previous page

V_a^t	Voltage across accommodation channel
V_e^t	Voltage across excitatory channel
V_i^t	Voltage across inhibitory channel
V_h^t	Voltage across hysteresis channel
V_l^t	Voltage across leak channel
I_a^t	Current through accommodation channel
I_e^t	Current through excitatory channel
I_i^t	Current through inhibitory channel
I_h^t	Current through hysteresis channel
I_l^t	Current through leak channel
dt_{g_a}	Reactivity constant for the accommodation channel conductance
dt_{g_h}	Reactivity constant for the hysteresis channel conductance
b_a	Basis variable for accommodation (accounting for the recent activity of the unit)
b_h	Basis variable for hysteresis (accounting for the recent activity of the unit)
$dt_{b_a,inc}$	Reactivity parameter for an increasing accommodation basis variable
$dt_{b_a,dec}$	Reactivity parameter for a decreasing accommodation basis variable
$dt_{b_h,inc}$	Reactivity parameter for an increasing hysteresis basis variable
$dt_{b_h,dec}$	Reactivity parameter for a decreasing hysteresis basis variable
$\Theta_{a,a}$	Unit accommodation activation threshold (compared to b_a)
$\Theta_{a,d}$	Unit accommodation deactivation threshold (compared to b_a)
$\Theta_{h,a}$	Unit hysteresis activation threshold (compared to b_h)
$\Theta_{h,d}$	Unit hysteresis deactivation threshold (compared to b_h)
α'	Function of the expected layer activity level of the sending layer, as normalisation factor in the learning rule
$\hat{w}$	Non-linearly saturating weights (logistic transform)
γ	Steepness of the sigmoid in the expression of $\hat{w}$

In the Hodgkin-Huxley model used in the simulation, three parameters model the state of each trans-membrane ionic channel α .

Reversal potential
 E_α

PARAMETER E_α represents the reversal potential for the ion carried by channel α . This is the difference of electrical potential between the inside and the outside of the nerve cell at rest when the trans-membrane diffusion force of this ion is exactly countered by the electric force. The relative concentrations of ions on either side of the cytoplasmic membrane are considered constant because the currents are small and as biological mechanisms exist to maintain these concentrations. This first parameter is a static property of the cell.

Proportion g_α^t of
open channels

PARAMETER g_α^t represent the proportion of the total number of channels α that are open. This directly and linearly depends on the excitatory input, and is by definition within $[0, 1]$. g_α^t is therefore practically confounded with the unit's *net input*, as used in a traditional artificial neural network (ANN).

Maximal
conductance $\bar{g}_\alpha$

PARAMETER $\bar{g}_\alpha$ indicates the total conductance for α when all of the channels for α are simultaneously open. The product $g_\alpha^t \bar{g}_\alpha$ consequently represents the conductance for channel α at time t .

Channel current I_α^t

At timestep t , the current I for some channel α is calculated using Ohm's law, as the conductance for that channel at t multiplied by the potential for that channel at t . The potential V_α^t for α at t depends on the membrane potential at t (V_m^t) and on the equilibrium potential of α denoted E_α ; $V_\alpha^t = V_m^t - E_\alpha$. This gives Equation 4.1:

$$I_\alpha^t = g_\alpha^t \bar{g}_\alpha (V_m^t - E_\alpha) \quad (4.1)$$

The above expression applies to all five α channels and thus models all ionic currents of the point neuron. The following describes these five channels.

Leak channel l

$\alpha = l$: The *leak* parameters model the constant potassium leak channels that the membrane of all neurons exhibits. The reversal potential of the leak channel is $E_l = 0.15$, its maximal conductance is $\bar{g}_l = 0.1$ and is always reached, so the proportion of open channels $g_l = 1$ at all times.

Excitatory input
channel e

$\alpha = e$: The *excitatory* input synaptic channels let Na^+ enter the cell when glutamate released by the presynaptic cell binds to the synaptic receptor. Hence, the point unit model of the neuron lets g_e^t depend on the total current excitatory input. The arbitrary voltage and conductance units of the simulation are chosen such as $E_e = 1$ in the default case and $\bar{g}_e = 1$. The expression of the share of open excitatory input channels g_e^t is derived further below.

$\alpha = i$: The *inhibitory* channels are most often GABA-sensitive receptors that let Cl^- ions in to drive back the membrane potential towards the resting potential, as the reversal potential is the same as the membrane potential, -70mV (0.15 in simulation units). It is possible to explicitly model inhibitory units and result in very similar expressions for g_i^t as for g_e^t (see Equation 4.3), but it is usually not necessary as inhibition is mostly a local network cooldown process. There is no evidence of long-range inhibitory projections in the cortex, but of local inhibitory interneurons that receive co-lateral inputs from neighbouring neurons and long-range excitatory projections. Hence, to save computational resources, it is common to integrate the whole inhibitory substrate into a layer-wide inhibitory algorithm that functionally reflects the effect of inhibitory interneurons. The k-WTA algorithm is used in Emergent for inhibition. Section 4.3.4, and in particular Equation 4.11, give detailed explanations on the functioning of k-WTA inhibition in Emergent.

Inhibitory channel i

$\alpha = a$: Together with hysteresis (subscript h), *accommodation* allows the neuron to temporally integrate its state. The accommodation parameters are an abstraction of all ionic channels that are sensitive membrane potentials and ionic concentrations indicative of electrical activity, and which opening results in re-polarisation, and hence in inhibition. The mechanism models neuronal fatigue. The default value of the resting potential $E_a = 0$ follows from the expected behaviour of accommodation to bring back the membrane potential to the resting value, while the value of $\bar{g}_a = \frac{1}{2}$ is empirically determined. Along the same line, the expression of g_a^t (Equation 4.4, page 92) is a piecewise linear expression of its previous value that increases when neuron activity as modelled by b_a^t (Equation 4.5) rises above a given threshold Θ_a , and that decreases when decreasing below another lower threshold Θ_d .

Accommodation channel a

$\alpha = h$: *Hysteresis* is the reciprocal of accommodation: it designates the excitatory opening of channels in response to cell depolarisation, even in the absence of immediate synaptic input. This phenomenon does not conflict with accommodation because it operates on shorter time scales. The equations involved are formally the same, but the default dt_{b_h} parameter is larger than dt_{b_a} , resulting in the time scale differences observed *in vivo*. Other parameters are also changed to match observed behaviour (see [1, p. 461]).

Hysteresis channel h

For all channels, E_α and $\bar{g}_\alpha$ remain constant throughout each simulation. The leak channels do not adjust their opening rate, so typically $g_l^t = 1$. For other channels, g_α^t is variable. The precise update

equation depends on the channel α considered, as ionic channels are sensitive to the different factors described above.

Projections

Before giving the update equation of the excitatory input channel, it is necessary to describe the organisation of the unit inputs into *projections*. A projection consists in the conceptual grouping of inputs from the same cortical region. Projections are useful in EVAC because activity levels of inputs from different regions can greatly differ. With only one projection, the net input to the unit is equal to the average of normalised inputs: $g_e^t = \langle x^t w \rangle$. Within a projection k , the same calculation leads to the average input of the projection. Once scaled to correct for its expected activity α_k , this average is the connection's contribution to the total input, called the excitatory conductance of the projection $g_{e_k}^t$:

*Excitatory
conductance $g_{e_k}^t$ of
a projection*

$$g_{e_k}^t = \frac{1}{\alpha_k} \langle x^t w \rangle_k \quad (4.2)$$

Then, the total input to the unit is the average of the excitatory conductances of all n_p projections $\frac{1}{n_p} \sum_k g_{e_k}$. To update the excitatory conductance of the whole unit, an arbitrary term $dt_{\text{net}} \in]0, 1[$ is used that scales g_e^{t+1} to a value between the previous value g_e^t and the net input.

Additionally, the effect of a unit bias weight term β that aggregates all differences in excitability that arise between any two neurons is integrated [1, section 2.5.1, page 43].

The resulting expression used to update the net input to a unit is shown in Equation 4.3.

*Net excitatory input
 g_e*

$$g_e^{t+1} = (1 - dt_{\text{net}}) g_e^t + dt_{\text{net}} \left(\frac{1}{n_p} \sum_k g_{e_k}^{t+1} + \frac{\beta}{N} \right) \quad (4.3)$$

The update equation of the conductance of the accommodation and hysteresis channels have the same form, shown in Equation 4.4:

*Accommodation and
hysteresis inputs g_a
and g_h*

$$g_{\alpha}^{t+1} = \begin{cases} g_{\alpha}^t + dt_{g_{\alpha}} (1 - g_{\alpha}^t) & \text{when } b_{\alpha}^{t+1} > \Theta_{\alpha,a} \\ g_{\alpha}^t + dt_{g_{\alpha}} (0 - g_{\alpha}^t) & \text{when } b_{\alpha}^{t+1} < \Theta_{\alpha,d} \end{cases} \quad (4.4)$$

where $dt_{g_{\alpha}}$ is a reactivity constant, and the variable b_{α} (introduced below) is compared to an activation threshold $\Theta_{\alpha,a}$ or a deactivation threshold $\Theta_{\alpha,d}$.

The above update equations constitute the generic form for both channels. The instantiated equations only vary in the value of their parameters because the behaviour of both the accommodation and hysteresis channels is that of *activation-gated* channels. In the term "activation-gated", the meaning of activation is akin to the temporally summed electrical activity of the cell, and is represented in Equation 4.4 by the term b_{α}^t , the average output of the unit in its recent past

(called *basis variable*). The update equation of the conductance of hysteresis and accommodation (Equation 4.4) is hence naturally defined piece-wise, in terms of this estimate of recent unit activity b_α^t . b_α^t is defined by equation 4.5:

$$b_\alpha^{t+1} = b_\alpha^t + dt_{b_\alpha} (y^{t+1} - b_\alpha^t) \quad (4.5)$$

*Recent unit activity
 b_a and b_h*

where y^t is the rate-coded output of the unit, and the term dt_{b_α} adjusts the reactivity of b_α^t , hence reflecting how much time is integrated. The parameter dt_{b_α} can be differently valued if the basis variable is increasing or decreasing, leading to two parameters $dt_{b_\alpha, inc}$ and $dt_{b_\alpha, dec}$.

In Equation 4.4, the conductance will rise or decrease at a speed that depends on the value of dt_{g_α} . When the activity b_α^t is greater than the activation threshold $\Theta_{\alpha, a}$, the conductance will increase towards a ceiling of 1, and inversely decrease towards 0 when $b_\alpha^t < \Theta_{\alpha, d}$, where $\Theta_{\alpha, d}$ is the deactivation threshold of the channel.

As a result, the relatively longer setting delay of accommodation compared to hysteresis is implemented by varying the time integration parameters dt_{b_α} and dt_{g_α} present in Equations 4.4 and 4.5.

*Difference between
 g_a and g_h*

Having determined how to compute all dynamic channels' conductances g_α , the current flow at the cell membrane can be calculated at each timestep using Equation 4.1.

From there, the trans-membrane electrical potential V_m is updated at each timestep following Equation 4.6:

$$V_m^{t+1} = V_m^t - dt_{vm} \sum_{\alpha=e,i,l,a,h} I_\alpha^t \quad (4.6)$$

*Membrane potential
update*

where dt_{vm} is a time-averaging parameter slowing down the change in membrane potential [1, page 37], similarly to dt_{net} in Equation 4.3. I_α^t is the total electrical current going through the membrane at time t , defined in terms of channel properties and dependent on the membrane voltage (Equation 4.1, page 90).

From Equation 4.6, the integrate-and-fire mechanism of the neuron can be modelled either by Equation 4.7 that produces an approximation of the firing rate, or by an actual discrete action potential-producing function. The equation approximating the activation rate y^t is denoted in Equation 4.7.

$$y^t = \frac{\gamma [V_m^t - \Theta]_+}{\gamma [V_m^t - \Theta]_+ + 1} * \mathcal{N}\left(x, \frac{1}{200}\right) \quad (4.7)$$

*Instantaneous firing
rate y*

where Θ is the membrane threshold for action potential generation, V_m^t is the trans-membrane electrical potential defined in Equation 4.6, and $\mathcal{N}\left(x, \frac{1}{200}\right)$ is a Gaussian noise kernel with variance 0.005. Equation 4.7 is the positive half of a sigmoidal activation function with a gain parameter γ (defaulting to 600), like often encountered in artificial neural networks. In the present case, it is convoluted with a

Gaussian function in order to better simulate the effects on instantaneous firing rate of the activation noise observed with live nerves. The parameter values have been determined by O'Reilly & Munakata [1].

4.3.4 *Layer structure and connectivity*

The previous section detailed the mathematical basis of EVAC at the lowest level of individual neurons. The present section presents the architecture and processing mechanisms of the modelled network, which aims at imitating some aspects of human visual pathways.

Overview

The processing mechanism described below is that of the model developed. In reality, the human brain transmits richer visual signals, as was expounded in the previous chapter, and along more complex pathways. The EVAC model only retains those mechanisms essential for object-based reflex attention tasks. These mechanisms are based on the relevant biological visual pathways, following the general principles of cortical connectivity:

*Guiding principles
of the model*

- afferent connections are accompanied by symmetrical efferent connections, effectively making networks recurrent [157],
- the structure of maps and intra-map connectivity follow similar principles all across the cortex (see page 98),
- inhibitory competition dynamics regulate intra-layer activity levels (see page 99), and
- learning is implemented as a biologically plausible mix of Hebbian learning and two-phases error propagation (see page 103).

The model of attention was build on previous simpler models of vision and of attention from O'Reilly & Munakata [35]. The connectivity follows the general principle of layered cortical maps with receptive fields (RF) connectivity and feedback. The specific connectivity parameters and the general architecture of EVAC stem from the literature [1]. Figure 4.4 presents an overview of the network. The input layer (at the bottom) accepts contrast images (greyscale). The LGN layer combines the processing of centre-surround ganglion cells of the retina and of the thalamus, letting V_1 cells respond to oriented edges with a polarity preference. V_2 cells respond to more complex features, while remaining spatially representative. The dorsal stream (left) then specialises in responding to the spatial location of the stimulus, while the ventral stream (right) learns to respond to the category of the input. All inter-layer connections are excitatory (arrowheads) and reciprocal starting from V_1 , letting higher layers influence feed-forward

processing during learning and testing. All hidden and output layers recurrently inhibit themselves (k-WTA intra-layer inhibition), thereby regulating their own maximum activity levels.

The black-and-white input picture is imposed onto a 48×48 input layer, from which lateral geniculate nucleus (LGN) units take their input. In practice, this LGN layer combines the processing of centre-surround ganglion cells of the retina and of the lateral geniculate nucleus by representing the single input image on two LGN maps, one meant to represent the processing of off-centre, on-surround ganglion cells, and the other, for off-centre, on-surround cells. Thus, the processing performed by the retinal maps and LGN in humans (Figure 3.2 page 54) is encompassed in EVAC by representing the input map as two complimentary pictures meant to highlight contrast edges. Specifically, as inputs are composed of 1-unit wide lines, the map of on-centre/off-surround feature detectors merely replicates the input map, while the map of off-centre/on-surround units creates a trace that surrounds and complements the input. Figure 4.5 illustrates the processing performed by these two complementary maps of the LGN layer.

*Retinal input and
LGN model*

As a result, this sub-cortical pathway to V_1 pre-processes the visual signal by enhancing contrast areas and by encoding the signal as activations of on-centre/off-surround and off-centre/on-surround detectors. This simple processing is very limited as compared to the actual processing of the retina and LGN: many temporal (motion) and static features are processed in the equivalent biological neural tissues that are not taken into account in the model's LGN. But one of the interests of modelling is to breakdown functionality into conceptual units for easier understanding, and EVAC is limited to the processing of one feature: contrast edge orientation.

The first visual cortex (V_1) receives inputs from the lateral geniculate nucleus (LGN). Hence, the model's V_1 units (corresponding to groups of neurons in the V_1 area in Figure 3.2) are sensitive to the orientation, position and polarity of contrast edges. This is a minimal set of features a human may use to recognise an object,⁷⁴ as shown by the good performance in recognising objects from their monochrome outlined shape. These V_1 units are organised in cortical minicolumns, topographically adjacent ones covering adjacent regions in input space with some amount of receptive field (RF) overlap. Assuming a uniform distribution of input over the visual field, all the minicolumns can be set up to share a common set of weights. V_1 is modelled by a 24×24 minicolumns map of 8 units per minicolumn, with weights that are pre-computed to selectively respond to a specific orientation and polarity in the RF. Over the 8 units of each V_1 minicolumns, 4 respond to edges oriented at 0° , 45° , 90° and 135° over RF of 6×6

*Model of the
primary visual
cortex*

⁷⁴ Among the features ignored by EVAC, real V_1 units are shown to be sensitive to motion direction, colour, ocular disparity, and texture.

Figure 4.4: Overview of the structure of the visual attention network.

Figure 4.5: Illustration of the processing performed by on-centre and off-centre LGN maps in EVAC

on-centre LGN units, and the 4 other units of the V_1 minicolumn respond to the same angles, but over 6×6 RF of the off-centre LGN map. This simplification is functionally equivalent to the result of learning these orientations with a larger number of neurons per minicolumn, and saves computation time and memory.

This edge detector is connected to a V_2 model of 12×12 minicolumns of 36 units each; the receptive fields are of 16 V_1 minicolumns. As a feature detector, V_2 is peculiar as it is the last map of the common visual pathway, thus is encouraged to learn more global and more abstract features at the same time. This V_1 -to- V_2 connectivity is a straightforward translation of the biological pathway (Figure 3.2 page 54).

V_2 is connected to a simple ventral pathway starting with 6×6 minicolumns V_4 map with 81 units per minicolumn. The RF of these minicolumns on V_2 is 4×4 minicolumns with a 50% overlap. The output of V_4 is fully connected to a simple map of 12×12 units labelled IT, an intermediary representation which output is input of a semantic output layer. That output layer of 6 units is trained to associate one unit per object identity. It is meant to represent very roughly any higher level abstraction of the input (past the inferotemporal cortex); this abstract knowledge is learnt in a supervised manner, so as during the running phase, the semantic output layer shows the identity of the object. The biological equivalent of this V_4 to semantic output pathway does involve successive cortical maps sensitive to shapes of increasing complexity, but the equivalence is mostly qualitative in high areas. For instance, the specificities of the human IT area (cf. Figure 3.2) are not really represented by the IT layer in EVAC, as these details are out of the scope of the model.

V_2 is also connected to a dorsal stream. V_2 feeds its output to a 6×6 minicolumns spatial map (S_1) where each 4-units minicolumn is

*Model of the
secondary visual
cortex*

*Ventral stream:
 V_2 - V_4 -IT- S_{out}*

Figure 4.6: The six symbols that the network learns

*Dorsal stream:
V2-S1-S2*

connected to 4-by-4 minicolumns region of V_1 , with a 2 minicolumns neighbour overlap. Finally, S_1 feeds forward to the spatial output layer S_2 with a 8x8 units RF. S_1 and S_2 do not exactly correspond to specific post- V_2 maps of the dorsal pathway, but rather translate the general principle of chained maps of increasing abstraction.

*Back-connections
are systematic.*

Backward connections run back through both streams up to V_1 . Such feedback projections are also a typical pattern of connectivity throughout the cerebral cortex [158, page 329]. The S_1 layer receives feedback from the spatial output layer, and in parallel, V_4 from the semantic output layer through IT. V_2 , which is forward-connected to S_1 and V_4 , receives feedback from these layers, thus has the unique role of integrating them. Backwards connectivity is omnipresent in the cortex; it is modelled here because it is assumed to be both a primary learning mechanism, used for the back-propagation of error signals, and to have a central role in the amplification of changes in relative activation due to neuronal fatigue.

Toroidal maps

All neural maps are typographically toroidal to avoid the effects of under-connectivity at the edge. Indeed, *in vivo*, networks are of much larger scale, and maps are regions of a continuous neural substrate. Consequently, edge effects are not observed and should be avoided in simulation. Toroidal maps, where units at opposite borders are connected as if they were adjacent, prevent these undesirable effects.

The glyph-like images that the network learns to identify are presented in Figure 4.6. They are unique at least with regards to the number or edges of a given orientation, which should suffice for the network to learn and classify them in the ventral stream.

Cortical networks structure

Cortical maps are physically six-layered, and conceptually three-layered. The term “layer” will henceforth mean an abstract layer if not otherwise qualified. The input, hidden and output layers of a cortical map respectively correspond to physical layers 4, 2 and 3, 5 and

6.⁷⁵ These three stages (marked as *input (Phys. layer 4)*, *hidden (Phys. layer 2, 3)*, and *output (Phys. layer 5, 6)* in Figure 4.7) are replicated horizontally, and it is their horizontal combination that translates into the multi-layered architecture used in the model of emergent visual attention under channelopathy (EVAC). This is visually illustrated in the highlighted stages in Figure 4.7 (B). This *tiered organisation* is very consistent across the whole cortex, although the preponderance of each type of layer varies according to the use of the map. V₁ maps have a much thicker physical layer 4, for instance, as V₁ is an input area and physical layer 4 is the main input layer. The connectivity between and within layers is illustrated in Figure 4.7.

The physical organisation of logical layers in the cortex

Part (A) of Figure 4.7 is a schematic representation of the layered organisation of a cortical map. The circular connectors represent inhibition, and the triangular ones are excitatory. Part (B) shows how such structures integrate to form the connectivity between cortical regions. In the diagram, some abstract cortical regions a, b and c are chained into a processing pathway starting at input map a, passing through the “hidden” map b, and ending at output layer c. The position of a given map relative to cortical input and output correlates with the development of the corresponding layer within the map.

Inter-map connectivity in the cortex

Another principle of cortical networks is the *bidirectional character of map connectivity* [158], hence the feedback connections from b to a and c to b on Figure 4.7. This is a necessary condition for map development, as error-driven learning has to imply a form of error backpropagation, and a primordial mechanism of the regulation of reflex of visual attention.

It is to be noted that inter-layer connectivity is assured by the hidden layers and not by output-to-input connections. In result, cortical maps that are functionally hidden⁷⁶ have their physical layers 2 and 3 (hidden layers) most developed and with the most important role. For instance, the “hidden” map b of the diagram is mostly composed of layers 2 and 3, as inter-map connectivity is mostly assured by physical layers 2 and 3. The same goes for input layers in the input maps and output layers in the output maps.

Hidden layers of the model correspond to physical layers 2 and 3.

Modelling of lateral intra-map connectivity

Within each map, the lateral connectivity is inhibitory on long neighbourhood range, and excitatory on shorter range. Lateral excitatory connections are represented by Gaussian-weighted connections to the topographical neighbours, while the more global inhibitory mechanisms are not explicitly modelled using inhibitory interneurons, but by the application of a map-level algorithm, the k-Winners-Take-All inhibition scheme. This form of inhibition is an important compon-

Summary of intra-map connectivity

⁷⁵ Layer 1 contains few actual neurons.

⁷⁶ “Functionally hidden” means not primarily receiving thalamic input or producing motor output.

Figure 4.7: Abstraction of cortical intra and inter-map connectivity (adapted from [1])

ent of the activity modulation mechanisms in place in the model, as explained below.

Throughout the cortex, the short-term regulation of the total output of neurons – individually and in populations – is ensured by several mechanisms: the leak current of each neuron, the phenomenon of short-term accommodation (neuronal fatigue) and hysteresis, and the action of inhibitory interneurons. The leak current is a static property of each unit. The leak current, being a passive membrane ion channel, is modelled by the elements indexed by the l subscript in the membrane potential update equation, and accommodation and hysteresis have the subscript a and h .⁷⁷ The action of interneurons is a necessary complement to these membrane mechanisms as it provides feed-forward and feedback inhibition of the units in accordance to the activity level of the map. Schematically, inhibitory interneurons respond to the activity level of either the preceding layer (feed-forward inhibition) or of the layer they modulate (feedback inhibition). These mechanisms are of graded dynamism: leak current is membrane-local and constant, accommodation mechanisms are unit-local and dependent on past input to the unit, and interneurons-mediated inhibition integrates recent activity of part of the network. Consequently, the complexity of modelling of these mechanisms increases with their dynamism. While channel mechanisms can be integrated in the membrane potential equation, inter-neuronal inhibition is a layer-wide phenomenon that needs to be taken into account at the level of layer activation. This layer-wide inhibition is the objective of the k -WTA algorithm.

Recap of unit-level activity regulation mechanisms

Inhibition via interneurons is a map-level complement of unit-level activity regulation mechanisms.

The k -WTA algorithm is directly derived from classical SOM learning algorithm, widely used in neural modelling.⁷⁸ In maps where units are functionally clustered in several minicolumns, the algorithm is applied two times. It is first applied among all units in a minicolumn and then to all units in a layer.

In its simplest form, the k -WTA function computes a common inhibitory channel opening g_i for all units of the map so that the k units with the highest excitatory input g_e are the only ones above threshold. The paragraphs that follow derive the mathematical expression of g_i and discuss simple and more elaborate inhibition schemes.

Principle of the k -WTA inhibition algorithm

Recalling Equation 4.1: $I_\alpha^t = g_\alpha^t \bar{g}_\alpha (V_m^t - E_\alpha)$ that defines the current at channel α , the net current I_{net}^t for all channels can be written as:

$$I_{\text{net}}^t = \sum_{\alpha} (g_\alpha^t \bar{g}_\alpha (V_m^t - E_\alpha)) \quad (4.8)$$

Let Θ a threshold membrane potential value, $g_i^\Theta(l)$ the inhibitory conductance at the threshold Θ of unit l , which is the inhibitory conductance necessary to bring unit l to the AP-production threshold

⁷⁷ Detailed specifications of channels can be found pp. 90 – 93

⁷⁸ I review of SOM-based modelling in Chapter 3, page 66

given its current excitatory input, and g_e^* the excitatory input deprived from the bias input β (see Equation 4.3).

Then, setting the net current to null and the membrane potential to the threshold $V_m = \Theta$ in Equation 4.8 leads to the expression of the inhibitory conductance at the threshold g_i^Θ in Equation 4.9:

$$g_i^\Theta = \frac{\overbrace{\sum_{\alpha \in \{l, a, h\}} (g_\alpha \bar{g}_\alpha (\Theta - E_\alpha))}^{\substack{\text{net current for l, a, and h} \\ I_{\text{net}}^t - I_i^t \text{ (without bias)}}} + \overbrace{g_e^* \bar{g}_e (\Theta - E_\alpha)}^{\text{net excitatory current}}}{\underbrace{E_i - \Theta}_{\text{potential between inhibitory equilibrium and AP threshold}}} \quad (4.9)$$

Inhibitory conductance at threshold g_i^Θ

From that point, two ways of computing the proportion of open inhibitory channels g_i are proposed:

- A simple mechanism consists in ordering units by their current level of excitatory input g_e and taking g_i between the inhibitory conductance at the threshold of unit k . Unit k is to be the weakest activated unit still included in the k winners, and unit $k + 1$, the strongest activated unit of the losers. A parameter q , defaulting to $\frac{1}{4}$, determines how close to $g_i^\Theta(k + 1)$ and $g_i^\Theta(k)$ the value of g_i is taken:

$$g_i = g_i^\Theta(k + 1) + q (g_i^\Theta(k) - g_i^\Theta(k + 1)) \quad (4.10)$$

- A more sophisticated calculation consists in taking the distributions of the top k units and of the remaining units by placing g_i between the respective averages of g_i^Θ for the top k units and bottom $N - k$ units, as in Equation 4.11:

$$g_i = \langle g_i^\Theta \rangle_{N-k} + q (\langle g_i^\Theta \rangle_k - \langle g_i^\Theta \rangle_{N-k}) \quad (4.11)$$

where the average of the inhibitory conductances at the threshold of the top k units is:

$$\langle g_i^\Theta \rangle_k = \frac{1}{k} \sum_{j=1}^k g_i^\Theta(j)$$

and for the bottom $N - k$ units:

$$\langle g_i^\Theta \rangle_{N-k} = \frac{1}{N-k} \sum_{j=k+1}^N g_i^\Theta(j)$$

The use of Equation 4.11 for updating the inhibitory input does not enforce that exactly k units are selected each time, because the distribution of excitation among units in the layer is most often non-linear. Differences in the shape of such distribution can result in more than

Inhibitory input g_i when taken between units k and $k + 1$

Inhibitory input g_i when taken between the average activity of the k winner units and of the $N - k$ losers

k units activated, when the plateau of high excitatory input spreads to more than the top k units. They can also result in less than k units activated, when some of the top k units are in the low activity valley. For the purpose of modelling biology, this is not an issue if the overall activity of the map is sufficiently high [1, page 102]. In consequence, it may be necessary to adjust the expected activity level a_k (as introduced in Equation 4.2), increasing it when the activity level is insufficient. In spite of this inconvenience, this type of average-based update of the proportion of open inhibitory channels g_i gives more flexibility to the learning algorithm, because the sparse representation can make use of the level of map activity.

It is to be noted that the simplification consisting of modelling the effect of inhibitory interneurons at map level is debatable as it assumes that functionally equivalent inhibitory interneurons between autistic and neurotypical subjects, or only allows the modelling of the effects on winners selection of any cell-level disturbance.

Equation 4.11 adds a parameter, but is more powerful.

4.3.5 Biologically plausible learning

The goal of the learning algorithms used is to set the weights of the synaptic connections so as the resulting pattern is likely to model the pattern of synaptic strengths in the corresponding areas in the brain. Consequently, it is not necessary that the method used to set these weights models the development and learning processes that take place in the brain. However, I argue that the method used to set these weights should try to be faithful to its biological counterpart when the latter is known, or be a plausible model of it when it is not known.

Indeed, in Artificial Neural Networks, synaptic form often follows the details of the learning algorithm used. Thus, if I want the synaptic weights to form a network that is likely to reasonably model its cortical analogue, I have to employ a learning algorithm that mimics learning in the cortex. For that, I employ the Leabra algorithm, published by O'Reilly [3] and implemented in Emergent [1]. Additionally, the network is trained using a sure self-organising Hebbian association for features learning in the LGN-to- V_1 projections.⁷⁹

Biologically plausible learning results in synaptic weights more likely to be faithful to biology.

During learning, the input layer is presented with a symbol and both output layers (semantic and spatial) are clamped to the desired value. As the input symbol position and scale are randomised, a separate imperative program is used to determine the desired activation of the spatial output layer.

⁷⁹ As explained in Section 4.3.4, the model's V_1 minicolumns can be minimised into 8 units, covering 4 segments orientations and 2 polarities. In that case, LGN-to- V_1 projection weights are not learnt but pre-defined. This is the case in the final implementation of EVAC.

Self-organised learning

*Summary of
self-organised
learning in EVAC*

The self-organised SOM learning performed by EVAC extracts statistical correlations from the input patterns through re-normalised and contrast-enhanced Hebbian weights update. For hidden (non-clamped) units, once the net input has been computed, the k-winners-take-all (k-WTA) activation algorithm is applied as described page 101. The weights from LGN to V_1 are learnt by Hebbian self-organisation with k-WTA,⁸⁰ giving rise to typical patchy topographic maps.

*k-WTA SOM learning
models
NMDA-mediated
LTP/D.*

Hebbian learning is present in the self-organising aspect of cortical maps. The features learnt by the input units are selected by a blend of the self-organisation mechanism of SOM and by the effect of error-driven learning. The only known mechanisms that support learning in the brain are the calcium-mediated long-term potentiation of the neuron membrane (LTP) and long-term depression (LTD). Primarily, the NMDA-mediated potentiation and depression mechanism is proposed as the most likely contributor to neuronal Hebbian learning.⁸¹ Consequently, the Hebbian associative learning mechanism that is the basis of the formation of self-organising cortical maps is believed to function based on those.

Oja's rule for Hebbian learning has been used to create learning algorithms that can be shown to converge to a representation of the principal components of the training set [159]. Oja's learning rule can be expressed as $\Delta w_{ij} = \epsilon y_j (x_i - y_j w_{ij})$, where i and j index respectively the pre-synaptic and post-synaptic units, and x and y denote input and output. O'Reilly & Munakata [1] proposed a variation of Oja's rule that performs conditional principal components analysis (CPCA) when used in presence of the inhibitory competition mechanics exposed in Section 4.3.4. The basic CPCA weight update equation is:

*Simple CPCA
learning rule*

$$\Delta w_{ij} = \epsilon y_j (x_i - w_{ij}) \quad (4.12)$$

However, the Hebbian learning rule in EVAC is a more complex blend of this basic CPCA weight update with a version of the CPCA weight update that is re-normalised by the sending layer's activity α . This weight update rule is shown in Equation 4.13.

*CPCA rule with
re-normalisation*

$$\Delta w_{ij}^{\text{hebb}} = \epsilon \left(y_i x_i \left(\frac{.5}{\alpha'} - w_{ij} \right) + y_j (1 - x_i) (0 - w_{ij}) \right) \quad (4.13)$$

The parameter α' is defined as a function of α , the expected layer

⁸⁰ The weights in higher layers are set by the supervised Leabra algorithm, which includes elements of Hebbian learning as well.

⁸¹ The N-methyl-D-aspartate (NMDA) channel allows Ca^{2+} to enter the cell when both presynaptic and post-synaptic cells are strongly active, resulting indirectly in an improvement of the excitatory receptors efficacy. It is hypothesised that in the case of different post- and pre-synaptic activities, the NMDA channel allows the entry of fewer calcium ions into the cell, the resulting lower concentration in turn triggering different reactions leading to LTD [1, p. 117].

activity level of the sending layer like introduced in Equation 4.2, of ϵ , a learning rate set heuristically, and of q , a parameter that balances the renormalisation of the layer:

$$\alpha' = \frac{1}{2} - q \left(\frac{1}{2} - \alpha \right) \quad (4.14)$$

When $q = 0$, Equation 4.13 becomes the basic (not re-normalised) CPCA weight update equation: $\Delta w_{ij} = \epsilon y_j (x_i - w_{ij})$. The interest of re-normalising the weight update according to the sending layer activity is to compensate for layers where the most active units still have a low activity, reducing the dynamic range of the weights. The parameter q allows the tuning of the maximal renormalisation (up to the full unit range).

While the linearly learnt w weights are used for update, an additional step can be used when applying the learnt weights to compute the net input to the unit. CPCA learning biases the sensitivity of units to the first principal component of their input. This characteristic can be enhanced by computing the non-linearly saturating weights $\hat{w}$:

$$\hat{w}_{ij} = \frac{1}{1 + \left(\Theta \frac{w_{ij}}{1 - w_{ij}} \right)^{-\gamma}} \quad (4.15)$$

Equation 4.15 results in an effective weight that is related to the actual weights by a sigmoidal function, with a steepness controlled by γ and an offset controlled by Θ .

*Sigmoidal weight
update filter*

This model of neuronal Hebbian learning can be used as such between LGN and V_1 , but is completed by the deeper task or error-driven learning in higher layer. This is a necessity both biologically, as living organisms happen to learn from interaction with the environment, and computationally, as multilayered networks generally can not be trained by the Hebbian learning rule alone.

Error-driven learning

With the Leabra algorithm, the learning of the hidden layers and output layers weights is aimed at being a biologically realistic mixture of Hebbian learning and error-driven learning [3].⁸²

Error-driven learning is a fundamental problem in neural networks, and as such has been given numerous solutions, including the most notable back-propagation algorithm [160]. Although back-propagation is often used in coarse neural simulation, it lacks biological support because it involves mathematical operations that are not thought systematically plausible in biological networks. Instead, the Emergent simulator offers a solution based on the Generalised Recirculation algorithm, or *GeneRec*, introduced in [161]. The GeneRec

⁸² See Equations 4.13 and 4.17a for Hebbian and error-driven learning respectively, and 4.18 for their combination.

Principle of GeneRec

algorithm assumes that task learning involves two successive phases, "minus" and "plus". In the minus phase, the inputs are used to generate an expectation, as the output units run free. In the plus phase, both input and output units are clamped. This plus phase corresponds to the "observation of reality" that leads to an error signal when temporally juxtaposed to the expectation phase.

The supervised learning of GeneRec consists in modifying the weights according to the difference between the activations of the receiving units in each phase, proportionately to the average activation of the sending unit between the two phases (Equation 4.16).

Unidirectional
GeneRec learning
rule

$$\Delta' w_{ij} = \epsilon (y_j^+ - y_j^-) \langle x_i \rangle_{-,+} \quad (4.16)$$

where y_j^+ represents the activation of the post-synaptic unit in the plus phase, and y_j^- , its output during the minus phase. $\langle x_i \rangle_{-,+}$ is the average activation of the pre-synaptic unit during the two phases. The learning rate constant ϵ is arbitrary and empirically chosen.

It can be shown that GeneRec approximates back-propagation [154]. This update mechanism is only valid when connectivity is roughly bidirectional and symmetric, so the average of the weight update in both directions is used to update the weights, leading to Equation 4.17a, which corresponds to the update mechanism of the *Contrastive Hebbian Learning* algorithm.

Symmetrical CHL
rule

$$\Delta w_{ij}^{\text{err}} = \frac{\Delta' w_{ij} + \Delta' w_{ji}}{2} \quad (4.17a)$$

$$= \epsilon (x_i^+ y_j^+ - x_i^- y_j^-) \quad (4.17b)$$

The biological plausibility of this type of error-driven learning is factored by the connectivity requirements, the signals transmitted, and the plausibility of the two-phase activation. The signals transmitted are all local activations, readily available, and the connectivity requirements of rough (non-individual) bidirectionality are supported by observation [1, p161]. O'Reilly & Munakata [1] also provided support for the biological basis of two-phase activation by citing the observation of the P300 electrical cortical wave, associated with the violation of expectancy. In that case, the P300 would correspond to the back-propagating plus-phase activation.

Biological
plausibility of CHL

Another potential issue regarding the realism of such learning concerns the weakening of weights in the case of neuronal false positive. Correlated activity during the minus phase ($x_i^- y_j^- \simeq 1$) and not during the plus phase leading to decrease of synaptic strength seems puzzling. The authors proposed an explanation that is consistent with the non-linear calcium-activated any long-term change in synaptic strength (either LTP or LTD) (LTP/D) mechanism proposed earlier: when $x_i^+ y_j^+ \simeq 1$, the plus and minus phase are temporally close enough to cumulatively raise calcium concentration for LTP to

happen, while when $x_i^+ y_j^+ \simeq 0$ and $x_i^- y_j^- \simeq 1$, the correlated activation in minus phase alone is unable to do so but still sufficient to reach the calcium concentration necessary for LTD. See [1, p. 169-270] for a more detailed discussion.

Combined self-organised and error-driven learning

The weights are updated as a weighted combination of error-driven learning and CPCA task learning in the SOM context. Thus, for a unit indexed in its topological map by the subscripts i and j , Equation 4.18) is obtained:

$$\Delta w_{ij} = \epsilon \left(k^{\text{hebb}} \Delta w_{ij}^{\text{hebb}} + (1 - k^{\text{hebb}}) \Delta w_{ij}^{\text{err}} \right) \quad (4.18)$$

*Full Leabra weights
update rule*

where $\Delta w_{ij}^{\text{hebb}}$ is the associative learning of Equation 4.13 and $\Delta w_{ij}^{\text{err}}$ is the error-driven change in weight defined in Equation 4.17a. k^{hebb} controls the proportional influence of Hebbian association in total learning.

This combined weight update equation, which is central to the Leabra algorithm, is a model of the biological observation of task learning combined with model learning in cortical networks, and in particular of the specific synaptic mechanisms mentioned earlier, as these are omnipresent in the cortex. Functionally, the presence of k -WTA and Hebbian learning adds additional constraints to the error-driven learning. In the vertical error-driven task learning, the deeper the network, the more unstable the propagation of weights in the deep layers, due to the large freedom available for changing the weights. The horizontal k -WTA provides additional regularising constraints to the Hebbian learning: as it drives the network weights update into reflecting the statistical structure of the data, the k -WTA rule improves the generalisation power of the network [1, section 6.4.1, p. 183].

*Functional benefits
of combined
self-organised and
error-driven
learning*

4.3.6 *Learning and testing implementation*

The design of the learning module obeys the general principles of functional programming, albeit C-Super-Script (CSS), the modelling language of Emergent. It is not an intrinsically functional language, but proposes an original multi-paradigm approach combining objects and stateful subroutines. The lower-level programming is object-oriented, and comprises the calling of methods of networks, units, and object representations of data processors. The routines that the user of Emergent must implement are exemplified by a set of standard learning and training routines that can be easily imported into a project. In addition to calling each other, these routines (called programs by Emergent) are often stateful. For instance, the program that uses one exemplar out of all samples of an epoch and presents it to the network until activation settles contains a variable incremented

*Implementation
constraints in
Emergent*

at each call of the program. This is used to index the input samples in the input data table. This sort of stateful programming idiom is what I attempt to eliminate in the design of the learning module in particular, and in software design in general. In the previous example, it means either parameterising the index (integer argument), or parameterising the row of the input data table (datatable row argument). More functional code is a desirable property, as many programming errors lie in the complexity of mutable state.

Figure 4.8 gives an overview of the implementation of the learning and testing routines. The following paragraphs explain the role of each routine, starting from the highest level of control to the individual simulation cycle. In these explanations, I omit to mention any code used for the real-time update of the visual representation of the network, the gathering of statistics at various levels of abstraction, interfacing with the graphical user interface (GUI), and any other side effects or programming details.

*RepeatedExperiment
automates training
and testing.*

The learning programs are conceptually divided by timescale, each program managing a one timescale and only calling the programs from the timescale below. At the highest level, the `RepeatedExperiment` program automates the collection of statistics over numerous runs of the full network training and testing. `RepeatedExperiment` expects the following parameters: the network on which to operate, the training and testing datasets, the random seed, and the number of repetitions. For each repetition, this program calls `LeabraTrain`, collects the statistics of the training, then calls `Test` and writes the training and testing data in one new row of the `RepeatedExperimentOutputData` table.

*LeabraTrain runs
epochs until the
network is trained.*

The `Test` program is a convenience wrapper calling the `Epoch` program in testing mode (parameter switch). The `LeabraTrain` program repeats `Leabra` training (call to the `Epoch` program in training mode with random permutations of the input table) until a defined number of successive epochs with less errors than a threshold is encountered, or, failing that, until a maximum number of epochs. Its arguments are: the network to train, the datatable containing training or testing samples, and some parameters for the two stopping criteria.⁸³

The `Epoch` program accepts as parameters: the network to train, the input datatable, and the type of presentation order – sequential, random permutation, or random with repetition.

*Epoch implicitly
iterates over the
input trials.*

This `Epoch` program calls the `Trial` program in a loop over the input table. `Emergent` makes a specially optimised `NetDataLoop` control structure to iterate over input samples. Efficient permuted, sequential or random access to the the input table and memoised reading of each sample are automatically managed by `NetDataLoop`. Hence, nowhere

⁸³ The stopping criteria parameters are the maximum allowable count of erroneous trials per epoch, how many successive times that higher error count bound must be satisfied, and the maximum number of epochs to run if the error count has not been met earlier.

Figure 4.8: Functional overview of the implementation of the experiment in Emergent

in the programs is there any explicit call to a function reading the input datatable.

The selection of the plus phase to run in training mode is made in Trial.

The Trial program is called by Epoch with a parameter switch that defines whether it is to be run in training or testing mode. In training mode, it performs Leabra training, with its *plus* and *minus* phases (see Section 4.3.5). It means that the Trial program will call the Settle program twice per call when performing training: once without output clamping (*minus* phase), and once with it (*plus* phase). In testing mode, trial will only apply the input once with the Settle program.

Settle updates the simulation until a steady state.

Settle iterates over cycles of updating. At each iteration, it calls a program named Cycle. It stops when the network has settled into a stable state, or after a maximum number of cycles. Parameters include the network and the input data, which determines if the output is clamped or not.

The steady state criterion for stopping is either based on a maximum change in activation or on a maximum activation value. Although the criterion checks are made at each loop by Settle, the parameter values are network properties.

Settle calls Cycle, a one-step simulation routine.

The Cycle program is the user-defined program at the lowest level of the time scale, as it runs one atomic cycle of network activation, calling Emergent's discrete event simulation engine. Cycle's only parameter is the network on which to operate.

This chapter described in depth the design and rationale of the simulated experiment carried out, and the structure and software implementation of the modelled network in Emergent. Based on this, the next chapter presents the results of the simulations.

RESULTS

This chapter lists and describes the numerical and ordinal output of the simulated tasks defined in Chapter 4. The results described here are interpreted and discussed in Chapter 6.

5.1 UNIT BEHAVIOUR

It is necessary to verify that the proposed model of channelopathy triggers the expected electrical behaviour of individual neurons. This section describes in details the parameters changed in the unit model described in the previous chapter, and their observed effects on the simulated electrophysiological timeline.

The dysfunctions of L-type voltage-gated calcium channel (LTCC) present in cases of autism⁸⁴ lead to an excess of calcium influx [36]. LTCC are the main actors of the chain reaction that generates action potentials. We can model the excess of calcium influx during AP generation by increasing $\bar{g}_e$ above its control value, resulting in an increased depolarisation of the membrane for the same input. On short term, the unit is more active, but on longer term, the hysteresis and accommodation mechanisms, being activated by intra-cellular calcium concentrations, are expected to modulate the activity of the unit earlier. These changes in timings of accommodation and hysteresis are simulated by decreasing respectively `acc.b_inc_dt` and `hyst.b_inc_dt`, as explained below.

Equation 4.4 page 92 shows that the basis variables b_a and b_h , which integrate recent activity, are compared to activation and deactivation thresholds to control accommodation and hysteresis. These basis variables further depend on time integration terms dt_{b_a} and dt_{b_h} to adjust the reactivity of the basis variables (Equation 4.5 page 93). The constant dt_{b_a} is differently valued when b_a is decreasing ($dt_{b_a,dec}$) than when it is increasing ($dt_{b_a,inc}$). The latter parameter, named `acc.b_inc_dt` in the Emergent programming environment and in Table 5.1, is increased to model the effect of greater calcium influx on unit accommodation in calcium channelopathy. Similarly, the constant dt_{b_h} is valued $dt_{b_h,dec}$ and $dt_{b_h,inc}$, depending on the direction of change of b_h . The parameter $dt_{b_h,inc}$ corresponds to Emergent's `hyst.b_inc_dt`, and is used to simulate the effect of greater calcium influx on hysteresis.

L-type voltage-gated channelopathy is simulated by raising $\bar{g}_e$ and decreasing $dt_{b_a,inc}$ and $dt_{b_h,inc}$.

⁸⁴ The problems of calcium homeostasis in autism, and L-type voltage-gated calcium channelopathies in particular, are discussed on page 49.

The parameters defining an Emergent model neuron that implement these changes are described in Table 5.1. This table lists the various parameters that control the unit hysteresis and accommodation in Emergent. Accommodation and hysteresis are each modelled by one ionic channel that can be seen as letting positive or (respectively) negative charges into the cell, although it is likely that the phenomena of hysteresis and accommodation in biological neurons involve more channels and other triggers. These parameters are directly mapped from Equations 4.4 (p. 92) and 4.5 (p. 93).

Table 5.1: Hysteresis and accommodation parameters in Emergent

Parameter	Description
<i>Hysteresis</i>	
gc.h	Hysteresis channel conductance
vcb.hyst	Hysteresis channel basis variable - it reflects the reactivity of hysteresis
hyst.b_inc_dt and hyst.b_dec_dt	Variables modulating the basis variable to define the speed at which the activation and deactivation threshold are reached
hyst.g_dt	Adjusts the rate of change in conductance of the hysteresis channel
hyst.a_thr	Hysteresis channel activation threshold
hyst.d_thr	Hysteresis channel deactivation threshold
g_bar.h	Maximum conductance of the hysteresis channel
<i>Accommodation</i>	
gc.a	Accommodation channel conductance
vcb.acc	Accommodation channel basis variable, adjusting the reactivity of accommodation
acc.b_inc_dt and acc.b_dec_dt	Variables modulating the basis variable to define the speed at which the activation and deactivation threshold are reached
acc.g_dt	Adjusts the rate of change in conductance of the accommodation channel
acc.a_thr	Accommodation channel activation threshold
acc.d_thr	Accommodation channel deactivation threshold
g_bar.a	Maximum conductance of the accommodation channel

Figure 5.1: Trace of the activation of a single unit (a) as usually used in modelling studies, and (b) simulating hysteresis and accommodation.

*Comparison of the
electrodynamics of a
neuron with and
without modelled
accommodation and
hysteresis*

Figure 5.1 on page 113 illustrates the behaviour of the model neuron subject to the approximations of accommodation and hysteresis expounded earlier, as compared to that of units that neither include hysteresis nor accommodation. The *net input* on Figure 5.1 traces the external stimulus sent to the unit, turned on from cycle 10 to 200; *membrane voltage* is the membrane potential, *net current* traces the transfer of electrical charges through the membrane, and *net output* shows the rate-coded cell output. Part (a) of Figure 5.1 traces the state of the simpler unit. The membrane potential rises to a plateau with the net input to the cell and comes back to the resting potential right after the input ceases. In contrast, the trace of the same membrane potential in the cell with hysteresis and accommodation (part (b) of Figure 5.1) reveals hidden dynamics, as the membrane voltage does not follow the net input. Changes in potential at points A and C correspond to the same changes in graph (a), triggered by the opening and, respectively, closing of glutamatergic excitatory ion channels. However, changes B, D and E are due to the hysteresis and accommodation mechanisms, with both phenomena stopping after the input to the cell has stopped: D marks the closing of hysteresis channels, and E the end of accommodation.

*Comparison of the
electrical activity of
a control and of a
channelopathic
neuron*

Figure 5.2 superposes the activation of the control unit (faded traces, $\bar{g}_e = 0.4$, $\text{hyst.b\_inc\_dt} = \text{acc.b\_inc\_dt} = 0.01$) to the activation of the unit implementing the proposed ion channel disturbances (dark traces, $\bar{g}_e = 0.45$, $\text{hyst.b\_inc\_dt} = \text{acc.b\_inc\_dt} = 0.03$). As in Figure 5.1, the net input sent from the external stimulus to the recorded unit is active from cycle 10 to 200. This net input, the membrane potential, the transfer of electrical charges through the membrane, and the rate-coded cell output are all traced twice, once for the control unit, and once for the channelopathic unit. I observe that the model of channelopathy triggers a higher net input, following the larger $\bar{g}_e$, and an increased output rate, expected result of the raised excitability. Hysteresis takes place much earlier, and the start of the accommodation is slightly delayed by the influence of the larger earlier activity that on the basis variable. Accommodation and hysteresis lasts longer than in the control unit. While the total depolarisation obtained after hysteresis is higher than in the control, the difference is entirely accounted for by the larger initial depolarisation, and the additional offset brought by hysteresis is the same. Similarly, the hyperpolarisation that results from accommodation is of the same amplitude as in the control case. The fact that the voltage offsets of hysteresis and accommodation do not vary with channelopathy is reasonable, because hysteresis and accommodation are the phenomenologically modelled effects of a variety of cell mechanisms that are not expected to make direct use of LTCC [1, pp. 66–67].

Figure 5.2: Trace of the activation of a single unit with accommodation and hysteresis. The traces in faded lines correspond to a unit with standard parameters, and in saturated lines, to a unit modelling a calcium channelopathy.

5.2 NETWORK LEARNING

Network learning is the process by which the long-range connection weights between units of connected maps are adapted. The network weights are learnt using a combination of self-organising Hebbian features learning and of error-driven task learning. All mathematical details of the learning process can be found in Section 4.3.5, page 103. In practice, simulating unit hysteresis and accommodation during learning lengthens the process because of the extra computations involved and slower settling of units; hence, the learning of weights was first done without hysteresis or accommodation.⁸⁵ Hysteresis and accommodation were activated after this initial full learning. This resulted

*Practicalities on
training the network*

⁸⁵ Fully learning the weights of this network without hysteresis and accommodation takes about 24 hours using 8 Intel i7 processor cores and 16GB of RAM. In that time, around 800 epochs (500 trials each) are run.

Figure 5.3: Plot of the average sum of square errors per epoch in the output layers of the network during a 700-epoch learning batch (thin line), and moving average centered on 50 epochs (thick line)

in a small increase of the error rate on the training sample, but re-learning with accommodation and hysteresis from that new starting point was comparatively much shorter than full learning with those enabled from the start.

The geometry of the spatial output map is 3×3 , with a target total rate activation of 1, and k-WTA settings encouraging a maximum activity of 1 spread over 1 to 4 units. Target spatial output is by design determined by a maximum of 4 units. Each unit is active in proportion to the spatial share of the object present in its preferred feature zone.⁸⁶ This, combined with the prior bias of localised projections, makes that the sum-of-square error (SSE) for that layer does not exceed 1.5. It can take any value up to 1.5, whereas the SSE for the semantic output map is expected to be close to 2 on erroneous trials, or 0 on successful ones. Indeed, the semantic output map contains 6 units (one per input symbol), and the k-WTA inhibition is set for that map so as $k = 1$.

The semantic output map's binary error signal is much less informative than the spatial output map's continuously-valued error signal, and this results in a much slower learning of weights in the dorsal stream than in the ventral stream. The spatial weights are usually

*Sum-of-square error
calculations for each
output map*

⁸⁶ By this I refer to the centre of "third-degree" receptive field, following the shortest path from input through feed-forward network connectivity.

learnt within 100 epochs, and the semantic maps's weights take 4 to 6 times more epochs to be learnt.

Figure 5.3 plots the epoch-average sum squared error⁸⁷ between the expected and actual output vectors during the minus phases of one full learning cycle. The error in the spatial stream output is minimised much faster than the semantic error.

5.3 NETWORK BEHAVIOUR

5.3.1 Statistical analysis

Notation

In this section and in subsequent references, the following symbols are used (with additional subscripts):

$\bar{m}$ refers to the *sample arithmetic mean of a dataset*,

s designates the *sample standard deviation of a dataset*, defined using all available exemplars as the square root of the expected deviation from the *sample mean*,

μ denotes the *expected value of a theoretical random variable*,

σ denotes the *standard deviation of a theoretical random variable*, defined as the integral of its probability density function multiplied by its value,

$\hat{\mu}$ denotes the *location parameter of the distribution underlying a theoretical random variable*, when distinct from μ , and

$\hat{\sigma}$ denotes the *scale parameter of the distribution underlying a theoretical random variable*, when distinct from σ .

As explained in Section 4.2 the network's response is measured in three task conditions:

NEUTRAL condition measuring the reaction time when the stimulus appears immediately after 50 time units without any stimulus.

GAP condition measuring reaction time when an interval of 50 time units separates the disappearance of the central stimulus and the appearance of the peripheral one.

OVERLAP condition measuring reaction time when the first stimulus remains visible during 50 time units after the second stimulus is displayed.

*Summary of the
three task conditions*

⁸⁷ The epoch-average sum squared error is calculated as the mean of the sum-of-square differences between the vector of the activity rates of all output units and the expected output of the network.

In these three conditions, maximal excitability, accommodation and hysteresis parameters are varied to simulate a channelopathy.⁸⁸ As a reminder, the values of the parameters that vary between the two networks are as follows:⁸⁹

	$\bar{g}_e$	$dt_{b_h,inc}$	$dt_{b_a,inc}$
control	0.4	0.01	0.01
channelopathic	0.45	0.03	0.03

Simulation setup

Each of the three task conditions of the experiment is repeated between 3900 and 4300 times per each of the two sets of parameter values. One parameter set corresponds to one group of subjects in the experimental setup. Invalid trials are defined as those with no response after 300 time units, or where the spatial or semantic output mismatch the second stimulus. Such invalid trials are removed from all datasets.

Table 5.2: Descriptive statistical summary of the dataset, in full and clustered by network, task, and combinations.

	<i>Dataset</i>	<i>Minimum</i>	<i>First quartile</i>	<i>Median</i>	<i>Mean</i>	<i>Standard deviation</i>	<i>Third quartile</i>	<i>Maximum</i>
network	Full	15	36	54	56.4	22.97	68	290
	Control	15	35	57	58.3	23.95	73	223
	Channelopathic	25	38	53	54.5	21.72	65	290
	Neutral	42	52	55	55.1	4.374	58	74
	Gap	15	31	33	35.3	14.78	37	290
	Overlap	38	69	76	80.1	18.86	87	240
network and task	Control × Neutral	44	55	57	57.5	3.947	60	74
	Control × Gap	15	31	33	33.5	4.279	35	87
	Control × Overlap	42	75	83	86.7	17.9	94	223

continued on next page

⁸⁸ See page 114 for more explanations.

⁸⁹ For further discussion of the effects of L-type voltage-gated calcium channel (LTCC) dysfunctions on neuronal excitability, please refer to Chapter 3. The increase in maximal excitatory conductance and in the hysteresis and accommodation time integration constants corresponds to these expected changes in calcium dynamics.

continued from previous page

network and task	Dataset	Minimum	First quartile	Median	Mean	Standard deviation	Third quartile	Maximum
	Channelo. × Neutral	42	50	52	52.4	3.112	54	66
	Channelo. × Gap	25	31	34	37.2	20.64	38	290
	Channelo. × Overlap	38	65	70	73.7	17.55	78	240

Table 5.2 gives a statistical overview of the resulting datasets. It details the total and interquartile ranges, median, arithmetic mean and standard deviation:

- overall – row named **Full**. This refers to statistics computed from all available data, aggregating both types of network, control and channelopathic, and all three task conditions, Neutral, Gap and Overlap.
- for each network regardless of the task condition – **Control** and **Channelopathic** rows. These statistics aggregate all types of experiments, only differentiating both types of network.
- for each task condition, regardless of the networks type – **Neutral**, **Gap** and **Overlap** rows. This aggregates both types of networks, but gives the descriptive timing statistics for each task condition.
- for each network in each task condition – rows **Control × Neutral**, **Control × Gap**... until **Channelo. × Overlap**). These six rows of descriptive statistics are the most informative; they correspond to each network, the control network and the channelopathic network, in each task condition (Neutral, Gap and Overlap).

Table 5.2 reveals drastically different responses of the network in presence of channelopathy compared to the control network. This is hinted by the 10% difference in mean response time in the “network” rows. Furthermore, the “network and task” rows show that this difference in response time between the channelopathy-simulating and the control networks is dependent on the task. Indeed, on one hand, in the Neutral and Overlap conditions, the mean response time of the control network is higher than that of the channelopathic network. On the other hand, the effect is reversed in the Gap condition, where the control network’s response is faster than the channelopathic network’s response.

Additionally, the standard deviation of the response is a little higher in the control network for the Neutral and Overlap conditions,

*Descriptive
statistical tables and
figures*

but much lower in the Gap condition. However, this is seemingly contradictory with the observation that the difference in inter-quartile ranges between the Gap condition in the channelopathic network and the same condition in the control network does not show the same drastic difference. This, and the fact that the full response range to the Gap condition in the channelopathic network is much higher than the response range to the Gap condition in the control network, hint that there may be many more high-response time outliers in the response to the Gap condition in the channelopathic network.

In complement to Table 5.2, the three histograms that correspond to the network response time for the control parameter set in the three task conditions are presented in Figure 5.4. The histograms of the response times of the control network in repeated overlap, neutral, and gap conditions are plotted in that order on a common abscissa, with invalid trials removed. The tail of the overlap condition histogram is cut off for readability, as it extends to time units 223. The x-axis range and bin widths are also the same as on Figure 5.5 to ease comparison.

The three stacked histogram in Figure 5.5 correspond to the channelopathic parameter set. Like in Figure 5.4, the histograms only show the dataset in the [24, 124] range, without the long tails, for better readability.

H_0 : all groups
responses follow the
same distribution.

Between-factor =
network type

Within-factor = task
condition

Group = task $\times$
network

Let “The control and channelopathic networks timings datasets in each of the three task conditions are samples of the same population.” the null hypothesis (H_0). I use the usual vocabulary of statistical analysis to refer to the simulated experimental design. The network parameter set is the *between-group* (or *between-factor*) part of the design, with two levels: Control and Channelopathic. The Control group is the name given to the reference network, and the Channelopathy group is the name of the network using units that model channelopathy. The task condition is the *within-factor* (or *within-group*), with three levels: Overlap, Neutral and Gap, as described on page 117. I refer to the intersection of a between-factor and within-factor as *group*, and denote the intersection using the following relational symbol: $\times$.

To confidently determine whether the distributions underlying the reaction time of groups differ between one another, I first attempt a two-way fixed-effects ANOVA. The dataset appears heteroscedastic across groups, as Bartlett’s test of the null hypothesis that the variances in each of the groups are the same rejects the hypothesis at any conceivable level of confidence ($\chi^2 = 23911$, 5 degrees of freedom, $p < 2.2E^{-16}$). However, the experiment design is balanced (sample sizes are similar), a case in which ANOVA’s F-test is robust against inequality of variances.

Table 5.3 summarises the results of the two-way analysis of variance on the attention shift network response times. Network type (Control, Channelopathic) and task condition (Overlap, Neutral, Gap) are the factors, and response time during the attention shift is the depend-

Figure 5.4: Histograms of the response times of the control network in repeated overlap, neutral, and gap conditions

Figure 5.5: Histograms of the response times of the channelopathic network in repeated overlap, neutral, and gap conditions

ent variable. The resulting low p-values (column Pr(>F) in Table 5.3), statistically significant below the $1/1000^{\text{th}}$ level, are strong evidence against H_0 .

A priori, the ANOVA rejects H_0 .

Table 5.3: Summary of the results of two-way ANOVA on the attention shift network response times.

Summary of hypothesis testing

	Df	Sum sq	Mean sq	F value	Pr(>F)
Task	2	8121401	4060700	22764	$< E^{-16*}$
Network	1	130039	130039	729	$< E^{-16*}$
Task×Net	2	280103	140052	785	$< E^{-16*}$
Residuals	24437	4359179	178	* : 0.001 significance	

Table of means and effects

	Task condition			Network parameters	
	Overlap	Neutral	Gap	Channelop.	Control
Mean	80.1	55.07	35.32	54.09	58.71
Effect	23.66	−1.37	−21.12	−2.35	2.26
Count	7877	8284	8282	11988	12455

Table of means and effects (continued)

Task condition × Network parameters					
Overlap × Channelop.	Overlap × Control	Neutral × Channelop.	Neutral × Control	Gap × Channelop.	Gap × Control
73.74	86.65	52.45	57.51	37.25	33.52
−4.09	4.21	−0.23	0.21	4.32	−4.03
3996	3881	3996	4288	3996	4286

Grand mean: 56.44   Total count: 24443

To validate these results, I test the normality of the residuals dataset (assumed by ANOVA) by plotting its quantiles against the quantiles of the maximum-likelihood fitted Gaussian. This reveals that the normality assumption is not held by the ANOVA residuals. The deviation in the heavy tail of the data is a large deviation from normality, which seriously undermines the value of the statistics computed by the test, and further prevents the derivation of any useful linear model of effects.

However, residuals are not normally distributed.

To address this issue, it is possible to apply a normalising transformation to the whole dataset, with the expectation that the residuals will thence be normalised and the heteroscedasticity, reduced. The traditional non-parametric dataset transformations (e.g. logar-

Box-Cox
transformation

ithm, or square root) are advantageously replaced by a more general optimised power transform of the Box-Cox family.⁹⁰

I optimise a Box-Cox function to normalise the data, leading to $\lambda = -0.5703$, and apply ANOVA on the transformed data. The results are shown in Table 5.4.

Table 5.4 is a summary of two-way analysis of variance on the attention shift network response times after applying a Box-Cox normalisation with $\lambda = -0.5703$. Network type (Control, Channelopathic) and task condition (Overlap, Neutral, Gap) are the factors, and response time during the attention shift is the dependent variable.

Table 5.4: Summary of the results of two-way ANOVA on the Box-Cox transformed response times of the attention shift network.

<i>Summary of hypothesis testing</i>					
	Df	Sum sq	Mean sq	F value	Pr(>F)
Task	2	31.93864	15.969320	73946.28	$< E^{-15*}$
Network	1	0.14355	0.143548	664.7016	$< E^{-15*}$
Task×Net	2	0.51095	0.255475	1182.9823	$< E^{-15*}$
Residuals	24437	5.27738	0.000216	* : 0.001 significance	

The residuals on the Box-Cox transformed dataset are still not normally distributed, although the departure is lesser. ANOVA is often used in the literature to analyse effect size, in spite of unequal sample sizes, non-normality or heteroscedasticity, because ignoring the assumption of this test under the pretense of its “robustness” is common practice. I believe that such analysis would not be very valuable in the face of the violation of the ANOVA assumption of normality. However, in consideration of the combination of low p-values and the differences in shapes and locations in the histograms (see Figures 5.4 and 5.5), the hypothesis H_0 that the control and channelopathic networks timings datasets in each of the three task conditions are samples of the same population can be confidently rejected.

H_0 rejected

Parametric fitting

To further characterise these results, I try fitting normal, log-normal, gamma and Weibull distributions to the data clustered by task condition $\times$ parameter set using maximum likelihood estima-

⁹⁰ The Maximum Likelihood optimisation of the power of the Box-Cox function

$$y_i^{(\lambda)} = \begin{cases} \frac{y_i^\lambda - 1}{\lambda}, & \text{if } \lambda \neq 0, \\ \log(y_i), & \text{if } \lambda = 0. \end{cases}$$

yields an exponent λ , such that the transform of samples y_i to $y_i^{(\lambda)}$ leads to a dataset that is optimally close to normality.

tion (MLE) and report the results in Table 5.5. Among these usual parametric distributions, the log-normal distribution provides the best fit in all these cases.⁹¹ This table gives a better indication of the spread of the output timings, heaviness of the right tails, and gives a clearer picture of the dispersion of the datasets.

Table 5.5 is organised by network type column-wise, and by condition row-wise. It contains several MLE-fitted log-normal distributions and parametric moments of reaction timing data: for each task condition each of both the channelopathic and control networks parameter sets. The symbols $\hat{\mu}_{a,b}$ and $\hat{\sigma}_{a,b}$ denote the estimated location and scale parameters, *id est* the mean and standard deviation of the logarithm of the timing values. Symbols $\mu_{a,b}$ and $\sigma_{a,b}$ refer to the arithmetic mean and standard deviation of the estimated log-normal distribution for parameter set a and condition b. Fitted trimmed histograms and quantile-quantile plots are shown in the bottom of each cell to illustrate the goodness of fit. The range of the dataset and the Gini coefficient are also reported to quantify statistical dispersion.

Description of Table
5.5

Table 5.5: MLE-fitted log-normal distributions and parametric moments of reaction timing data

Control parameter set	Channelopathic parameter set
<i>Neutral condition</i>	
$\hat{\mu}_{ct,N} \approx 4.04959$ $\hat{\sigma}_{ct,N} \approx 0.06874$ ± 0.0010497 ± 0.0007415 $\mu_{ct,N} \approx 57.51$ $\sigma_{ct,N} \approx 3.958$	$\hat{\mu}_{ch,N} \approx 3.9580$ $\hat{\sigma}_{ch,N} \approx 0.0594$ ± 0.0009408 ± 0.0006644 $\mu_{ch,N} \approx 52.45$ $\sigma_{ch,N} \approx 3.122$
Range: [44, 74] Gini: 0.03839	Range: [42, 66] Gini: 0.03309

continued on next page

⁹¹ Person's Chi-squared is used to assess the goodness of fit and determine that the fit of other classical positively skewed continuous distributions on $\mathbb{R}^+$ is much worse.

continued from previous page

The arithmetic mean μ and standard deviation σ of the estimated log-normal distributions are obtained from the estimated location and scale parameters $\hat{\mu}$ and $\hat{\sigma}$ by Equations 5.1 and 5.2:

$$\mu = e^{\hat{\mu} + \frac{1}{2}\hat{\sigma}^2} \quad (5.1)$$

and

$$\sigma = \mu \sqrt{e^{\hat{\sigma}^2} - 1} \quad (5.2)$$

Fitting the output data with a log-normal distribution is justified by the following:

1. Log-normal is a likely distribution for the corresponding clinical experiment described in subsection 6.6.1. Indeed, across scientific disciplines, left-bound experimental data is often found to be log-normally distributed [162], and in particular in reaction times studies, along with the gamma and Wald distributions [163]. However, these two alternatives provide a worse fit to the simulated data. By estimating the parameters for a log-normal fit, I estimate the parameters for the expected distribution real-world data.
2. Conceptually, the log-normal distribution is the result of the multiplicative combination of many independent variables. This is a simple and fundamental concept, pendant to the additive nature of the normal distribution, which may explain why both are ubiquitous across sciences. In the case of EVAC, the multiplicative dynamics are present in the neuronal interactions. The assumption of independence does not hold, as these individual neuronal components are connected. However, as the network scales up towards the size of biological neural networks, the interdependence of the response of any two units becomes more indirect, resulting in an approximately similar situation overall as if the statistical distribution of the response of the network followed that of a set of independent random variables.

Log-normal is a plausible distribution of clinical timing response, and a model for multiplicative independent normal processes.

Furthermore, the output of the discrete-time simulation is distributed on a finite integer range, as the atomic time unit is one simulation cycle; over enough simulation runs, this results in a majority of ties, and in a discretely sampled probability distribution. It might be argued that this discrete data would be more easily fitted with a discrete distribution. However, it has the disadvantage of ignoring the expected clinical data, which is to be modelled with continuous functions. Hence, modelling the simulation output with continuous distributions is sensible in view of comparison with the output of related experiments in experimental psychology.

The theoretical probability density functions superposed to the density histograms in the inline figures of Table 5.5 seem to show

Log-normal fits the Neutral task, but not the Gap and Overlap tasks responses due to their heavy right-tails.

a suboptimal fit, in particular for the channelopathic parameter set. This is because their long tails are not shown in that histogram. The quantile-quantile plots at the bottom of the data cells reveal how the heavy tails of the simulated data do not match their MLE-fitted log-normal hypotheses.

For both network parameter sets, the log-normal distribution provides a good fit to the Neutral condition, but much less to the Overlap and Gap conditions. The heavy tails of the corresponding datasets are culprits, as shown on the quantile-quantile plots of Table 5.5. It is notable that the Neutral condition does not exhibit this long tail on the right, and in both networks, the range of the data for that condition remains the lowest of all conditions. The ranges are [44, 74] and [42, 66] for the control and channelopathic networks respectively. The Gap condition is not as symmetrical between the control and the channelopathic networks: the former spreads on [15, 87], while the latter varies from 25 to 290.

In both networks, Overlap has the longest mean reaction time and Gap the shortest.

The ordering of task conditions by performance (mean reaction time) does not vary between parameter sets. The Overlap condition is the slowest in both cases ($m_{ct,O} \approx 86.65$ and $m_{ch,O} \approx 73.74$ in the data, $\mu_{ct,O} \approx 86.58$ and $\mu_{ch,O} \approx 73.8$ with the log-normal model), followed by the Neutral condition ($m_{ct,N} \approx 57.51$ and $m_{ch,N} \approx 52.45$ in the data, or $\mu_{ct,N} \approx 57.51$ and $\mu_{ch,N} \approx 52.69$), and finally the Gap condition is the fastest ($m_{ct,G} \approx 33.52$ and $m_{ch,G} \approx 37.25$ in the data, $\mu_{ct,G} \approx 33.5$ and $\mu_{ch,G} \approx 36.79$ in the model).

Channelopathy helps attention shift in Overlap and Neutral, and hinders it in Gap.

In the Overlap and Neutral task conditions, channelopathy facilitates attention shift, as it takes a longer mean time to shift attention for the control network than for the network modelling channelopathy (Overlap: $m_{ct,O} \approx 86.65$ is longer than $m_{ch,O} \approx 73.74$ – Neutral: $m_{ct,N} \approx 57.51$ is more than $m_{ch,N} \approx 52.45$), while in the Gap task condition, channelopathy hinders attention shift ($m_{ct,G} \approx 33.52$ is shorter than $m_{ch,G} \approx 37.25$).

Standard deviation in the control network is highest for the Overlap task.

Sample standard deviation of the timing response in the control network is clearly lowest in the Neutral task, slightly higher in the Gap condition, and much higher for the Overlap: $s_{ct,N}(3.947) < s_{ct,G}(4.279) < s_{ct,O}(17.9)$. The Overlap condition has a higher standard deviation than the other, less demanding disengagement conditions both in the fitted distributions and in the data. However, the Neutral task of the control network has less output variability than the Gap task, which contrasts with the log-normal model, where the standard deviation of the modelled distribution of the timing response increases as follows: $\sigma_{ct,G}(3.925) < \sigma_{ct,N}(3.958) < \sigma_{ct,O}(16.45)$. Albeit the modelled standard deviation of the time response in the Neutral task condition for the control network ($\sigma_{ct,N} \approx 3.958$) is quite close to that in the Gap condition ($\sigma_{ct,G} \approx 3.925$), this ordinal failure of the log-normal model reveals that its fit to the gap condition is too poor to be useful: the control data's sample standard deviation for

the gap condition is $s_{ct,G} \approx 4.279$, much higher than the estimated distribution's standard deviation $\sigma_{ct,G} \approx 3.925$.

The variance of the response delay to each task in channelopathic network is ordered differently from that of the control network. The Neutral condition also has the lowest variance, but the Overlap condition now takes the second place, and the Gap condition has the greatest standard deviation: $s_{ch,N}(3.112) < s_{ch,O}(17.55) < s_{ch,G}(20.64)$. The log-normal models present a similar issue as with the control parameter set. They do not preserve the ordering of the datasets by variance: $\sigma_{ch,N}(3.122) < \sigma_{ch,G}(8.729) < \sigma_{ch,O}(13.25)$. Like in the control network, the largest error is on the Gap condition, where $\sigma_{ch,G} \approx 8.729$ fails to model $s_{ch,G} \approx 20.64$.

The channelopathic network shows slightly less variance overall than the control network ($s_{ct} \approx 23.95$, $s_{ch} \approx 21.72$). The Neutral task follows that pattern, as channelopathy leads to slightly less variance than the control network ($s_{ch,N}(3.112) < s_{ct,N}(3.947)$). This last result is correctly modelled by the log-normal distribution ($\sigma_{ch,N}(3.122) < \sigma_{ct,N}(3.958)$), as expected thanks to its better fit to the Neutral task timings. In the Overlap task, the control group timings are also a little more volatile, with both variance values remaining of the same order: $s_{ch,O}(17.55) < s_{ct,O}(17.9)$. Finally, contrary to the Neutral and Overlap tasks, the standard deviations of the Gap task are very different between network parameter sets. Moreover, the Gap condition is the only one where the relation is inverted: $s_{ct,G}(4.279) < s_{ch,G}(20.64)$.

The ranges and dispersions of the groups differ. The range and Gini coefficient are displayed at the bottom of each cell in Table 5.5, together with quantile-quantile plot to fitted log-normal to visualise the tails. The neutral condition shows the lowest dispersion with both networks: the control $\times$ Neutral and channelopathy $\times$ Neutral groups have the lowest Gini coefficients among all groups. The output of these simulations are not as heavy right-tailed as others and their ranges are also the smallest. Without consideration of network type, the most dispersion between conditions is reached with the Overlap condition. However, contrary to the Neutral task, there is a large variation of range and/or dispersion between networks within both the Gap and and the Overlap conditions, hinting at a deep effect of the modelled channelopathy on attention shift. The effect is not the same between the two conditions:

- while channelopathy in the Overlap condition does result in an increase in response range span, the dispersion only changes by 9.8% ($Gini_{ct,O} \approx 0.1056$, $Gini_{ch,O} \approx 0.0952$);
- in contrast, in the Overlap task condition, both the response span and dispersion increase much with channelopathy ($Gini_{ct,G} \approx 0.06436$ and $Gini_{ch,G} \approx 0.0952$ correspond to a 99.2% Gini increase, and the range span of 72 time units of the con-

Standard deviation in the control network is highest for the Gap task.

The Gap task has a much higher variance with the channelopathic network than with the control network.

Timing output is more dispersed (tails, ranges, Gini) in attention switch tasks.

trol network becomes 265 time units wide in the channelopathy network).

5.3.2 Network performance

After presentation of the target input, for which the reaction time of the network is measured, the semantic layer is expected to output the class of the object presented within 300 time units. A trial is considered valid as soon as the corresponding identity unit is activated by more than 60% of its maximum output rate.⁹² When this expectation is not met, the trial is considered erroneous, and excluded from the response timing statistics shown in Section 5.3.1.

Measures of network performance: r_ϵ for classification error rate and r_ζ for activation failure rate

To complement the timing statistics on successful trials, the following section presents the performance of the trained network, calculated for each group (task $\times$ network) of N trials as:

CLASSIFICATION ERROR rate r_ϵ – if N_ϵ is the number of trials where the wrong semantic output unit is activated by more than 0.6 in less than 301 time units, then $r_\epsilon = N_\epsilon/N$.

ACTIVATION FAILURE rate r_ζ – if N_ζ is the number of trials where no semantic output unit gets activated by more than 0.6 in less than 301 time units, then $r_\zeta = N_\zeta/N$.

The results are presented in Table 5.6, clustered by task condition and network.

Table 5.6: Classification error rates and activation failure rates by Network type (control, channelopathic) and task condition (Overlap, Neutral, Gap)

Task condition	Control		Channelopathy	
	Classification error rate	Activation failure rate	Classification error rate	Activation failure rate
All tasks	0	0.03179	0.001	0
Neutral	0	0	0.001	0
Gap	0	0.0004664	0.001	0
Overlap	0	0.09492	0.001	0

⁹² The 60% threshold is arbitrary, but follows the convention of O'Reilly & Munakata [1]. Moreover, in the model of emergent visual attention under channelopathy (EVAC), the spurious activation of two or more semantic output units by more than 60% each is made impossible by the use of a strict 1-WTA inhibition rule over the semantic output layer.

The control network in Neutral condition outputs the correct class of the target unit within 300 time units in 100% of cases. This is an expected result, as the control $\times$ Neutral group reflects both the training parameters and training inputs of the network: the training parameters are the same as the control parameters, except of course for the synaptic plasticity, with no influence on the independent dynamics of individual units. Also, like the target inputs of the Neutral task inputs, the training inputs are not preceded by a distractor.

This is unlike all other task $\times$ network groups, where either or both of the network type and task conditions differ significantly from the training parameters and trial-level input sequences.

In both networks, the classification error rate remains consistently equal across all task conditions. For the control networks, that is $r_{\zeta,ct} = 0$: spurious activation of an incorrect unit appears sufficiently unlikely not to be reflected in the datasets. The channelopathic network classification error rate is $r_{\zeta,ch} = 0.1\%$ in all tasks. This corresponds to exactly $N_{\zeta,ch} = 4$ incorrectly classified trials for each task group. This is a coincidence, and the 4 inputs that lead to the erroneous trials are not the same across tasks: the input table is a uniform random selection of symbol and retinal location that is different for each group (task condition $\times$ network), and the location, rotation and scaling parameters are continuously valued, making collisions very unlikely. In any case, individual inspection of the inputs that triggered the classification errors shows that they all differ in location and identity.

Contrary to the classification error rate, the activation failure rate is null in all channelopathy output datasets, and positive in control datasets. In the latter, the Gap task has a very low activation failure rate of $r_{\zeta,ct,G} \approx 0.047\%$, corresponding to $N_{\zeta,ct,G} = 2$ errors over $N_{ct,G} = 4288$ trials. This result contrasts with the high activation failure rate of the control network in the Overlap task condition: $r_{\zeta,ct,O} \approx 9.5\%$. In that case, most failures to activate the output unit above 0.6 come from a remarkable phenomenon: although the network state changes with the appearance of the target and further with the disappearance of the distractor, it reaches an equilibrium state in which enough activity correlated with the distractor is carried over after its disappearance to lead to a hybrid state where this spurious activity remains and prevents the complete transition to the target output. Typically, in that case, the semantic output units that correspond to the distractor and to the target both exhibit a low level of activity. Although I have no evidence that this network state is in any way similar to transitional stated during successful attention shift in the Overlap condition, I deem it plausible, and call this type of phenomenon an *incomplete attention shift*. Formally defining and identifying it is not easy, because formalising incomplete attention shifts requires understanding the sparse distributed representations learnt by hidden layers in order

The control $\times$ Neutral group reflects the training conditions, and expectedly performs flawlessly.

The channelopathic network has a 0.1% classification error rate that appears to be independent of the task at hand.

The activation failures of the control network in attention shift tasks (particularly in Overlap) may be due to incomplete attention shifts.

to map the transition processes and be able to determine if a given snapshot of activity is part of a transition between two given inputs.

The systematic activation of the channelopathic network in all tasks ($r_{\zeta,ct} = 0$) indicates that such hypothesised “activation ties” are very unlikely to happen in this model in presence of channelopathy.

Due to recurrent dynamics, network activity slowly fades during the inter-stimulus pause of 50 cycles of the Gap task. The state of the network when the target appears is thus quite different from the Neutral case, which is only preceded by 50 cycles without input. Thus, a plausible explanation of the small activation failure rate of the control network in the Gap task is that it can also be prone to incomplete attention shifts.

This chapter expounded the quantitative results of the simulation of the EVAC model. It revealed significant differences in the timing of attention shift between the simulated task conditions, as well as between the network with control parameters and network with channelopathic parameters. Error rates in each settings and sporadic insights into the dynamics of the network were also presented. Chapter 6 recapitulates the study and discusses those results in light of the research questions and possible future work.

DISCUSSION

6.1 RECAPITULATION OF THE STUDY

Having reviewed past and current studies on the psychology and neurology of autism, I determined that simulating the hypothetical effect on visual attention of channelopathies possibly linked to autism would be novel and significant.

Hence, I reviewed the most prominent of relevant studies on modelling of neurological mechanisms, and in particular on the modelling of autism and other pervasive developmental disorder (PDD), taking note of the common use of SOM and abstract point neurons. As such, current works on autism in computational neuroscience are modelling abstract issues related to the psychological descriptions of autism. The introduction of more biological realism in such models could make them more useful to neurologists. Reflex visual attention is seen as a mechanism that could benefit from such modelling of intermediate complexity. Seeking to strike a compromise between computational expense and biological realism, I reviewed available software, keeping in mind that my imperfect understanding of the detailed functioning of neurons prevents me from using some of the more complete packages. At the same time, considering that the human visual pathways are among the most studied neural systems and have a number of computational models describing them, I built a model of early dorsal and ventral visual pathways and their interactions for low-level attention, and implemented it in Emergent, along with learning and testing programs. The testing programs correspond to three tasks: two reflex attention shift tasks, and one simple attentional task that does not involve such attention shift. After implementing a phenomenological model of L-type voltage-gated calcium channelopathy, I verified that the cell model had the expected electrophysiological behaviour. I repeatedly ran the trained network on those three tasks and using either unit parameters simulating channelopathy or control unit parameters, collected a total of about 25000 measures of response time and performance of the network. Finally, I used simple statistical tools to ensure that those response times differ significantly and to describe their empirical distributions. I quantified the classification error rate and activation failure rates of the networks for each group and qualified some note-worthy dynamics during activation failure in the overlap task of the network modelling channelopathy.

*Autism in 3.2 p. 44
and relation to
channelopathies in
3.3.1 p. 48*

*Literature on
modelling in 3.3.4
p. 65*

*Visual attention in
3.3.3 p. 55*

*Software review in
Appendix B p. 201*

*Human visual
pathway in 3.3.2
p. 50*

*Architecture of
EVAC: 4.3 p. 86*

*Computational
experiment in 4.2
p. 81*

*Channelopathic
neuron model
behaviour in 5.1
p. 111*

*Timing and
performance
statistics in 5.3
p. 117*

6.2 THEORIES EMBODIED IN THE MODEL

Main theory

The model of emergent visual attention under channelopathy (EVAC) presented in this part of the thesis primarily embodies a preliminary theory of the emergent functional influence of disturbances in the dynamics of cortical networks. Those disturbances are due to defects in the mechanisms of L-type voltage-gated calcium channel (LTCC), called calcium channelopathies. The effects of these channelopathies are hypothesised to be mediated by the mechanisms of excitability and accommodation of individual neurons. Such influence would possibly be behaviourally manifest in cases of autism through – for instance – atypical attentional response. Hence, EVAC can be seen as an illustration of plausible effects of this more general mechanism in the early visual streams.

The model assumes a number of theories related to human and primate visual processing, and to more general principles of cortical organisation. Such assumptions are necessary to concretise model implementation details. Additionally, simplifying assumptions are made to focus on mechanisms essential to the principles of the primary theory being modelled. These assumptions are summarised below:

Model assumptions

- The two-stream hypothesis of visual processing is explicitly implemented in the form of divergent connectivity from the output of units in the secondary visual cortical layer. This is schematised by Figure 4.4 page 96: connectivity forms a single chain from retinal input up to the layer modelling V₂. From V₂, separate dorsal and ventral streams lead to their respective output maps. The hierarchical organisation and duality of the streams are considered essential principles of primate vision, and as such are explicitly implemented in EVAC.⁹³ I present the dual-stream hypothesis in Section 3.3.2, and the implemented structure and connectivity are detailed in 4.3.4.
- The model of emergent visual attention under channelopathy (EVAC) implements systematic recurrent connectivity, intra-layer inhibition, and abstracts the physically 6-layered natural model of cortical maps [158] into an abstract model containing one layer of point neurons. The principles behind the simulation of neural maps using a k-WTA inhibited single layer of Hodgkin-Huxley neuron models are introduced in [1].
- It is also assumed that neuronal accommodation and hysteresis have a role in the dynamics of cortical networks in preventing the short-term emergence of simple attractors. As a corollary,

⁹³ Some secondary connections of the visual system, such as connections restricted to peripheral field representations in [40, p. 158], are not present, in view of producing a minimal model.

they are hypothesised as key elements in short-term bottom-up sensory-driven mechanisms of visual attention (*emergent attention*).

- Finally, EVAC ignores any possible external top-down or lateral influence of other neural structures (e. g. frontal, subcortical) on early visual streams. It is necessary, because such top-down connections are not as well mapped and understood as bottom-up visual streams. It is also justified by the fact that I am modelling a short-term mechanism akin to a “cortical reflex”: within the duration of the modelled phenomenon, the signal propagation is hypothesised to be limited to V_1 , V_2 and the early ventral and dorsal streams. Furthermore, in the proposed implementation for human subjects of tasks similar to those modelled in the attention network (see Section 6.6), care is taken to limit both amygdalic and prefrontal influences, by selecting stimuli without facial features, and ensuring that the participant remain in a relaxed, idle state of mind throughout the study.

In view of those assumptions, the model simulation results permit a number of qualitative and ordinal predictions, expounded in the next section.

6.3 RESULTS INTERPRETATION AND MODEL PREDICTIONS

The EVAC model predicts the existence of a difference in the mean reaction time between gap and overlap conditions, such as the overlap condition is in average longer than the gap condition reaction time; this is a well-know occurrence of experimental psychology called the *gap effect*. It has been observed in close real-world equivalents to the simulated Gap and Overlap tasks conditions. The study by Saslow [164] is an often-cited reference. They found that

a minimum approximate latency asymptote of about 150 msec was reached in the vicinity of a gap of 200 msec; a maximum approximate latency asymptote of about 250 msec was reached in the vicinity of an overlap of 100msec. The effects were quite large; the asymptotes were about 50 msec or more different from the approximately 200-msec latencies at synchrony. [164, pp. 1027–1028]

The EVAC model conforms basic expectations about the gap effect in psychology (like in [164]).

This result concords with the prediction of the EVAC model that the mean timing of the control network in the Gap task condition is significantly lower than in the Overlap condition. The concordance of the model with those experimental data about human gap effect is a comforting verification of its soundness.

EVAC makes several predictions about possible changes in the dynamics of attention shift in cases of disturbed accommodation and

hysteresis. In hypothetical example, this may happen when voltage-gated calcium channels function atypically; see Section 3.3.1.

The Neutral task is used as a reference point relative to the two other tasks and models attention capture from a non-attentional state, while the Gap and Overlap tasks model reflex visual attention shift from recent and, respectively, concurrent current attentional engagement. The results expounded in Section 5.3 are thus directly interpretable as predictions on the rates of classification error and of activation failure, and on the distribution of the reaction times in similar experiments involving biological neural networks. This includes experiments in psychology, if the measures performed can be commensurably related to the reaction of early visual streams modelled, and if the model of emergent visual attention under channelopathy (EVAC) is sufficiently accurate despite its assumptions (summarised in Section 6.2).

The most prominent prediction is that *current attentional engagement hinders attention shift* in both channelopathic and control subjects, as measured by the difference between the mean Overlap task timing and that of the Neutral task in both networks. The delaying effect of current attentional engagement measured in terms of ($\mu_O : \mu_N$) ratio – so, relative to the Neutral task – is larger in the control data (1.507) than in the channelopathic data (1.407). A relative measurement as ratio to the Neutral task is expected to be more robust to model simplifications than an attempt to get absolute timing predictions for direct comparison with clinical timing data. Indeed, the simulated experiment here is simplified in comparison to the reaction time study it intends to mimic. In particular, the reaction time in the real study would likely be measured using an indirect mean such as eye saccades, so the results would encompass the timing of motor activation as well. Even assuming that sufficient direct information about the attentional timeline of the neural maps involved is available, correspondence between the the absolute timings of the model response and of the human response seems unlikely, because the model of emergent visual attention under channelopathy (EVAC) is not sufficiently faithful to the biological details.

In their clinical study of the reflex attention in autism, van der Geest et al. [39] concluded that autism is correlated with a reduced gap effect. The prediction, drawn from the simulation of EVAC, that the relative negative effect of current attentional engagement on attention shift is less strong in channelopathy is thus concordant with the conclusions drawn in [39]. The concordance of EVAC with this experiment is conclusive evidence in favour of the main theory of the model, under the premise that channelopathy is involved in the cases of the cited study.

The second prediction of the model is that *recent attentional engagement facilitates attention shift*. More precisely, this second prediction

Current attentional engagement hinders attention shift. The relative effect is less strong in channelopathy.

Model simplifications prevent accurate enough behavioural predictions in absolute terms.

The model is compatible with results in [39]

means that attention⁹⁴ to a first stimulus followed by its removal for a short period, followed by the appearance of a distinct input stimulus, decreases the time required by the network to activate its representation of the second input stimulus. The ratios of the mean Gap timings to the Neutral timings are 0.5828 with the control network, and 0.7102 for the channelopathic network. In terms of the experimental psychology counterpart of the simulated experiment, this could mean that attentional capture from a completely blank field of view would take more time than attention shift in the Gap condition.

The third prediction of EVAC concerns the *spread of the timing responses* in each task. From the sample standard deviation of the groups in Table 5.2, I expect variance to be strongly increased by channelopathy in the Gap attention shift task ($\frac{s_{ch,G}}{s_{ct,G}} = 4.823$), and slightly decreased in the Neutral and Overlap tasks ($\frac{s_{ch,N}}{s_{ct,N}} = 0.7885$ and $\frac{s_{ch,O}}{s_{ct,O}} = 0.9799$). These predictions over dispersion are quite fragile, because the sources of noise included in EVAC are insufficient to accurately represent the plethora of sources of that variability are intrinsic to the biological system modelled.

The next group of predictions of EVAC relates to the *activation and success rates*. While those can be legitimately used to make predictions about the same rates in the biological equivalents of the neural systems modelled, it may prove harder to link the measured performance of the model to that of the subject in simple behavioural tasks in experimental psychology. The reason is that in EVAC, long-range top-down and sub-cortical inputs are ignored, but in reality, past the reflex saccadic movements of the eyes, other mechanisms exist that direct sight. It makes it unlikely that the inactive or erroneous steady state attained and maintained in the model in some proportions for a few groups actually occur behaviourally, in real subjects. Rather, states like the incorrect or incomplete attention shifts predicted in the EVAC model will be disturbed by the constant, adaptive cortical and sub-cortical inputs to the visual streams. In practical consequence, this means that the experimental psychologist looking for evidence of the predictions below should preferably find a way to record the state of the concerned populations of neurons as the tasks are administered; or, when in sole possession of timing records, irregularities in the response time shall be hypothesised to be due to low-level failure of the network like in the model, so as the likelihood of those hypotheses may be tested by the usual means of probabilities and statistics.

The model predicts that a channelopathic visual system is unlikely to fail to activate on any of the three tasks, but has a constant 1% classification error rate. In comparison, the visual streams of control

Recent attentional engagement hastens attention shift. The relative effect is less strong in channelopathy.

Variance increased in Gap $\times$ channelopathy

Non-timing predictions can be made, but are more difficult to test.

⁹⁴ It is worth reminding that I am using attention to refer to emergent bottom-up attention, which is the recurrent activation of the neural representation of an input stimulus, emerging as a winner of lateral competitive inhibition.

subjects are predicted to only be highly unlikely to fail to activate on the Neutral task, but to have a classification error rate very close to 0 on all tasks. However, like for the reaction times, I expect such absolute activation failure and classification error rates to be scaled and/or offset, such that the clinical results only concord with the model results in relative, ordinal terms. For instance, as $r_{\zeta,ct,N} = 0$ while $r_{\zeta,ch,N} = 0.1\%$ (Table 5.6), the classification error rate of the control subjects on the Neutral task is predicted to be lower than that of the channelopathic subjects. Along the same lines,

*Classification error
and activation
failure rates
predictions*

- the control subjects' activation failure rate is predicted to be the lowest in the Neutral task, higher in the Gap task, and the highest in the Overlap task,
- in all three tasks, the control subjects' classification error rates should be lower than the channelopathic subjects',
- in both the Overlap and Gap tasks, the channelopathic subjects' activation failure rate is predicted to be lower than the control subjects'.

*Existence of
incomplete attention
shifts*

The EVAC model also predicts the occurrence of incomplete attention shifts (see p. 131) in control subjects, most likely in activation failures of the Overlap task. Such error states would probably not be present in the form of steady equilibria like in the model, but might still be discernible, timing- or state-wise. Incomplete attention shifts have not been observed behaviourally.

6.4 RESEARCH CONTRIBUTIONS

Summary

This part of the dissertation presents a preliminary theory of the possible influence of voltage-gated calcium channels in disturbed cortical dynamics, for instance, in possible cases of autism. It specifically illustrates this by comparing the performance of a computational model of pure bottom-up attention in simulated cases of L-type voltage-gated calcium channelopathy to that of the unaffected model. This conceptual model of pure bottom-up attention is enacted by neuronal accommodation and hysteresis.

*Behavioural
predictions*

This primary theory embodied by the model of emergent visual attention under channelopathy (EVAC) (Section 6.2) permits the formulation of predictions about the possible behaviours of the corresponding biological neural networks (Section 6.3). Those predictions are mostly ordinal and qualitative, due to the model simplifications.

*Novel computational
model of calcium
channelopathy*

The proposed model combines a network model of emergent reflex attention and a unit model of the effects of calcium channelopathy on neuronal dynamics. To my knowledge, it is the first network-level computational model of the calcium channelopathy theory of autism outlined in [86, 91, 92, 165, 166]. This approach has the potential to

significantly impact research on channelopathies in several respects. First, it demonstrates the utility of neurocomputational models of low-level hypotheses of autism in understanding potential interaction with higher levels of cognition, possibly up to the behavioural level – as in reflex attention shift. Second, EVAC gives avenues of experimental research to try and explore the behavioural phenotype of channelopathic individuals. Emergent processes are not intuitive, thus simulations can improve the interpretability of computational models of low-level theories of autism causation. These improvements are potentially helpful in the general effort towards understanding, and eventually curing autism.

Impact and significance of proposed model for clinical research on channelopathies

The network model of attention in EVAC is also an improvement over existing models of emergent reflex attention. As was noted in Chapter 3, there are several possible models of visual attention, some of which are able to account for bottom-up attention shift. The emergent model of reflex visual attention used in EVAC most resembles the two models of attention presented in [1], as it is build upon their principles. However, O'Reilly & Munakata [1] develop a much more simplified model of accommodation-based attention shift: input and neural maps are topologically one-dimensional and contain very few units.⁹⁵ The second model of reflex visual attention shift developed by O'Reilly & Munakata [1]⁹⁶ is structurally simpler than EVAC, and often fails to perform the shift of attention. The attention model of EVAC does not have this drawback.

Differences with [1]

6.5 LIMITATIONS OF THE MODEL

The scale of the modelled pathways and the novelty of the theories implemented have made it necessary to strike numerous compromises between model simplicity and accuracy to the biological phenomena. As a conceptual model integrating two novel hypotheses – that of autism calcium channelopathy and that of accommodation-driven low-level attention shifts – the picture drawn is approximate in the scales of many of the parameters involved, for lack of published data. For instance, the values chosen for all parameters in Table 5.1 are, for the control network, the most common values used to approximate cortical neurons in general, as found in [1], rather than more specific values that could be tailored to neurons of the visual areas being simulated. The hysteresis and accommodation values for the channelopathic networks are chosen heuristically, to have a small but noticeable effect on individual units activation (Figure 5.2). The scal-

Some parameter values are chosen heuristically or by default.

⁹⁵ The first model of O'Reilly & Munakata [1] is a didactic proof of concept.

⁹⁶ This second model of O'Reilly & Munakata is in [35, pp. 269–272].

ing of these values could well be off, negating the value of absolute quantitative predictions.⁹⁷

*Prefrontal and other
external influences
are not modelled.*

More than the reduced scale of the model, the lack of long-range top-down and sub-cortical influences in the model of emergent visual attention under channelopathy (EVAC) is among the most hampering simplifications. It puts a hard limit on the timing that is interpretable as biologically plausible, as only short duration reflexes involving the early visual areas are accounted for. This is sufficient for the type of reflex attention studied here, provided that pre-existing frontal and sub-cortical influences (other than feedback) can be neglected. For that, the subject of a clinical experiment would preferably be performing the experimental tasks in a relaxed state of mind, and be instructed not to perform any voluntary visual search. But those safeguards only help against selective prefrontal biases, but not against possible sub-cortical pathways.

*Studying a
developmental
disorder without
modelling disturbed
development can be
seen as a
methodological
limitation.*

Another limitation of the model lies at the methodological level. On one hand, the aim is to make some progress in understanding possible causes of autism, which is a *developmental* disorder with proven anomalies in the structure of the central nervous system. A key feature of autism lies in the disturbed formation of the microstructure and connectivity of the brain during growth. On the other hand, I make use of a *statically-built* neural network to study a hypothesised cause of autism by looking at short-term reaction times, without considerations of possible developmental anomalies. Indeed, the simulated network is disturbed at unit level by the means of changes in hysteresis and accommodation, but remains the same structurally in control and channelopathic cases. The distribution of units and their connectivity patterns are static⁹⁸ and determined during the design of the network by following guiding principles of cortical structure and visual streams organisation. The values of parameters, such as the minicolumns and maps sizes or the valency of units in the network graph, are set by trial-and-error, to obtain a reasonable performance on the control network. This procedure is at odds with the fact that the disturbed formation of the network structure is central to autism, and as such ought to be accounted for in predictive models.

It is worthwhile to note that such omission of developmental anomalies in neurocomputational models of autism is widespread in the computational neuroscience literature. In the case of EVAC, I did not find any model of healthy neurodevelopment sufficiently complete to approximate the formation of the visual pathways and their connectivity. Developmental neurology is a much younger field of study than structural neurology,⁹⁹ resulting in few detailed models.

⁹⁷ This is the reason why Section 6.3 lists qualitative and ordinal predictions, rather than absolute numerical ones.

⁹⁸ The Leabra learning performed by the network does not approximate structural development, as it only adjusts synaptic weights.

⁹⁹ [167] is a good overview of the field.

Without modelling the process, it could also be argued that the simulation should reflect differences in the final structure of the network that result from channelopathy – *id est*, the result of disturbed development. But the control and channelopathic models share the same structure. Two reasons justify that choice. The first reason is that structural anomalies that would be due to calcium channelopathies as hypothesised in autism spectrum disorders (ASD) have not been studied. However, autism in general tends to be linked to such anomalies, and it can be supposed that they are found in channelopathic cases as well. A priori, the evidence of minicolumnar anomalies expounded in [168] applies to the visual pathways. However, they are not applicable to the quite abstract level of detail of the model of emergent visual attention under channelopathy (EVAC). This constitutes the second reason why both networks share the same structure: I could not find studies showing consistent anomalies at a scale that would affect the principles of cortical connectivity upon which the model is built. That, in itself, reveals a limitation of EVAC: it is too abstract to account for differences in the structure of units or of networks between autistic and control subjects. It should be made clear that this issue is not proper to the model presented in this part of the thesis, but is encountered in most current computational studies of autism and other developmental disorders.

The model's point units do not allow to model anomalies in neurons structure and synaptic densities.

6.6 FUTURE EXPERIMENTAL RESEARCH

The output of a computational model of brain activity gains value in the face of the results of a corresponding psychology experiment. For the attention task, the following experimental setup is proposed.

6.6.1 *Experimental protocol*

It is hypothesised that low-level bottom-up attention mechanisms are functioning differently in autistic children (test group) as compared to neurotypical children of the same age and IQ (control group). A set of tasks is designed to help determine the conditions in which subjects show an atypical response to the appearance of attention-capturing static visual stimuli.

Tested hypothesis

Experiment description

REALISATION: The hardware setup is illustrated by Figure 6.1. A computer is connected to a display device and to an eye tracking device. The computer runs a piece of software that schedules and executes each run of the experiment. The same computer can also be used to gather data from the eye tracker.

Experimental setup

Figure 6.1: Schema of the experimental setup

Description of the tasks

The following experiments are repeatedly performed on each subject of the control and test groups. The order of the experiments is randomised and recorded. The features of targets should be in line with those of the computational experiment: simple line drawings. The relative timings of the phases of each tasks should be consistent with the relative timings of the tasks of the simulated experiment, but the exact timings should be determined by conducting a preliminary study.

*Clinical equivalent
of the Neutral task*

TASK 1 The subject is asked to look at the empty screen. After a delay, a target appears at a random location A on the screen. The time elapsed between the appearance of this target and the saccade towards A is measured.

*Clinical equivalent
of the Gap task*

TASK 2 The subject is asked to look at the empty screen. A target appears at location A, close to the centre of the subject's visual field. The target disappears, and after a short delay, a new target appears at a random location B on the screen, horizontally aligned with A. The time elapsed between the appearance of the second target and the saccade towards B is measured.

*Clinical equivalent
of the Overlap task*

TASK 3 The subject is asked to look at the empty screen. A target appears at location A, close to the centre of the subject's visual field. After a delay, a new target appears at a random location B on the screen, horizontally aligned with A. After a short delay, the first target (at location A) disappears. The time elapsed between the appearance of the second target and the saccade towards B is measured.

These three tasks correspond respectively to the Neutral, Gap and Overlap tasks of the computational experiment, described in Section 4.2 page 81.

It would also be potentially useful to collect data for comparison with future, more complex models of visual attention. Task 4 below is an example of a task that naturally extends the emergent bottom-up attention to study the formation of scanpaths.

TASK 4 The subject is asked to look at the empty screen at the centre of his visual field. Two or more targets appear at different locations on the screen. The scan path is recorded and the time elapsed between all saccades is measured. Targets features and the number of targets may vary.

Extension of the clinical study

Control of tasks conditions

The proposed experiment is not concerned with voluntary top-down attention, only with attention capture. Hence, the experiment should be designed so as not to require abstract processing or top-down visual control. Moreover, the pace of the experiment should not let the subject lose interest in the visual scene. The overall goal of instructions to be given to the participants is to prevent them from inhibiting their reflex saccades.

To control the possible effects of extraneous variables, the subjects are to be matched by intelligence quotient (IQ), age and gender between the control group and the test group. Additionally, signs of possible changes in subjects' mood (boredom, anger) have to be noted. Autistic children may get more easily distracted from the task, so making the stimuli more visually appealing by giving it, for instance, flowers or animal shapes could help. However, face-like features (eyes in particular) should be avoided for any attention-capture stimulus, due to recent evidence for direct connectivity between the LGN and the amygdala, which enables a reflex response to facial stimuli with faster activation times than on V1-mediated pathways [169].

IQ, age, gender matching

Keeping young subjects interested

6.6.2 Related prior study

Gaze tracking has already been performed on autistic subjects. van der Geest et al. have performed Posner-like experiments that closely relate to the proposed study [39]. In this work, the sample size is 16 autistic children for 15 control children. The subjects are high-functioning, and they are given the task of making a reflex saccade from a fixation point to a suddenly appearing stimulus, in two conditions: the initial fixation point disappears before stimulus onset (gap-like condition), and the two overlap temporally (overlap-like condition). Eye movement is recorded by an electrooculograph. Eyes are not tracked. The measurement of latency between stimulus onset and eye movement in both conditions reveals that the Overlap condition has a relatively less distracting effect on autistic subjects. The

Autism associated with lower level of attentional engagement

authors concluded to a lower level of attentional engagement in autistic children as compared to controls.

Limitations of [39]

Trials where no saccadic response is made are ignored, which limits the scope of the conclusions of the study, as it makes it impossible to notice a different likelihood of performing saccadic response to new stimuli. It is also notable that no data are collected about gaze direction as only an electrooculogram (EOG) is used. In an experiment involving more complex stimuli, the use of a gaze tracker could prove beneficial. The conclusions reached in the article could also be further elaborated upon with testing with more varied of stimuli, in feature size. The detection threshold could be also studied by trying to estimate the minimum stimulus exposure duration to trigger a saccade.

6.7 FUTURE THEORETICAL RESEARCH

*Extension:
generalise to cortical
networks*

There are general dynamical and structural principles that govern information processing throughout the cortex. It would be beneficial to extend and generalise the study of visual attention networks so as to obtain results that potentially apply to any cortical network. In a first step, a model needs to be created that replicates the connectivity principles illustrated in Figure 4.7. Following that, it could be interesting to train and test the network with autism-like properties on different sets of input data, with properties replicating the known properties of different sensory or more abstract signals as measured in the brain. The study would use the apparent behaviour of the network as well as the dynamics of its complete state to gather understanding of the effect of disturbances in low-level parameters in the light of neurodynamics.

*Extension: apply to
other modalities*

In a less abstract but related track of experimentation and modeling, the same type of Posner-inspired task of attention capture could conceivably be adapted to experiments and models of other sensory modalities. The auditory and tactile senses appear particularly appropriate: they are easier to control in experimental conditions, and carry a spatial location component. An auditory task inspired by the Posner test could involve exposing the subject to a primary tone in a silent environment, and determine the reaction time to the detection of a secondary tone. The channel of exposure could vary. As it is unknown whether autistic characteristics are uniform across modalities, and hence across brain areas, this experiment and the associated model aim at testing the constance of observed disturbances to neurodynamics across modalities in autism.

This model of emergent visual attention may also prove valuable in engineering, in providing a guideline towards an explicit sub-neuronal mechanism of computation that could improve the power of artificial neural networks (ANN). The discipline has long drawn inspiration from biology, but the abstraction of the computational role of

molecular mechanisms and their integration into point neuron equations is not straightforward. Neural accommodation and hysteresis may have key roles in biological systems, and this model abstracts one such possible role in map-level attentional processing. Yet, ANN have yet to take advantage of those mechanisms. I propose that future studies may implement those aspect of biological neural computation into artificial neural networks and determine their usefulness, for instance for online image processing.

Possible engineering applications

6.7.1 Methodological work

The process of building the model of emergent visual attention under channelopathy (EVAC) allowed me to identify a methodological limitation often encountered in computational neuroscience: reproducibility is limited by software dependencies. The EVAC model is implemented using the Emergent simulator, and replicating the study in another simulator requires programming the model in the new simulator's model programming language,¹⁰⁰ and implementing any functionality of Emergent not present in the new simulator.

As an immediate result, implementing EVAC in other simulators is impractical. This had direct consequences during the development of the model. When fault in the Emergent simulator hindered progress in the implementation of the model of emergent visual attention under channelopathy (EVAC), I found that using another simulator required:

- re-implementing the structure of the model in the model description language used by the new simulator, and
- implementing the functionalities of Emergent missing in the new simulator.

The major functionalities of Emergent that could not be found in other simulation frameworks included:

- Hodgkin-Huxley point neurons with channels models of unit accommodation and hysteresis, and
- the Leabra hybrid learning algorithm.

Hence, to reproduce EVAC in a different environment, I could have chosen a specific new simulator and implemented all missing functionalities, before re-implementing EVAC. However, this solution had the same potential drawback as the initial choice of Emergent: it locked down the choice of modelling environment and simulator, without improving reproducibility.

¹⁰⁰ Emergent uses C-Super-Script, a programming language designed specifically and solely for Emergent.

Instead, I decided to address the problem by implementing the missing functionalities and the model in a simulator-independent framework. Several scientific efforts exist with the goal of providing simulator-independent MDL and programming environments. Yet, none of the environments I reviewed provides the modeller with a network learning framework. Consequently, I identified that creating a simulator-independent learning framework for computational neuroscience would be a valuable contribution to the field.

Part iii of the thesis presents the result of these efforts in the form of PyCogMo, a simulator-independent learning framework for computational neuroscience. While the implementation of EVAC in PyCogMo remains a possible future work, PyCogMo has been developed into a complete and extendible learning framework.

Part III

PYCOGMO: A LEARNING FRAMEWORK FOR COMPUTATIONAL NEUROSCIENCE

In this part of the dissertation, I present a new software framework, PyCogMo, that provides computational neuroscientists with an environment to use and develop artificial neural network (ANN) learning algorithms to train simulated biological neural networks. PyCogMo fulfils a clear need in the neural modelling software ecosystem by making ANN learning algorithms available for use with several popular neural simulation programs. PyCogMo is built as a complementary overlay of PyNN, a simulator-independent language for building neuronal network models.

MOTIVATION AND BACKGROUND

7.1 REVIEW OF RELATED LITERATURE AND SOFTWARE

Computational models of neural structures are central elements of computational neuroscience. Simulation software may be used to formally describe and run simulations of individual neurons, groups of neurons, or brain areas, in isolation or encompassing their interactions. There is a wide variety of modelling and simulation software, often created to address a particular modelling need and targeting a specific hardware platform. The choice of an optimal model implementation strategy depends on the scope of the research task. For instance, the model of emergent visual attention under channelopathy (EVAC) developed in Part ii of the thesis is quite complex because it combines several models of visual pathways, attention, and LTCC channelopathies; hence, a specialised model implementation framework was used. However, small enough projects – for instance, exploratory or preliminary development – may be implemented from ground up using a general-purpose programming language, without future reuse in mind. In other cases, a simulation environment provides with model building blocks that often restrict the set of modelling formalisms usable, but allow for faster development within its conceptual scope. The implementation of the model of emergent visual attention under channelopathy (EVAC) was done using the Emergent simulation environment. Emergent was chosen based on the scale and algorithmic requirements of the EVAC model.¹⁰¹

*Computational
simulations
implementation
strategies*

Software simulation environments popular in computational neuroscience are usually not compatible with one another. This observation is hardly surprising, as significant effort is required to maintain cross-compatibility between systems. This difficulty is amplified by the lack of direct incentive to do so at the team level: simulators are often the work of one research entity, primarily developed for their own research needs. Furthermore, simulators are often free, and their source code available for modification under a permissive license for research. The distribution of models based on such simulators further diminishes the interest of creating new simulation software from

*Few immediate
incentives to develop
cross-compatible
simulators*

---

¹⁰¹ Two major criteria motivated the choice of the Emergent simulator for EVAC:

- it uses Hodgkin-Huxley neurons with channels modelling accommodation and hysteresis, and
- it allows training the network with the Leabra learning algorithm.

ground-up, while getting rid of the cross-compatibility issues that that arise when building on existing implementation-specific models.

*The difficulties of
porting a simulation
between simulators*

Whereas the diversity of free or open source simulator implementations could be advantageous in the development of reproducible, robust neurocomputational models, translating a given model's implementation from one simulator to another is often a difficult task. First, the formalisms used to describe all aspects of one model are completely different between simulators' Model Description Languages (MDL). Second, the sets of features that each simulator makes available to the implementer most often differ enough that it is necessary to implement the functionality of one simulator in the other so as to be able to cross-run a translated model. Both of these issues would be eliminated, or greatly reduced if the simulators shared a model description language (MDL) and gave access to their functionalities through a common application programming interface (API). The MDL and API used by Emergent are not modular and hard to reuse in other projects. This results in the difficulty to port the EVAC model developed in Part ii to other simulators.

Several new projects¹⁰² have been developed to facilitate the reproducible implementation of simulations and models across simulators. I call those *unifying* or *unified software*. Social projects aimed at enhancing cooperative efforts towards cross-compatibility have spun in the form of online tools¹⁰³ and software. These pieces of unifying software address the problem of simulator heterogeneity at various levels. Their development follows one of two interaction approaches with the research groups in charge of the supported simulators:

*Approaches to solve
the incompatibility
issue*

- *standardisation approach*: the developers publish a standard communication protocol, interface, or language specification, and the maintainers of the supported simulators actively keep an implementation of this interface up to its latest standard and modify the simulators when necessary, or
- *independent approach*: the developers of the unifying software specify the common language/interface/protocol *and* implement its interaction with the supported simulators, with the means available, limiting change to the simulators themselves.

The standardisation approach puts less strain on the developers of the unifying software, at the cost of reduced development swiftness, but may result in a more tightly integrated system. It involves collaborative development with strong organisational and spatial distribution characteristics, as defined by Hildenbrand et al. [170], a solution

*Pros and cons of
each approach*

¹⁰² Those include NeuroML, NineML, SBML, BioPAX, MUSIC, and PyNN, all of which are described further down.

¹⁰³ The opensourcebrain.org website gathers up-to-date information about some important tools and groups.

which increases development cost. In contrast, the independent approach lends itself more to Agile software development methodologies, as without the strong collaborative development requirements, a smaller team can tackle the problem with more independence [171]. The software produced thus is more fragile to changes in the APIs of simulators and other mechanisms relied upon, and put more strain on the developers of the unifying software. This second strategy is less scalable to larger numbers of supported simulators.

One type of such unifying software consists of model description languages which often follow the standardisation strategy.¹⁰⁴ Among the most important MDL,

Standardised MDL

- NeuroML is used for the description of compartmental, structural neurons and networks [172],
- NineML, for the structure of networks [173],
- SBML is focused on quantitative models [174],
- and BioPAX, on modelling molecular pathways [175].

The second type of unifying software takes the complementary, dynamic approach to these static languages. It consists in frameworks overarching several simulators and allowing their control in a somewhat abstract manner. The goal is to achieve a degree of abstraction that either enables easy communication between simulators¹⁰⁵ or facilitates simulating the same model on any of the supported simulators. This category of software encompasses MUSIC , used to exchange of data among parallel simulators [176], and PyNN, a simulator-independent network builder and simulation controller [4]. These frameworks are more complicated than the first type of unifying software: they interact more intricately with the supported simulators and as a result, they typically support fewer of them. This part of the dissertation presents such a novel framework (PyCogMo), built as an extension of PyNN, and aimed at making Machine Learning techniques available to across simulators for the setting of synaptic weights in networks of point neurons.

*Coordination and
homogenisation of
incompatible
simulators*

PyNN [4] is an open-source unified programming interface for multiple simulators. It improves the reliability of computational studies by providing the means to produce a simulation that can be run on several simulators: NEST, NEURON, Brian, PCSIM and MOOSE.¹⁰⁶ It provides means to create a network of neurons structured by layers in three or less dimensions, to conceptually group units as populations sending projections that follow given patterns of connectivity. PyNN uses the Python 2 programming language. Python has been adopted

Overview of PyNN

¹⁰⁴ Supported simulators are to implement the published standard.

¹⁰⁵ Large-scale simulations may be distributed among multiple heterogenous and normally incompatible simulators.

¹⁰⁶ See Table B.1 in Appendix B for a short review of popular simulators.

as the user-facing language by several major pieces of computational software in neuroscience,¹⁰⁷ thanks to its flexibility and value as both an integration language and a scientific computations package [177]. This state of the domain makes that Python is a prime candidate for the development of a unified software interface to these simulators.

7.2 OVERVIEW OF PYCOGMO

*Functionality
overview*

The new software framework presented in this part of the thesis is named PyCogMo, originally meant as an acronym of Python Cognitive Modelling. PyCogMo is, like PyNN, developed entirely in Python following an independent development approach. PyCogMo provides the modeller with Artificial Neural Networks learning algorithms to set the connectivity weights of their networks, and with the necessary abstractions to easily define their own learning algorithms. An extensible visualisation module allows visualising the structure of networks and their parameters and activity in real-time, concurrently to the running of the network. I implemented a sample 3D-rendering visualisation front-end using the Visualisation ToolKit¹⁰⁸ as rendering engine, allowing me to benefit from the state of the art functionalities of this scientific visualisation library.

*Overview of design
principles*

PyCogMo is built on principles of modularity and extensibility, defining abstractions that isolate its operational implementation from changes in PyNN. Modularity isolates implementation from function. The implementation of each module is decoupled from that of all others, thus a change to the implementation of one module only affects the internal of that same module. Several alternative implementations of one module may also exist without increasing the complexity of the overall system.

I followed a test-driven development methodology combined with short release cycles, allowing me to produce a functional framework in spite of being its sole developer.

The rest of this chapter introduces potential use cases of PyCogMo. Chapter 8 presents current applications of Machine Learning in neurocomputational models and their constraints, the design of the framework, and selected snippets of implementation. Chapter 9 discusses the development method, the contribution of PyCogMo to the current research landscape and its limitations, and concludes with a list of possible future developments.

¹⁰⁷ NEURON, GENESIS, Brian, NEST, Topographica are among the simulators offering at least a user-facing programming interface in Python.

¹⁰⁸ The Visualization ToolKit (VTK is a software system for 3D graphics and scientific visualisation [47].

7.3 POTENTIAL CONTRIBUTIONS OF RESEARCH

PyCogMo was initially developed to address the limitations of the model of EVAC in terms of reproducibility. The goal was to allow the implementation of the EVAC model using PyNN, by complementing it with the Leabra learning algorithm.¹⁰⁹ However, I realised that I was in position to better contribute to the domain, by taking a more general approach to the problem.

PyCogMo fits in an important gap in computational neuroscience modelling software. Currently, learning algorithms are reimplemented for each simulator as the need arises. This negatively affects progress in neurocomputational modelling, reproducibility of research, and their overall use in neuroscience. PyCogMo aims at solving these problems. To give a clearer view of the contributions of PyCogMo, I briefly explain the neural basis of learning, its theoretical modelling, and present the abstractions offered by PyNN, upon which PyCogMo is built.

*Potential role of
PyCogMo in the
research community*

7.3.1 Synaptic plasticity

The connections between neurons, or synapses, convey excitation and inhibition with varying degrees of efficacy, called synaptic strength. Many models of biological neural networks use a scalar to represent synaptic strength. Thus, the weight w_{AB} between pre-synaptic neuron A and post-synaptic neuron B is a number used to factor the signal propagation from A to B, representing the strength of the synaptic connections between these neurons.

*Neural basis of
synaptic learning*

Learning in the brain is usually modelled by changing the synaptic weights to give the desired input-output associative response properties to the network. In vivo, the modification of the properties of synapses has been shown to happen on several timescales in response to neural activity, and modulated by numerous factors. The simulation software supported by PyNN includes some models of the dependence of synaptic weights on neural activity. This includes spike timing dependent plasticity (STDP) based models of learning mechanisms. These models make the reinforcement (called potentiation or facilitation) and decrease (or depression) of the overall synaptic weights between the pre- and post-synaptic neuron dependent on the rates and timings of the post-synaptic spikes relative to that of the pre-synaptic spikes [178], and possibly on [179]:

- the post-synaptic membrane potential,
- post-synaptic Ca^{2+} inflow, and
- the synapse type.

¹⁰⁹ The second issue that would eventually need to be addressed is the lack of unit models implementing accommodation and hysteresis.

*Types of synaptic
plasticity*

The change in synaptic weight induced can persist less than about one second, corresponding to short-term synaptic plasticity, or much longer, up to several hours, which corresponds to long-term depression and potentiation (LTD and LTP). Phenomenological models of STDP are used in simulations using abstract scalar synapses. The longer-term, possibly life-long consolidation of synaptic strength has been modelled by the theory of *synaptic tagging and capture* [180]. However, synaptic tagging is not available in PyNN because it is not modelled by the supported simulators. More generally, it is seldom used in computational neuroscience [181].

7.3.2 *The problem of synaptic consolidation in PyNN*

*Synaptic plasticity
in PyNN*

PyNN gives a unified access to the functionalities of the supported simulators, but does not add more functionalities. In PyNN, a `DynamicSynapse`¹¹⁰ contains one instance of `ShortTermPlasticityMechanism` to model short-term synaptic plasticity, and one instance of `STDPMechanism`, for LTP/D. The `STDPMechanism` is timing-, weight- and post-synaptic voltage-dependent. There is no explicit model of synaptic consolidation.¹¹¹

To obtain consolidated weights that are unified under the PyNN framework, one would have to implement the feature in each supported simulator, for instance as a simulation of the synaptic tagging and capture theory, and create an abstraction in PyNN.

Alternately, when the goal of learning is to mimic the final weights structure of a biological network with less consideration for the phenomenal accuracy of the learning process, it is possible to manually or programatically set the weights of the network by using static synapses in PyNN.

*ANN for learning
synaptic weights in
computational
neuroscience*

With a small number of synaptic connections, random connectivity, or repetitive patterns of connectivity, it is conceivable to use a manual process or an ad-hoc script to set the network weights. It is less conceivable to do so with complex connectivity patterns that emerge from the complex interactions between inhibitory and excitatory connections such as found in most cortical areas. To model the emergence of such weights patterns, information-theoretical algorithms are developed, very often with the strong cross-disciplinary influence of artificial neural network (ANN) research.¹¹² For instance, to accurately model visual cortical topographic feature maps, the connection weights of models of visual streams are learnt using self-organising algorithms [135, 182]. Their implementation varies in complexity de-

¹¹⁰ I use monospaced fonts to denote commands, class, function and variable names.

¹¹¹ It refers to the consolidation of synaptic weight, thought to have a role in the formation of memories.

¹¹² The influence goes both ways, but ANN research formalises the information-theoretical aspects of emergent learning, which is very useful to build useful abstractions of biological processes.

pending on the desired fidelity of the model to biological feature maps. Topographica¹¹³ illustrates the complexity involved in modelling more realistic weight patterns, as the whole software package is dedicated to the self-organised building of topographical feature maps, with a primary focus on the early visual system. Some other more generic neural simulators, such as Emergent,¹¹⁴ also provide the algorithmic means to build biologically plausible connectivity between neural maps. They do that by making a library of judiciously selected ANN learning algorithms available to the modeller.

Yet, no such library is available as unified software framework. This results in the need to re-implement ANN learning algorithms when they are not present.¹¹⁵ This situation further affects the reproducibility of studies across simulators, the reusability of model implementations, and consequently, their overall value for the scientific community. PyCogMo aims at remedying this situation by:

- making a library of ANN learning algorithms available for use with several simulators,
- providing, by its well-documented modular design, the means to easily add new learning algorithms, learning rules, or combine existing ones, and
- allowing the free or open-source sharing of reusable algorithm implementations for learning synaptic weights.

*Neuroscientific
importance of
PyCogMo*

7.3.3 Scientific goals of PyCogMo

The learning algorithms developed in PyCogMo aim at producing useful approximations of the patterns of aggregated synaptic weights between neural maps observed in the central nervous system. This is a general scientific goal that is to be refined by the user (modeller) depending on the requirements of their model, but inspiration can be drawn from existing neurocomputational studies.

*The objective of
PyCogMo*

Computational neuroscientists have been making use of a number of supervised, unsupervised and hybrid learning algorithms to model synaptic connectivity.

All unsupervised learning algorithms used in computational neuroscience are based on Hebbian principles.¹¹⁶ The relatively uniform connectivity principles of cortical networks, illustrated by Figure 4.7

¹¹³ See [135] for a journal publication introducing Topographica, or topographica.org for download, documentation and latest news.

¹¹⁴ Emergent is used in Part ii of this dissertation as the main simulation framework. See [1] for a didactic presentation of PDP++, its precursor, in most parts still applicable to Emergent.

¹¹⁵ For instance, in a collection of simulators deemed appropriate based on their available neuron models

¹¹⁶ Hebb's rule formalises associative learning, in which neurons that fire at the same time see their synaptic weight increased [183].

SOM with k-WTA is the fundamental unsupervised learning algorithm in computational neuroscience.

For error-driven learning, CHL might advantageously substitute backpropagation.

page 100, make self-organised feature maps (SOM)¹¹⁷ the most popular choice. Competitive learning rules such as the k-winners-take-all (k-WTA) algorithm are considered good approximations of the effect of competitive inhibition on the formation of topographic maps of input features sensitivity [1].

Supervised learning, including reinforcement learning, is observed at the behavioural level in individuals, but its computational implementation in models of neural substrate is subject to debate. Indeed, most usual ANN algorithms implementing supervised learning assume the back propagation of an error signal. The backpropagation algorithm [160] being the most popular supervised learning algorithm, with many implementations freely available, may be used to model the results of the supervised learning tasks in neurocognitive computational models. Yet, the concept of differential error backpropagation is incompatible with current knowledge of biological neural networks information processing. Without biological basis, such implementations of supervised learning lose a lot of potential scientific value.

One alternative approach for computational modelling of learning compatible with neurobiology is presented in [154]. The study presents the Generalised Recirculation algorithm (GeneRec), and its specialised variant called contrastive Hebbian learning (CHL). These algorithms appear as biologically plausible alternatives to backpropagation. Emergent makes use of the Leabra algorithm, presented in [3], is a two-phase hybrid learning schema that combines CHL and k-WTA-driven self-organisation to model synaptic connectivity. This learning process is used in Part ii, and detailed in Section 4.3.5, page 103.

Hence, PyCogMo should implement or readily allow the implementation of:

- SOM learning, with a variety of learning rules and k-WTA / inhibition,
- backpropagation, as it is de facto used by many studies,
- CHL/Generalised Recirculation (GeneRec),
- hybrid learning algorithms, such as Leabra, as a combination of the above.

7.3.4 Design goals of PyCogMo

The design and implementation of PyCogMo follows design principles that support the realisation of the goals stated above:¹¹⁸

¹¹⁷ SOM are input topology-preserving ANN trained by unsupervised learning algorithms [10].

¹¹⁸ To recapitulate: PyCogMo should be a unified, reliable, reusable, modular framework that supports SOM, backpropagation, CHL, Leabra, and the implementation of new learning algorithms with little overhead

REUSABILITY THROUGH ABSTRACTION: Internal components of PyCogMo are reusable with minimal changes thanks to the careful design of their interfaces. The systematic use of system-wide datatypes ensures the regularity of the modules' interfaces. Higher-order functional programming also helps to keep a clear and abstract design, by enforcing conventions about the order, type and semantics of parameters of functions passed as arguments of second-order functions.

RELIABILITY THROUGH SYSTEMATIC, AUTOMATED TESTING:

Functional and, to a lesser extent, object-oriented designs lend themselves to test-driven development. The functional programming style naturally calls for live testing with sample function calls. As functions are stateless objects, the testing only consists in passing sample arguments and visually checking the output against the expected result. These natural functional programming mannerisms most easily translate into the systematic creation of test cases, by recording the live tests into a file rather than discarding them. For object-oriented design, the live testing performed during development is often more scarce, as it is subject to the maintenance of a particular state in the object being tested. Unfortunately, this also means that automated test cases should be preceded by the same setup, demanding more investment than in the functional case.

Design principles

EASY ACCESS TO THE UNDERLYING DATA STRUCTURES:

PyCogMo is a framework to help the network designer implement their own learning algorithm for a model of biological neural network. For this purpose, it is necessary to have direct access to the underlying PyNN or simulator-dependent representations of each object. Although PyCogMo is designed to be modular, with each component having a streamlined external interface, it should also be possible to access and modify the internal representations of conceptually central objects such as weights, networks, units, exemplars, neural maps, projections, or scheduling objects.

SHORT, INCREMENTAL RELEASE CYCLE: Short release cycles are especially useful when the development resources are limited. I develop PyCogMo on my own, so I have chosen a continuous release cycle. In a continuous delivery scheme, the product has to be kept in a consistent, working state at all time. Most steps that stand between implementation and release are automated. In practice, the consistency of the system is systematically determined by automated unit and integration tests, and the production of software delivery medium such as an installable package is automatic and integrated with the version control server. Thanks to this integration and automation, releasing

a new version of PyCogMo corresponds to the act of checking in an updated version of PyCogMo that passes all the automatic tests.

I created PyCogMo based on the above guiding principles. Chapter 8 presents the system and components designs, and discusses their implementation and testing.

IMPLEMENTATION AND TESTING

8.1 SYSTEM DESIGN

The system design supports the principle of separation of concerns through a choice of modules and sub-modules that are easily decoupled. I determine the value of a component as top-level module based on its the number and complexity of conceptual links with the rest of the system. The three top-level modules of PyCogMo are as follows:

`NETTRAINING` encapsulates learning algorithms, while remaining decoupled from the low-level details of their implementation in the discrete event simulator.

`SCHEDULING` implements the discrete event simulation environment. Importantly, it defines low-level events for use by the learning algorithm implementations in `nettraining`.

`VISUALIZATION` is a module that contains functions to visualise the structure and activity of the network.

Figure 8.1 schematises the top-level design of the system using Unified Modelling Language (UML) notation. The three UML package symbols represent the system modules: `nettraining`, `scheduling`, and `visualization`. The arrows that connect system modules stand for runtime dependencies. The abstraction of PyNN datatypes through PyCogMo fundamental types is encapsulated by the utilities component, called `utils` in the listings and diagrams. The `utils` component is not represented as a package because it is a convenience module, grouping public utility functions and the data types used by more than one package. It is not fundamental to the system design, but rather an implementation convenience.

On the contrary, the `scheduling` and `nettraining` modules are functionally central to PyCogMo. The former implements a Discrete-Time Simulation engine to allow the other packages to control, keep track of, and manipulate all PyCogMo events related to learning or data collection – for instance, the sampling of unit output rates for visualisation or for learning, or the clamping of input pattern for a duration. This discrete-time scheduling of the learning and data collection tasks is kept in sync with the simulator’s own scheduler, and both simulation clocks run synchronously.

The learning algorithms are implemented in the `nettraining` module. This module also provides the utilities and base structures to let the user define their own learning rules and algorithms.

*Top-level modules of
PyCogMo*

*Role of the `utils`
component*

*Role of the
`scheduling` module*

*Role of the
`nettraining`
module*

Figure 8.1: UML package diagram of the top-level system design. Arrows represent dependencies.

*Role of the
visualization
module*

Finally, the visualization module is a tool to support modelling and simulation. As such, it is not essential for network learning. It should thus be an optional component of PyCogMo. This is enabled by only having the visualization depend on other modules, while strictly no hard dependency on the visualization module can appear in other top-level modules.

The rest of the chapter discusses the individual design of these top-level modules (Section 8.2) and their test-driven implementation in Python (Section 8.3).

8.2 COMPONENTS DESIGN

8.2.1 The learning module

The nettraining module is the highest level of abstraction of the system, as it provides its essential functionality. Conceptually, it is the most remote from the interface with PyNN, relying on several abstractions. With the exception of the visualization module, the

rest of the system is designed to support the functionalities provided by *nettraining*.

The *nettraining* module relies primarily on the scheduling module to generate simulator events. The scheduling module provides *nettraining* with the service of abstracting away the events creation, manipulation, and deletion routines, as well as all the details related to the management of PyCogMo's discrete-event simulation core. Thanks to this, *nettraining* contains mostly the fine details of the learning logic, without lower-level implementation. This helps towards the design goal of creating a modular and pleasant environment for the user to implement their own learning algorithm.

Because of the very mathematical nature of the ANN learning algorithms hosted, the *nettraining* module is designed functionally. This functional design means that each callable element performs one operation with as few side effects as possible.

All learning algorithms are divided by timescale. Figure 8.2 illustrates the functional timescale hierarchy used throughout the *nettraining* module. The top-level *training function* makes repeated calls to the *epoch function* until the *stop condition* is satisfied. For each *training sample*, the epoch function calls the *presentation function*, and applies the *learning rule*. The epoch function returns the network *error metric* used by the training function to assess the stop condition.

Scheduling events may happen at all these levels, but running the simulation is deferred until the lowest level of timescale. With reference to Figure 8.2, that means that the presentation function first schedules the input presentation¹¹⁹ and data collection events. Second, after these events are set up, the presentation function runs the simulation.

If one chooses to distinguish the learning algorithm from the learning rule, the presentation function represents the lowest level of a learning algorithm's implementation, and as such is the one interfacing the most with the scheduling module.

*Relation of
nettraining with
the rest of the system*

*Functional design of
the nettraining
module*

*The presentation
function schedules
and executes events.*

¹¹⁹ Or, for the second phase of two-phase error-driven learning, it schedules input and output layers clamping

Figure 8.2: Diagram of the division of the learning functions by timescale in the nettraining module. Each box represents a functional level, implemented by a function. An arrow joining a box on the left represents a call to that function. An arrow exiting a box on the right represents the exit of the function, and the return of the control flow to the caller. Arrows exiting a box from the bottom represent function calls. The return of the execution flow to the callee is denoted by an incoming arrow at the bottom of its box.

8.2.2 The scheduling module

The scheduling module manages the discrete event simulator (DES) used internally by PyCogMo to handle data collection events and experimental training and testing events. In order to have accurate data when data collection events occur, PyCogMo's DES must be synchronised at event time with the corresponding simulation timeline in PyNN. For that, its `run_simulation` function, primarily called by the `nettraining` module, is in charge of aligning the PyNN simulator's current time with the time of the next event.

Primary roles of the scheduling module

Further, the scheduling module defines a hierarchy of event types used in the DES. These classes are the main mechanism of abstraction of the events management. They enable the programming of events by the other modules while abstracting away specific details of the functioning of the DES. Each event is characterised by its constructor and its execution method, named actions. The constructor initialises information used by the actions method. The event's actions method manipulates the network, clamps unit inputs, gathers data, or generally performs the operations described by the name of its event type. It is also conventionally the event itself which schedules, through its actions method, future repetitions of itself when necessary (e.g. recurrent measurement of units voltage).

Organisation of the scheduling module

Figure 8.3 represents the tentative organisation of classes and functions of the scheduling module. The modeller is expected to follow this convention. In this theoretical use case, two hypothetical learning algorithms named "A" and "B" have been defined in the `nettraining` module (not represented). These algorithms make use of the events defined in the scheduling module as represented in the diagram. The grey boxes labelled "A" and "B" frame the event process specifications that are specific to each learning algorithm. The event process specifications that is not in a grey box is a common resource, used by both algorithms. These process specifications are descendent classes of the `DESProcessSpec` abstract class – a hierarchy represented by the arrows with hollow triangular head, as per UML conventions. This abstract `DESProcessSpec` class is the type of object expected by the discrete event simulation engine of PyCogMo. Each process specification class is accompanied by a dedicated interface function that is used for scheduling the event. The scheduling function first performs any computation that may be necessary to create the parameters passed to the constructor of the event process specification class. It then creates the event object and inserts it in the event timeline. The scheduling function of an event is akin to its public interface.

Event classes hierarchy

Dedicated scheduling functions as interfaces to event creation

In Figure 8.3, dashed arrow lines represent the possible call of an event's scheduling function from within an event's actions method. Specialised event processes A3 and B2 are represented creating instances of the general event C through its interface function. There

Events can create other events through their scheduling functions.

Figure 8.3: Guideline for the implementation of event processes in the scheduling module

is no particular limitation on the possibilities that event processes have to create other events. For instance, event process B1 here may recreate itself during its execution.

This model is sufficiently powerful to define Hebbian type and error-driven learning. Figure 8.5 (page 171) shows the event classes corresponding to the implementation of the Leabra error-driven learning algorithm in PyCogMo using the same schematic conventions.

8.2.3 *The visualisation module*

The visualisation module provides facilities to graphically represent the structure of the network, both in terms of the connectivity between neural populations, and in terms of the spatial arrangement of units within neural maps. The module makes it possible to monitor any numerical property of units in real-time. For instance, a modeller may want to visualise inputs to individual units at the same time as their outputs, or monitor map-level aggregate statistics about net inputs. It might also be useful to be able to monitor weight changes during learning.

Primary goals of the visualisation module

Although such functionalities are not essential to the core mission of PyCogMo, the visualisation module makes mechanisms and data structures available to easily implement and execute real-time visualisation. An implementation of the real-time three-dimensional representation of the network based on the visualisation module, using the libraries of the Visualization Toolkit (VTK) is presented in Section 8.3.3 page 174.

The visualisation module is designed to allow for the implementation of real-time visualisation using any library. It is also designed under the constraint of a property of the standard Python interpreter (CPython): the simultaneous execution of more than one interpreter thread on Python bytecode is unsafe and forbidden because the interpreter's memory management is not thread-safe [184, pp. 83-84]. This constraint is named *Global Interpreter Lock* (GIL).

Visualisation is constrained by the GIL, a constraint circumvented by using a separate process.

The requirement that the visualisation module is flexible enough to let the modeller use any available graphical library for real-time representation conflicts with the constraint of the GIL. Although some graphical libraries with components written in a compiled language offer some threading capabilities outside of the Python interpreter, a lot of the information processing and conversion of the data into their graphical representations are done on main interpreter time, at the expense of simulation. To make better use of multiprocessor systems, it is thus desirable to use a dedicated process for all computations performed by the visualisation module. As a result, when using Py-

CogMo for training and visualisation, two processes are running in parallel, the simulation process and the visualisation process.¹²⁰

The visualisation module is designed to provide flexible data serialisation structures for inter-process communication (IPC) between the simulation and the visualisation processes. These data structure can easily be extended and tailored to particular graphical libraries and representations of network structure. The architecture of the visualization module in the context of the current Visualization Toolkit (VTK) backend is presented in Figure 8.4.

Figure 8.4 presents both the class hierarchy of the visualization module and the interaction of its internal components across and within its two processes. In the simulation process, the PynnToVisuAdapter and VisualisableNetworkStructure are central pieces of the architecture of the visualization module. The adapter is used to create objects that are then serialised into messages sent over POSIX pipe to the visualisation process. Two main types of objects can be used as information vectors: VisualizableNetworkStructure objects and lightweight objects containing activity update data. Typically, before the simulation is started, the PynnToVisuAdapter creates a VisualizableNetworkStructure defining the layout in 3 dimensional space of the units that should be represented, and sends it over to the visualisation process. Then, periodically, during the running of the simulation, the same PynnToVisuAdapter creates an update object containing a representation of network state and activity for the units and projections of the VisualizableNetworkStructure used. This update is serialised and sent to the visualisation process.

In the separate visualisation process, the `interpret_simu_to_visu_message` function is used to act upon receiving these serialised objects. When a VisualizableNetworkStructure is received, a new VisualisableNetwork is created. This VisualisableNetwork is the central element of the visualisation process, as it contains and updates all the graphical elements used to represent the stateful network on the screen. Hence, when an activity update message is received, the `interpret_simu_to_visu_message` passes the update object to the existing instance of VisualisableNetwork, to update the graphical representation.

When registering the elements of the network that are to be represented graphically, the interaction model with PynnToVisuAdapter takes the form of a series of transactions. Each transaction is a call to a method of the adapter to register a population or a projection that is to be represented in the graphical output. Once all elements have been added, the `commit_structure` method prepares the serialisable

*Class hierarchy and
objects interactions
in the simulation
process*

*Class hierarchy and
objects interactions
in the visualisation
process*

*Transactional
building process of
the visualised
network structure*

¹²⁰ In addition to those, the interactive VTK rendering pipeline may run in separate processes managed by VTK, and the simulator managed by PyNN may also be running in one or more separate processes.

Figure 8.4: Class hierarchy and interaction between processes in the visualization module

representation of the network structure, which also determines the representation of activity update messages.

8.2.4 *The utilities component*

The `utils` component, implemented as a Python module, contains objects that either do not belong to any other modules, or which are central elements of several modules. In practice, the distinction is made between the `utils` and the `pynn_utils` modules:

- The `utils` module regroups utility functions definitions related to file management, activity logging, data manipulation, and numerics.
- the `pynn_utils` module contains functions and classes definitions that act as interface between PyNN and PyCogMo. The contents of `pynn_utils` are more central to the functioning of PyCogMo than those of `utils`.

Several classes in the `pynn_utils` module are important building blocks of PyCogMo. I shortly review them below.

The `Weights` class provides the normalised representation of PyNN weights. A normalised representation of weights is necessary for learning and visualisation. For the purpose of normalisation, a model-dependent maximal weight range is provided by the modeller.

An `InputSample` is a representation of the static input used when clamping a layer. It allows building a mutable array of floating-point numbers from a variety of possible input sources, including files, functions and other indexable objects. It also allows to build a compact immutable representation of an input, relying on the underlying object.

The `RectilinearLayerAdapter` is used throughout PyCogMo as base class for all classes that wrap PyNN layers of neurons arranged in a grid. It provides common functionalities such as unit-indexed adapter addressing, and initialisation. The `RectilinearInputLayer` and `RectilinearOutputRateEncoder` classes described next inherits from `RectilinearLayerAdapter`.

The `RectilinearInputLayer` is a class wrapping one PyNN layer. It is used to apply an `InputSample` to the layer for a given duration. The current source is a configurable PyNN class.

`RectilinearOutputRateEncoder` adapts a topographically rectilinear PyNN map to keep track of the weighted averages on a sliding window of the output rates of all units in the population of units. It is used for learning and visualisation.

All adapters are accessed using factory functions to avoid duplication.

pynn_utils
contains PyCogMo's
representation of
network weights and
input data samples.

pynn_utils also
contains classes that
contain or generate
data homotopically
to a rectilinear
neural maps relative
to its units. These
include *RectilinearInputLayer*
and *RectilinearOutputRateEncoder*.

The `utils` and `pynn_utils` modules also contain the definition of application-level exception classes, constants, and utility functions. I do not discuss these less central parts of the implementation.

8.3 IMPLEMENTATION AND TESTING

I implemented the design described in the previous sections using Python 2.7. The language offers both functional and object-oriented programming facilities, among which *Python modules* and *Python classes*. These elements are used to directly translate the top-level modules and components design. The scientific computing `numpy` and `scipy` packages [185] are used for most computational operations, along with the Python `itertools` and `functools` modules which provide additional iterators, composition operators, and higher-order programming facilities. Iterators and list composition utilities are particularly useful as many operations are applied to collections of units. In this section, I highlight some noteworthy aspects of the implementation of PyCogMo.

Aside from PyNN, PyCogMo also uses the NumPy and SciPy scientific computing libraries.

8.3.1 Implementation of the scheduling module

The scheduling module needs to provide discrete event simulation services to the other modules.¹²¹ For that, I use the SimPy environment [186], a process-based discrete-event simulation framework for Python.

Figure 8.3.3 gives an overview of the implementation of the Leabra algorithm [3] within the scheduling module of PyCogMo. This implementation of the components of Leabra in the scheduling module follows the model illustrated in Figure 8.3 page 164: all discrete events are accessed through a dedicated interface function.

The `LeabraPhases` event process class programs two `InputPresentation` events. First, it programs the clamping of the input layer to the input pattern during the minus phase of Leabra, in which the network is tested on the sample input. Second, the `LeabraPhases` event process programs the clamping of both input and output layers to the correct patterns during the subsequent plus phase of the Leabra algorithm.¹²² It also schedules the calculation of units activities in the populations (`RateCalculation`). `InputPresentation` and `RateCalculation` are primitive events classes that are used by most learning algorithms.

Learning event processes can create other learning events.

¹²¹ See Section 8.2.2 for a description of the internal design of PyCogMo.

¹²² See page 105 for further explanations about Leabra.

Listing 8.1: The InputPresentation event process class and interface method

```

1 class InputPresentation(sim.Process):
2     def __init__(self, input_layer, input_sample, duration):
3         sim.Process.__init__(self)
4         self.name = "Presentation of " + str(input_sample) + \
5             " to " + input_layer.pynn_population.label
6         if input_layer.shape != input_sample.shape:
7             raise InvalidMatrixShapeError(input_layer.shape[0],
8                                           input_layer.shape[1],
9                                           input_sample.shape[0],
10                                          input_sample.shape[1])
11
12         self.input_layer = input_layer
13         self.input_sample = input_sample
14         self.duration = duration
15
16     def ACTIONS(self):
17         LOGGER.debug("%s starting", self.name)
18         self.input_layer.apply_input(self.input_sample,
19                                     get_current_time(),
20                                     self.duration)
21
22         yield sim.hold, self, 0
23
24     DEFAULT_PRESENTATION_DURATION = 200
25     def schedule_input_presentation(population,
26                                     input_sample,
27                                     start_t=None,
28                                     duration=
29                                         DEFAULT_PRESENTATION_DURATION
30                                     ):
31
32         global SIMULATION_END_T
33         input_layer = get_input_layer(population)
34         if start_t == None:
35             start_t = SIMULATION_END_T
36         p = InputPresentation(input_layer, input_sample, duration)
37         p.start(at=start_t)
38         if start_t + duration > SIMULATION_END_T:
39             SIMULATION_END_T = start_t + duration

```

InputPresentation and RateCalculation are fundamental event processes used by algorithm-specific events to clamp a layer and (resp.) regularly compute the time-average output of units.

The InputPresentation event process simply presents and input sample of class InputSample to a layer of class RectilinearInputLayer for a given duration. The InputPresentation class and scheduling function definitions are presented in Listing 8.1 to illustrate how an event is implemented. The InputPresentation class and scheduling function definitions are presented in Listing 8.1 to illustrate how an event is implemented. This snippet of code implements the eponymous class and its scheduling function represented in Figure 8.5. The InputPresentation process class contains two methods: its constructor (`__init__`), and the `ACTIONS` method. The constructor verifies the topographic compatibility of the input sample

Figure 8.5: Class diagram illustrating the implementation of the events supporting the Leabra algorithm

with the input layer and records both for later use. The `ACTIONS` method is called by PyCogMo's discrete event simulator to perform the scheduling of the presentation. This class is not meant to be directly instantiated by user code. The `schedule_input_presentation` helper function provides a procedural interface for that purpose.

Due to space limitations, listings for other events are not presented in the dissertation. Instead, the reader is invited to consult the original source code online at [187].

The `RateCalculation` event process is a recurrent process initiated by the `schedule_output_rate_calculation` function. It handles the `RectilinearOutputRateEncoder` object given at construction time by reading the time of the next update and scheduling the next rate calculation – in other words, by re-scheduling an instance of itself.

8.3.2 Implementation of the learning module

The `nettraining` module is designed to allow the functional implementation of high-level learning algorithms.¹²³ PyCogMo provides an implementation of self-organized k-winners-take-all (k-WTA) learning. Sample code is presented in Listing 8.2 to illustrate the functioning of the `nettraining` module. The `train_kwta` function is to be called by the user. It assumes that the trained population given in argument should have k-WTA compatible inhibition. The neighbourhood function passed as parameter takes a population and a unit as arguments, and returns a list of (unit, weight) tuples. This function is used for explicit neighbourhood learning only, not for inhibition. For the modeller, a more natural way of implementing neighbourhood learning could be to have populations with lateral excitation and several winners.

Like in the design principles presented in Figure 8.2 page 162, `train_kwta` then calls `kwta_epoch`, in which the computation of weight changes is done. Like in Figure 8.2, `kwta_epoch` uses the presentation function `kwta_presentation` for each input sample after retrieving the synaptic weights. Following the presentation, the k-winners-take-all (k-WTA) rule is applied to select the k most active units (l. 16). The neighbourhood function `neighbourhood_fn` that is subsequently applied to each of the winner units is not used for inhibition, but for neighbourhood learning only. The neighbourhood function takes a population and a unit and returns a list of units with associated weight. Hence, the `kwta_epoch` weights learning function is applied to each winner, and the learning rule is used to update the weights of its neighbours. This optional neighbourhood function is used as a fast implementation of neighbourhood learning. A slower but more accurate way of implementing k-WTA with neighbourhood

*Listing 8.2
illustrates the
implementation of
the k-WTA Kohonen
maps using the
nettraining
module of PyCogMo.*

¹²³ Refer to Section 8.2.1 page 160 for an overview of the design of the `nettraining` module.

learning consists in using populations with lateral excitation and several winners.

Listing 8.2: The implementation of k-WTA SOM learning in the nettraining module

```

1 def train_kwta(trained_population, input_population, projection,
  input_samples, num_winners, neighbourhood_fn,
  presentation_duration, learning_rule, learning_rate,
  max_weight_value, trained_pop_max_rate=None,
  input_pop_max_rate=None, min_delta_w=None, max_epoch=None):
  epoch_num = 1; stop_condition = False
3  while not stop_condition:
    delta_w = kwta_epoch(trained_population, input_population
      , projection, input_samples, num_winners,
      neighbourhood_fn, presentation_duration,
      learning_rule, learning_rate, max_weight_value,
      trained_pop_max_rate, input_pop_max_rate)
5    epoch_num += 1
    stop_condition = (epoch_num > max_epoch and max_epoch !=
      None) or (delta_w < min_delta_w)
7
def kwta_epoch(trained_population, input_population, projection,
  input_samples, num_winners, neighbourhood_fn,
  presentation_duration, learning_rule, learning_rate,
  max_weight_value, trained_pop_max_rate=None,
  input_pop_max_rate=None):
9    rate_enc = get_rate_encoder(trained_population)
    if neighbourhood_fn == None:
11      neighbourhood_fn = lambda _, u : [(u, 1)]
    max_deltaw = 0
13    for s in input_samples:
      weights = get_weights(projection, max_weight=
        max_weight_value)
15      kwta_presentation(trained_population, input_population, s
        , presentation_duration)
      argwinners = select_kwta_winners(trained_population,
        num_winners, presentation_duration)
17      for argwin in argwinners:
        main_unit = rate_enc[argwin[0]][argwin[1]][1]
19        for unit, factor in neighbourhood_fn(
          trained_population, main_unit):
          unit_index = trained_population.id_to_index(unit)
21          wv = weights.get_normalized_weights_vector(
            unit_index)
          pre_syn_out = presynaptic_outputs(unit,
            projection, t=presentation_duration)
23          pre_syn_out /= input_pop_max_rate
          post_syn_act = rate_enc.get_rate_for_unit_index(
            unit_index, t=presentation_duration)
25          post_syn_act /= trained_pop_max_rate
          new_wv = learning_rule(pre_syn_out, post_syn_act,
            wv, learning_rate * factor)

```

```

27         weights.set_normalized_weights_vector(unit_index,
           new_wv)
           max_deltaw = max(max_deltaw, max(numpy.abs(new_wv
           - wv)))
29     set_weights(projection, weights)
    return max_deltaw
31
def kwt_a_presentation(trained_population, input_population,
    sample, duration):
33     schedule_input_presentation(input_population, sample, None,
        duration)
        schedule_output_rate_calculation(trained_population, None)
35     schedule_output_rate_calculation(input_population, None)
        run_simulation()
37
def select_kwt_a_winners(population, k, presentation_duration):
39     argwinners = []
        if k > 0:
41         rate_enc = get_rate_encoder(population)
            rates = list(itertools.izip(splICE(rate_enc.get_rates(t=
            presentation_duration)), infinite_xrange()))
43         numpy.random.shuffle(rates) # randomise ties resolution
            winners = rates[0:k]
45         heapq.heapify(winners)
            for r in rates[k:]:
47             if r[0] > winners[0][0]:
                heapq.heapreplace(winners, r)
49         argwinners = [(w[1] / rate_enc.shape[0], w[1] for w in
            winners)]
    return argwinners

```

8.3.3 Implementation of the visualisation module

The visualization module implementation using VTK is a straightforward translation of its design, illustrated in Figure 8.4 page 167. The transaction model used for building the `VisualizableNetworkStructure` implies that there is a correct order in which to set up visualisation. The sample program in Listing 8.3 illustrates this process.

Listing 8.3 is a small program showing how to create and update the rendering of a network using PyCogMo.

The program in Listing 8.3 shows how the usability of PyCogMo benefits from its modular design. First, there are no unnecessary imports. Only the visualization and `pynn_to_visu` modules need to be imported (ll. 3 and 4) from PyCogMo, consistently with its use for monitoring the PyNN network during run-time.

Listing 8.3: Sample program making use of PyCogMo's visualization module

```

1 import pyNN.brian as nn
2 import multiprocessing
3 import ui.graphical.visualisation as visualisation
4 import ui.graphical.pynn_to_visu as pyNN_to_visu
5 from common.utils import LOGGER
6
7 SIMU_DURATION = 1000
8 SIMU_TO_VISU_MESSAGE_PERIOD = 100
9
10 parent_conn, child_conn = multiprocessing.Pipe()
11 p = multiprocessing.Process(target=visualisation.
12     visualisation_process_f, name="display_process", args=(
13         child_conn, LOGGER))
14 p.start()
15
16 nn.setup(timestep=0.1)
17 p1 = nn.Population(100, nn.IF_curr_alpha, structure=nn.space.
18     Grid2D())
19 p2 = nn.Population(20, nn.IF_curr_alpha, cellparams={'tau_m':
20     15.0, 'cm': 0.9})
21 prj1_2 = nn.Projection(p1, p2, nn.AllToAllConnector(
22     allow_self_connections=False), target='excitatory')
23 ... # (some steps omitted for brevity)
24
25 src = nn.DCSource(start=10.0, stop=SIMU_DURATION, amplitude=100)
26 src.inject_into(list(p1.sample(50).all()))
27
28 # Build and send the visualizable network structure
29 adapter = pyNN_to_visu.PyNNToVisuAdapter(LOGGER)
30 adapter.add_pynn_population(p1)
31 adapter.add_pynn_population(p2)
32 adapter.add_pynn_projection(p1, p2, prj1_2.connection_manager)
33 adapter.commit_structure()
34 parent_conn.send(adapter.output_struct)
35
36 # Run the simulator
37 for visu_i in xrange(SIMU_DURATION//SIMU_TO_VISU_MESSAGE_PERIOD):
38     nn.run(SIMU_TO_VISU_MESSAGE_PERIOD)
39     parent_conn.send(adapter.make_activity_update_message())
40 if SIMU_DURATION % SIMU_TO_VISU_MESSAGE_PERIOD > 0:
41     nn.run(last_chunk_duration)
42     parent_conn.send(adapter.make_activity_update_message())
43 nn.end()
44 # Wait for the visualisation process to terminate
45 p.join(VISU_PROCESS_JOIN_TIMEOUT)

```

PyCogMo's visualisation environment can be used to produce still representations of the network, but its use as a real-time visualisation tool requires the import of Python's built-in multiprocessing package, and the preliminary fork of a child process (ll. 11–12). This

Figure 8.6: Simple interface build using PyCogMo’s visualization module

new process that runs the `visualisation_process_f` function of the `visualization` module, communicates with the main PyCogMo process using the POSIX pipes created at l. 10.

After setting up a simple PyNN network and simulation (ll. 14–21), a `PynnToVisuAdapter` is created (l. 24) and used to record the simulation elements to be displayed (ll. 25–28). The generated `VisualisableNetworkStructure` can then be send to the child visualisation process (l. 29).

The last section of code in Listing 8.3 illustrates a possible use of the visualisation facility: the full simulation is divided into sections of equal duration (l. 32), and a visualisation update message is sent to the child process after running each simulation section (l. 34).

This short example is meant to be technically illustrative rather than scientifically useful, hence several procedures are unrolled that would normally be provided by a library function: the setting up of the child process and IPC (ll. 10–12), and the registration and sending of the whole network structure¹²⁴ (ll. 23–29).

Figure 8.6 illustrates the use of the `visualization` module for visualisation of network activity within a simple graphical interface. In this particular example, the VTK renderer used by the `visualization` module is integrated using PySide in a graphical user interface designed with QT 4.8.¹²⁵ Two push buttons allow running and inter-

¹²⁴ Explicit calls to these primitive `add_` and `commit_` methods should only appear in user code when the user needs fine-grained control over what is to be graphically represented.

¹²⁵ Please refer to qt-project.org and pyside.org for documentation.

rupting the simulation, and the status bar displays the current epoch of the simulation.

8.3.4 Implementation of the utilities module

The utilities module is less structured than the other modules, because it consists in a collection of fundamental classes that support the other, more structured modules. However, I would still like to highlight the `get_input_layer` and `get_rate_encoder` functions, as they are interesting uses of partial function application in Python (Listing 8.4).

Listing 8.4: Definitions of the `get_input_layer` and `get_rate_encoder` functions by partial application of `population_adapter_provider`.

```

1 import functools
2
3 def population_adapter_provider(pop_prov_dict,
4                                provided_class,
5                                population):
6     key = (population, provided_class)
7     if pop_prov_dict.has_key(key):
8         return pop_prov_dict[key]
9     else:
10        dim1, dim2 = rectilinear_shape(population)
11        inst = provided_class(population, dim1, dim2)
12    return pop_prov_dict.setdefault(key, inst)
13
14 POP_ADAPT_DICT = {}
15 get_input_layer = functools.partial(population_adapter_provider,
16                                    POP_ADAPT_DICT,
17                                    RectilinearInputLayer)
18 get_input_layer.__doc__ = ("Provides a unique input layer for the
19                             given population.")
20
21 get_rate_encoder = functools.partial(population_adapter_provider,
22                                     POP_ADAPT_DICT,
23                                     RectilinearOutputRateEncoder)
24 get_rate_encoder.__doc__ = ("Provides a unique rectilinear output
25                             rate encoder for the given population.")

```

This sample code from the `pynn_utils` module is a good illustration of the use of Python as a functional language.

It can be seen in the listing above that the `population_adapter_provider` function defined on ll. 3–12 is a factory function providing an adapter of the specified class for the population given in parameter. The `get_input_layer` and `get_rate_encoder` functions are produced by the partial applications of the `population_adapter_provider`, on ll. 15 and 20 respectively.

8.3.5 Testing

Automated testing follows a simple naming convention.

To ensure the correct operation of PyCogMo at all times, I use automated testing. Unit and component tests are written in a test module, with each submodule testing a module of PyCogMo. A test is conventionally defined as a function with a name starting with `test_` and defined in the test module or one of its submodules. This allows the automatic discovery of tests, a feature of the *nose* testing framework.¹²⁶

Testing is rendered even more important by the continuous delivery process, as each code change published should not affect the usability of PyCogMo. To preserve such consistency, new commits to the version control system (VCS) are guarded by the automatic running of the whole series of unit and component tests, such that a new commit to the VCS is only allowed when all tests pass.

Automated testing of individual functional units is straightforward; for the rest, the nose framework provides state management facilities.

Testing individual components is easier when their design is functional, as the expected result of an operation solely depends on the input parameters. Thus, my initial design choices proved useful. However, while unit tests are facilitated by pure functional design, this simplicity is not in order when the internal state of mutable objects or the side effects of functions are studied. Similarly, integrated component tests and system tests, where the interaction between stateful objects is tested, do not benefit as much from functional unit design as unit testing does. In those cases, the maintenance of a controlled environment and the expression of dynamic constraints can be tedious tasks. Several testing tools provided by the *nose* library help in this regard, including the ability to create *mock objects* and to execute *setup* and *teardown* functions before and after test code in order to control the execution environment.

The last chapter builds on this overview of PyCogMo design and implementation to discuss its capabilities, limitations, and possible future developments.

¹²⁶ More information about *nose* is available at nose.readthedocs.org.

DISCUSSION

This chapter concludes Part iii of the dissertation. It first presents the current capabilities, performance and limitations of the Python Cognitive Modelling framework (PyCogMo). Future plans to address these limitations and development directions are then discussed.

9.1 ISSUES AND LIMITATIONS

Sections 8.1 and 8.2 explained the design principles guiding the implementation of the overall system and of each of its modules. Although automated testing (see Section 8.3.5) limits the number of software faults due to implementation errors, it neither tests the design or the system nor its computational performance. Yet, the system design system needs to be critically assessed in regard to its goals, and the performance characteristics and limits should be known. This section reviews these two aspects of PyCogMo.

Automated tests do not find design flaws → design and performance need to be reviewed.

9.1.1 System design

One of the design goals of PyCogMo is modularity. However, the implementation does not entirely respect that: there is a *leakage of abstraction* between some modules. For instance, part of the logic about input presentation is contained in the InputPresentation discrete event simulator (DES) process class in the scheduling module. However, this logic conceptually belongs to the nettraining module. The issue with finding the right module to define these events is two-fold:

The principles of modularity and separation of concerns are not entirely respected.

- The nettraining module requires a particular design because it is the conceptual space where a modeller would implement their own learning algorithms. That space should be clear of scheduling-related implementation details that distract from the implementation of the learning algorithm.
- The opposite solution, consisting in defining the event classes in the nettraining module, also amounts to a leakage of abstraction; this time in the other direction, with implementation details of the scheduling module appearing within nettraining.

The possibility also exists to design an extra mechanism to abstract the creation of event types, for instance using generator functions. However, this type of design makes the program larger. It would increase the complexity of PyCogMo, and the event type generator

Design flaws can be addressed at the cost of increased complexity.

may not be able to cover all potential use cases of PyCogMo DES events for learning. Allowing the presence in the scheduling module of specialised nettraining events is a choice that I made to keep the design simple and straightforward, making the implementation a less error-prone process. Furthermore, a strict conceptual isolation of the nettraing and scheduling modules is not necessary. As explained in Chapter 8, it is only a strict requirement that no core module of PyCogMo relies on the visualization module; for other modules, the modular design is more of a guideline that can be compromised in favour of other design guidelines.

PyCogMo depends on private, unstable APIs of PyNN, increasing maintenance overhead.

Aside from these design inconsistencies, another potential weakness of PyCogMo lies in its reliance on internal, *private* APIs of PyNN. As PyNN is at the very base of PyCogMo, it is necessary that PyCogMo relies on the functionalities of PyNN. PyNN, as a modelling framework, offers a library that makes functions and objects accessible using a *public* and stable API.¹²⁷ However, this *public* API is geared towards running the simulation of networks without frequent external modification during the simulation.¹²⁸ One consequence is that some components of PyCogMo that interact with PyNN do so via objects and methods that are not part of the *public* PyNN API, but part of its internal, *private* API. It is, for instance, necessary to directly access some inner attributes of PyNN Projection instances, for performance reasons, because API methods either do not provide such access, or are too slow for repeated operations.

Maintaining software compatibility with new versions of PyNN is difficult.

These implementation-specific dependencies result in the fragility of PyCogMo to changes in the implementation of PyNN. For instance, at the time of writing, PyCogMo is compatible with the current version of PyNN (0.7), but not with the development version (0.8 α). Often, from one version of PyNN to the other, changes are required to the interface between PyCogMo and PyNN, making it harder to maintain software compatibility than if relying solely on the public API of PyNN.

Abstraction through PyNN and PyCogMo's own data types has computational costs.

A second consequence of PyCogMo's reliance on PyNN lies in the overhead brought by the increased number of software intermediaries. To perform training of a network, PyCogMo needs to:

- access and update the network weights after a few training samples presentations – each weight adaptation operation makes use of PyNN's `Weights` representation, converted to the proper scale for PyNN. This `Weights` class of the `utils` module is used to access and modify the weights representation in PyNN's `Projection` class. In turn, PyNN accesses and modifies the corresponding representation in the underlying simulator.

¹²⁷ The specification of the public PyNN API is published on neuralensemble.org.

¹²⁸ To a certain extent, this limitation reflects the assumptions of the underlying simulators.

- create an internal representation of each training exemplar, with a PyNN adapter attached to the concerned input layers and other clamped layers. PyCogMo then needs to activate this PyNN representation of the exemplar in the beginning of the presentation, and deactivate it the end. By default, the PyNN representation of a training exemplar uses a collection of PyNN DCSources objects. The steps of creating, activating and removing these simulated current sources m times per epoch for an m -sized training sample add a lot of overhead computations.

9.2 RESEARCH CONTRIBUTIONS

Besides these limitations, PyCogMo offers a novel convenient modular framework for learning in biological neural networks. Thanks to the abstraction offered by PyNN, the learning algorithms do not depend on the specific implementation of a simulator, but can be applied to many. Hence, the major contribution of PyCogMo is to offer a *common learning framework* for NEURON [43], NEST [44], PCSIM [45], Brian [46], as well as any simulator that will be supported by PyNN in the future.¹²⁹ In addition, the implementation of all core functionalities of PyCogMo is systematically tested to minimise the risk of software fault.

PyCogMo is a novel learning framework compatible with several simulators, and allows visualisation of the network being modelled.

The second contribution of PyCogMo lies in the *visualisation module*. However, the VTK-based implementation of the visualisation module is still a prototype, in that it offers few control over the display settings.

9.3 FUTURE WORK

There are several possible future development paths for PyCogMo. Due to the specificities of the project, these possible roadmaps do not only depend on my vision.

One factor that directly influences the development of PyCogMo is the evolution of PyNN and of the underlying simulators. While features of PyCogMo that are already implemented are stable because they rely on quite fundamental functionality in PyNN, future progress will be influenced by new functionalities made available through the development of PyNN.

The evolution of PyNN will influence PyCogMo.

The second factor that should guide the development of PyCogMo is community demand. As a framework, PyCogMo is meant to fulfil the needs of its users. So, a feature can only be justified if there is

Promotion of PyCogMo in the research community

¹²⁹ However, PyCogMo has only been tested with the NEST and Brian backends. There should not be any major obstacle to getting it to work on the other simulators supported by NEST.

potential interest in it¹³⁰. It is an advantage that PyCogMo is freely available and redistributable under the GNU General Public License, because any feature for which there is a demand in the computational neuroscience community can easily get added to PyCogMo, to the benefit of all.

Aside from community-driven development, there are a number of features that I would like to add to PyCogMo.

*Integration with
existing Python
ANN learning
libraries*

- The possible integration with one or several existing ANN learning libraries should be considered.¹³¹ In this case, PyCogMo would act as a bridge between the learning library and the simulation software. It would allow to directly benefit from all self-organising algorithms implemented, but gradient-based (error-driven) learning algorithms may not all be portable to biological neural networks, depending on the neuron model's activation function approximation. For spiking networks, consideration should also be given to the *SpikeProp* algorithm for error-driven learning [188].

*Relaxing the
constraint of
rectilinearity of
populations*

- Currently, PyCogMo is only designed to manipulate topologically rectilinear maps of neurons. Although this is a common modelling simplification, PyCogMo would gain in versatility if it gave modellers the possibility to handle arbitrary population topologies. This may be done by designing internal map representations and population adapters with unit and sub-population addressing mechanisms that are adapted to arbitrary topologies. With population representations that do not make topological assumptions, PyCogMo would allow the development of new, more versatile algorithms for cortical maps learning.

*Further developing
the visualisation
module into a GUI*

- Developing the visualisation module past its current prototype stage could help with the task of network modelling. The goal of these developments would be to make the visualisation easy to directly use without configuration for the most common use cases. These use cases include: displaying the whole network, interacting with each unit to show its properties or the weights of its incoming projections, and showing units activations when the simulation is running. The current visualisation module allows a lot of flexibility in choosing what to display, but few interactions with the network. Only viewport navigation is possible (pan, rotate, tilt, zoom). A simulation control interface could also be convenient.

¹³⁰ The features and algorithms currently available were justified by my interest in them. I developed them in reaction to difficulties I encountered when creating the model of emergent visual attention under channelopathy (EVAC) presented in Part ii of the dissertation.

¹³¹ Popular Python Machine Learning libraries that are potentially useful to train ANN include PyBrain, mlpy, Shogun, MDP toolkit, Orange, PyMVPA, and Monte.

In addition to these features, it would certainly be beneficial to work on improving the performance of the learning module and make PyCogMo easy to install for novice Python programmers. On one hand, the first point partly depends on community contributions to PyCogMo, as performance improvements mean tighter integration with internal components of PyNN, which require more time investment into development. On the other hand, improving the ease with which PyCogMo can be installed used could be an easier first task that may help increase the interest of the neurocomputational modeling community in PyCogMo.

*Improving the
performance of the
learning module*

Part IV

APPENDICES

PRINCIPLES OF ASTRUCTURAL NEURON MODELLING

Since the advent of Parallel and Distributed Processing (PDP) [26], runnable computational models of brain structures have flourished, with different modelling goals reflected by various abstraction from actual neural mechanisms. But regardless of the level of abstraction, connectionist models make use of atomic individual computational units modelled after the abstract properties of neurons or groups of neurons. This chapter introduces these abstractions, which are the cornerstone of computational neuroscience.

A.1 FROM EQUIVALENT CIRCUIT MODELS TO THE HODGKIN-HUXLEY MODEL

Neuronal information processing is an integrative electrochemical process. Figure A.1 illustrates that with a schematic neuron. A neuron is typically composed of a cell body (*soma*), of an axon that is the main output organ of the neuron, and of a dendritic tree where other neuron's axons connect via synapses, where information is transmitted by triggering synaptic currents in the dendritic tree of the recipient neuron. In this schema, the dashed line on the left represents the axon of some pre-synaptic neuron, and the dashed line on the right, the axon of the schematised neuron. The electrical currents that are transmitted from the dendrites to the soma, integrated, and further propagated into the axon as action potentials (APs)¹³², are local to the cell membrane. The information is propagated under the form of a local depolarisation of the membrane.

Figure A.1: Schema of the electrical information processing of a neuron

¹³² *Spikes*, another term for APs, are said to be *fired* by the neuron.

A.1.1 Simple electrical circuits

Model of neurons make use of abstract electrical components to form abstract electrical circuits, following the conventions of electrical engineering. The three most important components of these models are:

BATTERY a source of electrical potential E measured in Volt (V).

CAPACITOR an electrical insulator that accumulates charge on either side. Capacitance C is measured in farads (F). It is equal to the amount of electrical charge (in coulomb) divided by the voltage: $C = \frac{q}{V}$.

RESISTOR a component that resists the flow of current. Resistance is denoted by the letter R and measured in Ohm (Ω); its mathematical inverse, the conductance g , is measured in Siemens (S): $g = R^{-1}$.

The current flowing through the wires, measured in Amperes (A) is a signed quantity denoted I (the current flowing through something named α is denoted by I_α). The relation between current, electrical potential and resistance is given by Ohm's law: $V = IR$ or $I = Vg$ (conductance version).

The abstract circuit is made of perfectly conducting wires connecting these components. Kirchhoff's current law further states that the total current at a junction of such wires is zero.

A.1.2 Modelling a patch of membrane

With these tools, we can model the electrical properties of a small patch of the membrane of the neuron. The membrane separates the inside of the cell from the extra cellular medium that surrounds it, keeping a difference of concentration in various species of ions, and thus maintaining an electrical potential.

Membrane alone

The membrane is slightly permeable to the surrounding ions. Without ion channels, the circuit model of a patch of membrane is a resistor (for insulation) and a capacitor (for slight permeability) in parallel, like in Figure A.2.

As information is propagated within the neuron in the form of a depolarisation of the membrane, it is necessary to know the expression of the change in voltage across the membrane when an external current I_{ext} is injected in the circuit:

Figure A.2: Circuit model of the electrical properties of a patch of passive lipid bilayer membrane

Ohm's law $V = I_R R$ gives the current I_R passing through the resistor, and the capacitor definition $Q = CV$ gives the current I_C that goes through the capacitor. By deriving, we obtain:

$$\frac{dQ}{dt} = C \frac{dV}{dt} \quad (\text{A.1})$$

and as $I_C = \frac{dQ}{dt}$, we have

$$I_C = C \frac{dV}{dt} \quad (\text{A.2})$$

Using Kirchhoff's first law $I_{\text{ext}} + I_C + I_R = 0$, we get the expression of the change in membrane voltage, a first-order linear differential equation:

$$C \frac{dV}{dt} = -\frac{V}{R} + I_{\text{ext}} \quad (\text{A.3})$$

However, this model is insufficient for the purpose of understanding membrane dynamics in a neuron, because the membrane interacts with the ions through ion channels and pumps.

Membrane with ion pumps

The membrane's environment and default behaviour are as follows:

- higher K^+ , inside the cell,
- higher Na^+ , Cl^- , Ca^{2+} outside the cell,
- active ionic pumps exchanging Na^+ for K^+ to maintain the osmotic and electrical gradient,
- the osmotic potential results in an electrical potential because the ions move throughout the membrane following their concentration gradient until opposed by electrical forces.

The voltage at which osmotic and electrical forces are at equilibrium is called *Nernst potential*. It is the logarithm of the ratio of ionic concentrations multiplied by an expression of the universal gas constant k , the temperature T , the electric charge z and the Faraday constant F :

$$E = \log \left(\frac{\text{ions out}}{\text{ions in}} \right) \frac{kT}{zF} \quad (\text{A.4})$$

These pumps maintaining a difference of potential are modelled with an extra battery in the electrical circuit model, like shown in Figure A.3.

Figure A.3: Circuit model of a patch of cell membrane with an active ion pump

E represents the battery with a voltage of V_{rest} , the voltage of the membrane *at rest*, i. e. in its default state. With that battery, the voltage drop across the circuit is lesser, as part is accounted for by the battery, and part by the resistor. The expression of the change in voltage is now:

$$C \frac{dV}{dt} = -\frac{V - V_{\text{rest}}}{R} + I_{\text{ext}} \quad (\text{A.5})$$

We define the membrane constant $\tau = RC$, transforming Equation A.5 into:

$$\tau \frac{dV}{dt} = -V + V_{\infty} \quad (\text{A.6})$$

where V_{∞} is the steady state value of V .

In case of constant external input current I_{ext} , $V_{\infty} = V_{\text{rest}} + I_{\text{ext}}R$.

In case of a square pulse, (step-wise constant current input), the solution to the membrane voltage equation is

$$V(t) = \begin{cases} V_{\infty} \left(1 - e^{-\frac{t}{\tau}}\right) & \text{when there is current input} \\ V_{\infty} \left(e^{-\frac{t}{\tau}}\right) & \text{when there is no current input} \end{cases} \quad (\text{A.7})$$

Equation A.6 is sufficient to characterise the membrane in its default state, but does not model the mechanisms that make the membrane an information processor. The local depolarisation and repolarisation of the membrane happens during the generation and propagation of action potentials and of local changes in voltage thanks to ion-selective membrane channels (proteins) that can be voltage-gated, chemically-gated (binding to a molecule causes opening), or mechanically-gated (pressure or stretch). Voltage-gated ion channels are ubiquitous, and give the membrane its non-linear reactivity to depolarisation. They need to be modelled as well.

Ion	Potential
Na ⁺	50 mV
Ca ²⁺	150 mV
K ⁺	-80 mV
Cl ⁻	-60 mV

Table A.1: Nernst potential of major ions in the neuron's environment

Membrane with voltage-sensitive ion channels

The current flowing through one ion channel can be determined by Ohm's law, using the voltage across the membrane and the conductance of the channel: $I = Vg$. Each ion channel has its own conductance. The higher the conductance for one ion, the more the membrane potential is pulled towards the equilibrium potential of that ion, as defined by the Nernst potential. Table A.1 recapitulates the Nernst potentials of the ions used in most neuron models.

Each type of ion channel i is modelled by a resistor with conductance g_i in series with a battery with potential E_i :

$$I_i = g_i(V - E_i) \quad (\text{A.8})$$

In the model being developed, only sodium and potassium currents are explicitly modelled. Na⁺ currents tend to depolarise the membrane, and K⁺ currents tend to repolarise it.

Computing power at the ionic level: channel nonlinearity

It is possible to trace some of the non-linearities that give the neurons their computing power down to the experimental measurements and simulations done at ionic level. As one increases the input current, up to a certain point, the response of the membrane voltage scales linearly; however, past a given intensity, the system shows its excitability, as the membrane depolarises disproportionately. The following model reveals the mechanisms behind that non-linearity.

Figure A.4 is an equivalent model of a patch of membrane with two voltage-gated ion channels, calcium and potassium, and one passive leak channel with subscript "L", which stands for "leak"; it is a generic passive ion channel like the passive channels seen previously, e. g. in Figure A.3.

This circuit is now active, in that the conductances of the Na and K channels are now variable: g_{Na} and g_{K} depend on the voltage. We need to model this dependence for each type of channel

Figure A.4: Circuit model of a patch of cell membrane with voltage-gated sodium and potassium ion channels and one generic leak channel.

MODELLING POTASSIUM VOLTAGE-GATED CHANNELS Voltage-gated ion channels contain a molecular gate¹³³ whose opening determines the activation of the channel. The probability that the molecular gate of a voltage-gated channels is open increases with depolarisation (for K^+ channels) or polarisation (for Na^+ channels). In K^+ channels, this probability P_K depends on the configuration of four independent sub-units of the channel; $P_K \propto n^4$.

Each of the channel's subunits fluctuates between open (probability n) and closed ($1 - n$) states, and current only flows through the channel when all four units are open at the same time. It is to be understood that n does *not* directly depend on the voltage. Rather, the *transition* of each sub-unit between open and closed state occurs at a voltage-dependent rates $\alpha(V)$ for the closed-to-open transition and $\beta(V)$ for the open-to-closed transition. With that, n varies as such:

$$\frac{dn}{dt} = \alpha_n(V)(1 - n) - \beta_n(V)n \quad (A.9)$$

Equation A.9 describes the rate at which the open probability for a subunit gate changes, as $(1 - n)\alpha_n(V)$ is the opening rate times probability of finding the gate closed, and conversely, $n\beta_n(V)$ is the closing rate times probability of finding the gate open.

The voltage-dependence of α and β allows for a positive feedback loop on the transition rate (closed $\rightarrow$ open and open $\rightarrow$ closed), as a higher probability n of opened sub-unit lets more positive charges enter the cell, which increases the voltage. An increased voltage means a higher subunit transition rate, so this process feeds back on itself.

¹³³ This is not an abstraction, but a physical gating mechanism.

Reformulating Equation A.9 at steady state (i.e. for a fixed V), we obtain the rate equation for the gating variable $\frac{dn}{dt}$:

$$\tau_n(V) \frac{dn}{dt} = n_\infty(V) - n \quad (\text{A.10})$$

where

$$\tau_n(V) = \frac{1}{\alpha_n(V) + \beta_n(V)} \quad (\text{A.11})$$

and

$$n_\infty(V) = \frac{\alpha_n(V)}{\alpha_n(V) + \beta_n(V)} \quad (\text{A.12})$$

Equation A.10 is the normalised form of the potassium voltage-gated channel subunits transition rate equation, which we will use together with the sodium voltage-gated ion channel rate equations to write the Hodgkin-Huxley model conductance-based model of action potential generation.

MODELLING SODIUM VOLTAGE-GATED CHANNELS The Na^+ channel contains three gating subunits that work on the same principles as the subunits of the potassium channel. It also has an additional second gating mechanism that works in the opposite manner: it has to be de-inactivated.

Let m the probability of the sodium gate subunits to be open, and h that of the additional inactivation gate being open. Then, $P_{\text{Na}} \propto m^3 h$.

Higher voltage increases m (making it an *activation variable*), but decreases h (an *inactivation variable*). As a result, sodium flows are transient and self-limiting.

Since these two types of subunits obey the same principles as the gating sub-units of the potassium channel, their voltage-dependent transition rates are used in similar voltage-gated channel subunits equations as Equation A.9, giving the following overall system:

$$\begin{cases} \frac{dn}{dt} = \alpha_n(V)(1 - n) - \beta_n(V)n \\ \frac{dm}{dt} = \alpha_m(V)(1 - m) - \beta_m(V)m \\ \frac{dh}{dt} = \alpha_h(V)(1 - h) - \beta_h(V)h \end{cases} \quad (\text{A.13})$$

where $\alpha_m(V)$ and $\alpha_h(V)$ are the closed-to-open transition rates of the activation and inactivation gates, and $\beta_m(V)$ and $\beta_h(V)$ are those for the open-to-closed transition.

Using steady state values for a fixed V :

$$\begin{cases} \tau_n(V) \frac{dn}{dt} = n_\infty(V) - n \\ \tau_m(V) \frac{dm}{dt} = m_\infty(V) - m \\ \tau_h(V) \frac{dh}{dt} = h_\infty(V) - h \end{cases} \quad (\text{A.14})$$

This system of equations, integrated with the laws of Electricity introduced earlier, is sufficient to obtain an expression of the change in membrane voltage of an active voltage-dependent patch of membrane, giving the analytical expression of the Hodgkin-Huxley model of action potential propagation [189].

THE HODGKIN-HUXLEY MODEL Combining the individual ion channels models developed earlier, all voltage-dependent conductances g_i can be expressed as functions of maximal conductances $\bar{g}_i$:

$$\begin{cases} g_K(V) = \bar{g}_K n^4 \\ g_{Na}(V) = \bar{g}_{Na} m^3 h \end{cases} \quad (\text{A.15})$$

Using the expression of the capacitive current¹³⁴, the sum of ionic currents going through channels $-\sum_i g_i (V - E_i)$, and the external stimulus I_e , we have, by applying Ohm's and Kirchhoff's laws:

$$C_m \frac{dV}{dt} = - \sum_i g_i (V - E_i) + I_e \quad (\text{A.16})$$

Together with channel sub-units rate equations (A.13), Equation A.16 forms the Hodgkin-Huxley model as the system:

$$\begin{cases} \frac{dn}{dt} = \alpha_n(V)(1-n) - \beta_n(V)n \\ \frac{dm}{dt} = \alpha_m(V)(1-m) - \beta_m(V)m \\ \frac{dh}{dt} = \alpha_h(V)(1-h) - \beta_h(V)h \\ -C_m \frac{dV}{dt} = g_L (V - E_L) + \bar{g}_K n^4 (V - E_K) + \bar{g}_{Na} m^3 h (V - E_{Na}) - I_e \end{cases} \quad (\text{A.17})$$

The time constants $\tau_n(V)$, $\tau_m(V)$ and $\tau_h(V)$ determine the speed at which n , m and h reach their values for a certain voltage. They are plotted on Figure A.5. The graph of $n_\infty(V)$, $m_\infty(V)$, and $h_\infty(V)$ in function of the membrane voltage (Figure A.6) shows to which limit value each subunit probability converges. Together, these two graphs reveal the probable sequence of gates opening and closing during a depolarisation. As the faster-reacting variables are those with lower time constants, m (corresponding to sodium) is the fastest to respond to a change in voltage. The equilibrium potentials of sodium and potassium are at one end and the other of the graphs,¹³⁵ so, from a

¹³⁴ in the circuit model in Figure A.4 $C_m \frac{dV}{dt}$, the current going throughout the membrane capacitor C_m

¹³⁵ $E_{Na^+} = 50$ mV, and $E_{K^+} = -80$ mV, from Table A.1 – it is useful to take mental note of the resting potential of each ion on the graphs, so as to visualise the direction of the influence of the opening of that channel.

Figure A.5: Plot of the time constants of the subunits at steady state τ_n (continuous line), τ_m (dotted line), and τ_h (dashed line) in function of membrane potential

resting potential of -60 mV, a depolarization due to an external input current (or to a synaptic transmission) will trigger a fast response of the m parameter, shifting it towards $m_\infty(V)$ faster than other parameters. As m is the sodium activation variable, this fast increase in m means that the immediate response of the membrane to the external depolarising current is to shift the membrane potential towards the sodium equilibrium potential of 50 mV. As the dotted line reveals on Figure A.6, the increase in membrane voltage further increases the activation probability of m subunit. This positive feedback loop corresponds to the fast-rising initial depolarisation of the action potential, and lasts until the slower subunits, with time constants $\tau_n(V)$ and $\tau_h(V)$, start to have a sufficient effect to counteract it. As the activation probability of h (dashed line on Figure A.6) is an inverted sigmoid, its effect at higher voltages is to lower the overall probability of the sodium channel to be open, reducing depolarisation. At the same time, as the activation rate of n increases with a delay of the same order of magnitude, the potassium channel opens, bringing back the membrane potential towards $E_{K^+} = -80$ mV. This corresponds to the repolarisation and hyperpolarisation of the membrane potential, in the second phase of action potential generation.

Figure A.6: Plot of the subunits steady state activation and inactivation variables n_{∞} (continuous line), m_{∞} (dotted line), and h_{∞} (dashed line) in function of membrane voltage

The Nobel prize-winning Hodgkin-Huxley model of neuron firing is an elegant formulation of the molecular principles underlying voltage-gated sodium and potassium channels.¹³⁶

This model is an historical cornerstone of computational neuroscience. As such, it has served as the basis to develop other models of neurons. Some of these newer models are simpler, trying to abstract even more from the physical processes to only keep the abstract fundamental dynamics. Others are more complex models, taking more of the biochemistry and geometry of the neuron into account. Section A.2 that follows reviews some of the simplified models, as the models used in this thesis fall in this category.

A.2 FROM HUDGKIN-HUXLEY TO SIMPLIFIED MODELS

There are two main incentives to simplify the Hudgkin-Huxley model:

- analytical tractability: working formally on the mathematical properties of neurons may require to strip the mathematical formulation from details that are irrelevant to the problem studied.

¹³⁶ This formulation of the gating equation was a very impressive feat, as Alan Hodgkin and Andrew Huxley developed the model without knowing the existence of independent gating mechanisms or ion channels.

- computational tractability: the Hodgkin-Huxley model can be computationally expensive to simulate, in particular in large-scale simulations.

A.2.1 The Integrate-and-Fire model

When recording the firing patterns of different biological neurons, or even of the same neuron under different depolarizations, different firing patterns are noticeable, including:

- *bursting*, for instance in thalamic neurons when more depolarised,
- *irregular spiking*, for example of thalamic neurons when less depolarised,
- *intermittent regular spiking*, observed in many cortical neurons, and
- *regular continuous spiking* with slight variations in spike timing, which could be recorded from e.g. motor neurons.

The integrate-and-fire (IaF) model is a simple neuron model that can still present spiking and bursting modes. It consists in a set of behaviours triggered by threshold-based rules.

The IaF model qualitatively reproduces the dynamics of the neuron around its resting potential V_0 . Its central equation $f(V) = -\alpha(V - V_0)$ is a linear solution to the general form ODE expressing the dependence of the change in membrane voltage on its current value, with possibly an external input term $I(t)$

$$\frac{dV}{dt} = f(V) + I(t) \quad (\text{A.18})$$

In absence of input $I(t)$, the equation is similar to the passive membrane equation. We can then plot $f(V)$ around V_0 and notice that V_0 is a stable attractor.

The non-linear spiking behaviour in presence of external input is introduced with ad-hoc rules manipulating the membrane voltage:

$$\begin{cases} \text{if } V < V_{th}, & C_m \frac{dV}{dt} = -g_L(V - E_i) - I_e \\ \text{if } V \geq V_{th}, & \text{fire a spike: } V \leftarrow V_{max} \\ \text{if } V = V_{max}, & \text{reset the membrane potential: } V \leftarrow V_{reset} \end{cases} \quad (\text{A.19})$$

Although mathematically less elegant than the Hodgkin-Huxley equations, the IaF model is able to qualitatively reproduce a lot of the firing behaviours of neurons.

A.2.2 A more elegant model: The exponential Integrate-and-Fire

In the to the expression $\frac{dV}{dt} = f(V) + I(t)$, f is now of second order quadratic, or the combination of a linear plus an exponential:

$$f(V) = -a(V - V_0) + e^{\frac{V - V_{th}}{\Delta}} \quad (A.20)$$

This expression makes $f(V)$ cross zero again, with a second fixed point on the right of V_0 , after V_{th} , and Δ determines the sharpness of the rise of $f(V)$ in the exponential part. This new fixed point is unstable.

Thanks to this, the exponential Integrate-and-Fire model does not need an ad-hoc rule for spike firing. Nevertheless, it still need one to reset of the voltage when it reaches V_{max} .

A.2.3 An most elegant model: the theta neuron

The theta neuron is a one-dimensional model on the unit circle:

$$\frac{d\theta}{dt} = i - \cos \theta + (1 + \cos \theta)I(t) \quad (A.21)$$

where the phase θ represents the voltage.

$\theta = \pi$ corresponds to a spike, and the reset from $\theta = 2\pi$ to $-\pi$ corresponds to the $[2\pi]$ congruence.

This model is equivalent to an IaF model with quadratic non-linearity. However, it's notable that the model fires regularly when $I(t) = 0$, so it's a useful model of periodically firing neurons.

A.2.4 An extra dimension

The problem of the IaF model is that it's one-dimensional, so there is no other solution to its inelegance than patching its dynamics or working in a Lie group. A second variable u is needed to elegantly take care of inactivation:

$$\begin{cases} \frac{dV}{dt} = F(V) + G(u) + I(t) \\ \frac{du}{dt} = -u + H(V) \end{cases} \quad (A.22)$$

The FitzHugh-Nagumo model [190] is a classical example of that family of two-dimensional models:

$$\begin{cases} \dot{V} = F(V) - u + I \\ \dot{u} = a(bV - cu) \end{cases} \quad (A.23)$$

where f is a third-degree polynomial, and a, b, c are real constants.

Figure A.7 plots the phase arrows and nullclines of the FitzHugh-Nagumo model with $a = 0.08$, $b = 0.7$ and $c = 0.8$, modelling the squid giant axon.

Figure A.7: Phase plane diagram of the two-dimensional FitzHugh-Nagumo neuron model with $a = 0.08$, $b = 0.7$ and $c = 0.8$

The phase plane diagram

On the (u, V) plane (Figure A.7), the points where $\frac{dV}{dt} = 0$ and those where $\frac{du}{dt} = 0$ define the *nullclines* of V and u . The fixed points are at the intersection of the u and V nullclines. At any point (V, u) in the phase plane, the trajectory is determined by $\frac{dV}{dt}(V, u)$ and $\frac{du}{dt}(V, u)$, as sampled by the phase arrows of the diagram.

This model is an elegant mathematical explanation of the illusion of threshold dynamics: in this model, like in recorded neurons, all-or-none responses do not truly exist. Instead, a *quasi-threshold* can be observed on the unstable (middle) branch of the V nullcline. This system also models other dynamical properties such as excitation block, anodal break excitation, spike accommodation [13]. The curated Scholarpedia page about the model [191] is an excellent resource to further learn about this and other differential models.

A.3 MORE REALISTIC MODELS

The dendritic tree affects information processing. *In vitro* current injection at different locations on a dendrite with recording in the soma show that distance attenuates and broadens the propagated depolarisation wave, transmission speed varies with dendrite properties including diameter, and the propagation of excitation is bi-directional,

but not symmetrically so. Mapping neuron morphology to electrodynamics is an ongoing research effort. See for instance [192].

Although out of the scope of this thesis, structural models based on cable theory complement the more abstract point models presented above, and constitute an important field of computational neuroscience. Chapter 6 of *"Theoretical Neuroscience: Computational and Mathematical Modelling of Neural Systems"* by Dayan & Abbott [193] is an introduction to compartmental models that I have found clear and rigorous.

NEURAL SIMULATION SOFTWARE

There is a number of approaches to the simulation of biological neural networks, and several notable software frameworks that permit the modelling using some of these approaches. Hence, I had to review the capabilities of most frameworks and assess if their modelling capabilities are suited to implement the EVAC model of bottom-up visual attention.

The following table summarises the available software that allows realistic network modelling.

Table B.1: Overview of some available software

Name	Description
(P)Genesis	Genesis, and its parallel version PGenesis are advanced simulation systems for compartmental models of neurons and networks of neurons in three dimensional space [194]. Genesis and other compartmental modelling software (e.g. NEURON) require a deep understanding of the microstructure of cortical neurons and networks.
NEURON	Similarly to Genesis, NEURON is a mature and advanced compartmental modelling software suite for modelling detailed neurons and networks with in three dimensions. [43]
XPP-Aut	This tool is a differential equations solver developed for compartmental modelling of neurons. There are interfaces to use XPP-Aut in Python and Matlab. [195]
Brian	Brian simulates networks of spiking neurons on multiple platforms. It appears actively maintained, and although it lets the user choose between integrate-and-fire and Hodgkin-Huxley models, the neurons are not easily modified through biological parameters. PyNN can be used. [46]

continued on next page

continued from previous page

Name	Description
Catacomb	Catacomb is a Java application for developing simulations of biologically plausible neural networks with an emphasis on animal behaviour. It was last updated in 2007. [196]
(P)CSIM	CSIM stands for neural microCircuit SIMulator. It eases the use of heterogeneous units in single networks: for instance, spiking and rate-coded neurons can be mixed in a single network. The parallel version PCSIM can be used for distributed simulation and has a Python user interface. Although the development of CSIM appears stagnant, PCSIM is actively developed. [45]
KInNeSS	The KDE Integrated NeuroSimulation Software is a GUI-driven simulation tool for networks of compartmental Hodgkin-Huxley neurons with few compartments. Ionic channels are all fully customizable. However, the development stopped in 2008. [197]
MVASpike	Still under active development at the INRIA, MVASpike is targeted at simulating integrate-and-fire spiking point neurons with Spike-timing-dependent plasticity. [198]
NCS	The NeoCortical Simulator is made to build large networks of compartmental Hodgkin-Huxley neurons. It appears actively maintained. [199]
PDP / Emergent	Emergent, successor of the PDP framework, is a graphical environment to create realistic networks of point neurons. The units possess parameters that let the various properties of ionic channels be manipulated. The scripting language used is not meant to be manipulated by humans, as programming the networks is performed through a graphical interface. Emergent is actively developed and used in the computational neuroscience community. [1]

continued on next page

continued from previous page

Name	Description
Nengo	Nengo (previously NESim, Neural Engineering Simulator) aims at developing large networks of point neurons. It is maintained but not associated with published models. [200]
(Py)NEST	NEST is a command-line/scripting simulator for large networks of realistic neurons. Units are point neurons but some ionic channels parameters are adjustable. Although the programming language used to describe and run networks is a simple custom stack language using the reverse Polish notation, the PyNEST extension of the simulator lets the user use the much friendlier Python scripting language. [44, 201]
NSL	The Neural Simulation Language is a framework for developing a wide range of abstract or more realistic neural networks. Implementations are available in C++, Matlab and Java, but the last release is from 2003. [202]
SpikeNet	This software aims at running very large networks of spiking neurons, but does not appear actively developed. [203]
Topographica	Topographica is specialized in realistic topographical maps of the primary visual cortex. It implements all the latest knowledge about the formation, structure and connectivity of the V ₁ area, and lets the user build upon that. It is maintained. [135]
XNBC	This tool is a mature framework for the simulations of biological neural networks. It leaves the choice between a leaky integrator model of neurons or a Hodgkin-Huxley based. [204]

Of these possibilities, a good tool for my purpose would strike a balance between the granularity of biological details, with NEURON and Genesis allowing very detailed but complicated and slow simulations with few detailed neurons, and faster and simpler tools permitting to build more consequent networks, but that may however lack biological realism or parameters to simulate hypothesized defects of synapses, channels, units or networks.

With regard to the maturity of the packages listed in Table B.1, Topographica, Genesis, PDP++, Emergent, and NEST are the most promising. Genesis builds explicit three-dimensional models of neurons, and using it requires consequent knowledge in Neurology. I was unable to confidently manipulate it. Topographica is a newer program, and makes use of point neurons with few parameters, but, being targeted at the development of realistic topographical maps of the visual cortex, uses a more advanced and biologically justified type of self-organised feature map (SOM), with excellent results regarding the formation of realistic feature maps for early visual areas [135, 182]. PDP, Emergent and NEST use point neurons but expose more parameters to let one modify the static and dynamic electrical properties of neurons. Emergent is the successor of the PDP package, and as such has a consequent literature background. All design decisions, simplifications and algorithms used are made in light of actual *in vivo* behaviour and justified in the reference book [1]. Although written for PDP, the predecessor of Emergent, the book presents insights into emergent visual attention under the form of simulations that are already or easily implemented with Emergent. In addition, the rate-coded Hodgkin-Huxley neuron model of Emergent contains all the necessary parameters to simulate the modelled effects of channelopathy. Thus, I chose to build upon those foundations by using Emergent to implement the model.

CONCLUSION OF THE THESIS

This dissertation investigated some aspects of the brain-to-behaviour link. The first part placed the field in the current scientific and societal contexts, and introduced the remainder of the thesis. The second part explored a novel neural network model of the influence of calcium channelopathy on bottom-up visual attention (EVAC), where defects to L-type voltage-gated calcium channel (LTCC) are hypothesised to have a role in certain cases of autism. The third part presented a new software tool (PyCogMo) developed to address the lack of unified tools for weights learning in models of biological neural networks. The PyCogMo framework was created to improve specific issues with reproducibility in computational neuroscience, from which the current implementation of EVAC with the Emergent neural simulator also suffers.

The EVAC model is a recurrent neural network that implements the dual-pathway model of vision. In this model, information from the retinas is first transformed by a common processing pipeline of sub-cortical and cortical maps, and later processed by two separate streams in the cortex. Thus, the synaptic weights of the artificial neural network simulation must be trained to learn the correct input-output associations. In EVAC, the majority of the learning is done using the Leabra learning algorithm [1]. However, the Leabra algorithm is only implemented in the Emergent simulator, restricting the reproducibility of EVAC. PyCogMo offers the environment necessary to implement the Leabra algorithm and to run it on several independent simulators. Besides the trained dual-pathway network, EVAC also embodies the emergent theory of bottom-up visual attention, which was first modelled computationally by O'Reilly & Munakata [35]. In this theory, a shift of attention to a new salient stimulus is not the result of the activation of a unique saliency map, but an emergent property of the network that results from the *accommodation* property of individual units. Neuronal accommodation describes the lower response of neurons to input after sustained activity. This phenomenon is mediated by several processes, but mostly depends on the short-term history of activity of the neuron. Hence, it is theoretically affected by channelopathy. Thus, it is expected that channelopathy affects reflex visual attention. The model of EVAC is instantiated with two different parameter sets, resulting in: (1) a control network designed to model attention shift neurotypical subjects, and (2) a channelopathic network that aims at modelling the attention shift in subjects suffering from calcium channelopathy. The trained networks are tested on attention shift tasks inspired by the works of Posner on attention [2].

The change of activations of all layers partly mediated by accommodation is observed to result in attention shift with significantly different timings depending on the presence of channelopathy. The error rates also vary, but to a lesser extent. The simplifications made in designing the network model do not allow for a direct translation of simulated time units to real time. However, the relative time differences between the channelopathic network and the control network are expected to be qualitatively comparable to the clinical trials on Posner-type tasks. Hence, the results of the simulation of the control and channelopathic EVAC networks on three attention shift tasks results in several testable predictions about the relative attention shift performance of subjects suffering from channelopathy compared to controls. As it is plausible that channelopathy explains certain cases of autism, these simulation results are compared to existing studies of reflex attention in autistic children, showing concordance. Further experimental studies are proposed to explore bottom-up attention in calcium channelopathy and to better assess the plausibility of the EVAC model. Implementing the model using PyCogMo and PyNN should also be considered in view of improving the reliability of the simulation of the EVAC network. PyCogMo provides not only software abstractions to allow learning the network weights by any of several simulators, but also contains a visualisation module to examine network structure and activity. Interpretability is an important factor in models with a highly dimensional state space like artificial neural networks. The visualisation module of PyCogMo would ease the interpretation of the network dynamics of EVAC, thereby potentially improving the interpretability of the results.

BIBLIOGRAPHY

- [1] O'Reilly RC & Munakata Y. *Computational Explorations in Cognitive Neuroscience*. The MIT Press. 2000.
- [2] Posner MI. *Orienting of attention*. Q J Exp Psychol, 32(1), 3–25. 1980.
- [3] O'Reilly R. *The Leabra model of neural interactions and learning in the neocortex*. Ph.D. thesis, Carnegie Mellon University. 1996.
- [4] Davison AP, Brüderle D, Eppler J et al. *PyNN: a common interface for neuronal network simulators*. Frontiers in Neuroinformatics, 2. 2008.
- [5] Dennett DC & Kinsbourne M. *Time and the Observer: The Where and When of Consciousness in the Brain*. Behavioral and Brain Sciences, 15, 183–201. 1992.
- [6] Anderson JR, Matessa M & Lebiere C. *ACT-R: A Theory of Higher Level Cognition and Its Relation to Visual Attention*. Hum.-Comput. Interact., 12(4), 439–462. 1997.
- [7] Miikkulainen R, Bednar J, Choe Y et al. *Computational Maps in the Visual Cortex*. Springer. 2005.
- [8] Group PR. *Parallel distributed processing: explorations in the microstructure of cognition, vol. 1: foundations*. The MIT Press, Cambridge, MA, USA. 1986.
- [9] Laird JE. *The Soar Cognitive Architecture*. The MIT Press. 2012.
- [10] Kohonen T. *Self-organized formation of topologically correct feature maps*. Biological cybernetics, 43(1), 59–69. 1982.
- [11] Hooke R. *Micrographia: or, Some physiological descriptions of minute bodies made by magnifying glasses. With observations and inquiries thereupon*. The Royal Society of London. 1665.
- [12] Scholl BJ & Leslie AM. *Modularity, development and 'theory of mind'*. Mind & Language, 14(1), 131–153. 1999.
- [13] Izhikevich EM. *Dynamical systems in neuroscience*. The MIT Press. 2007.
- [14] Fodstad H. *The Neuron Theory*. Stereotactic and Functional Neurosurgery, 77(1-4), 20–24. 2001.
- [15] Sotelo C. *Viewing the brain through the master hand of Ramon y Cajal*. Nature Reviews Neuroscience, 4(1), 71–77. 2003.

- [16] Tejada-Vera B. *Mortality from Alzheimer's Disease in the United States: Data for 2000 and 2010*. US Department of Health and Human Services, Centers for Disease Control and Prevention, National Center for Health Statistics. 2013.
- [17] Kessler RC, Chiu WT, Demler O et al. *Prevalence, severity, and comorbidity of 12-month DSM-IV disorders in the National Comorbidity Survey Replication*. Arch Gen Psychiatry, 62(6), 617–627. 2005.
- [18] Bruns J Jr & Hauser WA. *The epidemiology of traumatic brain injury: a review*. Epilepsia, 44 Suppl 10, 2–10. 2003.
- [19] Baguley IJ, Nott MT, Howle AA et al. *Late mortality after severe traumatic brain injury in New South Wales: a multicentre study*. The Medical Journal of Australia, 196(1), 40–45. 2011.
- [20] Blumberg SJ, Bramlett MD, Kogan MD et al. *Changes in prevalence of parent-reported autism spectrum disorder in school-aged US children: 2007 to 2011–2012*. National Health Statistics Reports, 65, 1–11. 2013.
- [21] Yudkowsky E. *Levels of Organization in General Intelligence*. Springer-Verlag. 2007.
- [22] Muehlhauser L & Salamon A. *Intelligence explosion: Evidence and import*. In: *Singularity Hypotheses*, 15–42. Springer. 2012.
- [23] McCulloch W & Pitts W. *A logical calculus of the ideas immanent in nervous activity*. Bulletin of Mathematical Biophysics, 5, 115–133. 1943.
- [24] Anderson JR. *The Architecture of Cognition*. Harvard University Press, Cambridge, MA, USA. 1983.
- [25] Anderson JR & Bower GH. *A propositional theory of recognition memory*. Memory & Cognition, 2(3), 406–412. 1974.
- [26] McClelland JL & Rumelhart DE (Editors). *Parallel Distributed Processing, Explorations in the Microstructure of Cognition*, vol. 1. The MIT Press, Cambridge, MA. 1986.
- [27] Medler D. *A brief history of connectionism*. Neural Computing Surveys, 1, 18–72. 1998.
- [28] Sun R. *Introduction to computational cognitive modeling*. Cambridge handbook of computational psychology, 3–19. 2008.
- [29] Crick F. *The recent excitement about neural networks*. Nature, 337(6203), 129–132. 1989.

- [30] Hintzman D. *Human learning and memory: Connections and dissociations*. Annual Review of Psychology, 41(1), 109–139. 1990.
- [31] Lebiere C & Anderson J. *A connectionist implementation of the ACT-R production system*. In: *Proceedings of the fifteenth annual conference of the Cognitive Science Society*, 635–640. Hillsdale NJ: Erlbaum. 1993.
- [32] Sun R. *Integrating rules and connectionism for robust reasoning*. In: *Proc. AAAI Workshop on Integrating Neural and Symbolic Processes*. Boston, MA. Citeseer. 1990.
- [33] Dyan P. *Levels of Analysis in Neural Modeling*, 2102–2107. Nature Publishing Group, 1 ed. 2003.
- [34] Sun R, Coward LA & Zenzen MJ. *On levels of cognitive modeling*. Philosophical Psychology, 18, 613–637. 2005.
- [35] O'Reilly R & Munakata Y. *Perception and Attention*, chap. 8, 227–273. The MIT Press, 1 ed. 2000.
- [36] Piedras-Rentería ES, Barrett CF, Cao YQ et al. *Voltage-gated calcium channels, calcium signaling, and channelopathies*. In: Krebs J & Michalak M (Editors), *Calcium - A Matter of Life or Death*, vol. Volume 41, 127–166. Elsevier. 2007.
- [37] Courchesne E, Townsend J, Akshoomoff NA et al. *Impairment in shifting attention in autistic and cerebellar patients*. Behavioral Neuroscience, 108(5), 848–865. 1994.
- [38] Townsend J, Harris NS & Courchesne E. *Visual attention abnormalities in autism: Delayed orienting to location*. Journal of the International Neuropsychological Society, 2(6), 541–550. 1996.
- [39] van der Geest JN, Kemner C, Camfferman G et al. *Eye movements, visual attention, and autism: a saccadic reaction time study using the gap and overlap paradigm*. Biological Psychiatry, 50(8), 614 – 619. 2001.
- [40] Ungerleider LG & Haxby JV. *'What' and 'where' in the human brain*. Curr Opin Neurobiol, 4(2), 157–165. 1994.
- [41] "The Coq Development Team". *The Coq Proof Assistant Reference Manual*. INRIA. 2004.
- [42] Rice HG. *Classes of recursively enumerable sets and their decision problems*. Transactions of the American Mathematical Society, 74(2), 358–358. 1953.
- [43] Carnevale N & Hines M. *The NEURON book*. Cambridge Univ Pr. 2006.

- [44] Gewaltig MO & Diesmann M. *NEST (NEural Simulation Tool)*. Scholarpedia, 2(4), 1430. 2007.
- [45] Pecevski D, Natschläger T & Schuch K. *PCSIM: a parallel simulation environment for neural circuits fully integrated with Python*. Frontiers in neuroinformatics, 3. 2009.
- [46] Goodman DFM & Brette R. *The Brian simulator*. 2010.
- [47] Schroeder WJ, Martin KM & Lorensen WE. *The design and implementation of an object-oriented toolkit for 3D graphics and visualization*. In: *Proceedings of the 7th conference on Visualization'96*, 93–ff. IEEE Computer Society Press. 1996.
- [48] Anderson JR. *Cognitive Psychology and its Implications, Sixth Edition*. Worth Publishers. 2004.
- [49] Penn HE. *Neurobiological Correlates of Autism: a Review of Recent Research*. Child Neuropsychology, 12, 57–79. 2006.
- [50] Huber DE & O'Reilly RC. *Persistence and accommodation in short-term priming and other perceptual paradigms: temporal segregation through synaptic depression*. Cognitive Science, 27(3), 403–430. 2003.
- [51] Goodale MA & Milner A. *Separate visual pathways for perception and action*. Trends in Neurosciences, 15(1), 20–25. 1992.
- [52] Ames CS & Jarrold C. *Identifying Symbolic Relationships in Autism Spectrum Disorders: A Deficit in the Identification of Temporal Co-occurrence?* Journal of Autism and Developmental Disorders, 1–12. 2009.
- [53] Levy SE, Mandell DS & Schultz RT. *Autism*. The Lancet, 374(9701), 1627–1638. 2009.
- [54] Ashwood P, Wills S & Van de Water J. *The immune response in autism: a new frontier for autism research*. Journal of Leukocyte Biology, 80(1), 1–15. 2006.
- [55] Erickson C, Stigler K, Corkins M et al. *Gastrointestinal Factors in Autistic Disorder: A Critical Review*. Journal of Autism and Developmental Disorders, 35(6), 713–727. 2005.
- [56] Landrigan PJ, Lambertini L & Birnbaum LS. *A research strategy to discover the environmental causes of autism and neurodevelopmental disabilities*. Environ Health Perspect, 120(7), a258–a260. 2012.
- [57] Sparks BF, Friedman SD, Shaw DW et al. *Brain structural abnormalities in young children with autism spectrum disorder*. Neurology, 59, 184–192. 2002.

- [58] Courchesne E, Karns CM, Davis HR et al. *Unusual brain growth patterns in early life in patients with autistic disorder*. *Neurology*, 245–254. 2001.
- [59] Akshoomoff N, Pierce K & Courchesne E. *The neurobiological basis of autism from a developmental perspective*. *Development and Psychopathology*, 14, 613–634. 2002.
- [60] Nicolson R & Szatmari P. *Genetic and neurodevelopmental influences in autistic disorder*. *Canadian Journal of Psychiatry*, 48, 526–537. 2003.
- [61] Bailey A, Luthert P, Dean A et al. *A clinicopathological study of autism*. *Brain*, 121(5), 889–905. 1998.
- [62] Casanova MF, Buxhoeveden DP, Switala AE et al. *Minicolumnar pathology in autism*. *Neurology*, 58, 428–432. 2002.
- [63] Markram H. *The Blue Brain Project*. *Nature Reviews Neuroscience*, 7, 153–160. 2006.
- [64] Persico AM, d'Agruma L, Maiorano N et al. *Reelin gene alleles and haplotypes as a factor predisposing to autistic disorder*. *Molecular psychiatry*, 6(2), 150–159. 2001.
- [65] Skaar DA, Shao Y, Haines JL et al. *Analysis of the RELN gene as a genetic risk factor for autism*. *Molecular Psychiatry*, 10(6), 563–571. 2004.
- [66] Serajee FJ, Zhonga H & Huq AM. *Association of Reelin gene polymorphisms with autism*. *Genomics*, 87(1), 75–83. 2006.
- [67] Fatemi SH, Stary JM, Halt AR et al. *Dysregulation of Reelin and Bcl-2 Proteins in Autistic Cerebellum*. *Journal of Autism and Developmental Disorders*, 31(6), 529–535. 2001.
- [68] Fatemi SH & Halt AR. *Altered levels of Bcl2 and p53 proteins in parietal cortex reflect deranged apoptotic regulation in autism*. *Synapse*, 42(4), 281–284. 2001.
- [69] Fatemi SH, Merz A & Realmuto GR. *The Roles of Reelin, Bcl2, and Serotonin in Cerebellar Pathology in Autism*. *Journal of Developmental and Physical Disabilities*, 15(1), 1–22. 2003.
- [70] Nelson KB, Grether JK, Croen LA et al. *Neuropeptides and neurotrophins in neonatal blood of children with autism or mental retardation*. *Ann Neurol*, 49(5), 597–606. 2001.
- [71] Bauman M & Kemper TL. *Histoanatomic observations of the brain in early infantile autism*. *Neurology*, 35, 866–874. 1985.

- [72] Kemper TL & Bauman ML. *Neuropathology of infantile autism*. Journal of neuropathology and experimental neurology, 57(7), 645–652. 1998.
- [73] Courchesne E, Saitoh O, Yeung-Courchesne R et al. *Abnormality of cerebellar vermal lobules VI and VII in patients with infantile autism: identification of hypoplastic and hyperplastic subgroups with MR imaging*. American journal of roentgenology, 162(1), 123–130. 1994.
- [74] Lyons V & Fitzgerald M. *Asperger (1906–1980) and Kanner (1894–1981), the two pioneers of autism*. Journal of Autism and Developmental Disorders, 37(10), 2022–2023. 2007.
- [75] Happé F & Frith U. *The Weak Coherence Account: Detail-focused Cognitive Style in Autism Spectrum Disorders*. Journal of Autism and Developmental Disorders, 36(1), 5–25. 2006.
- [76] Hill EL. *Executive dysfunction in autism*. Trends in Cognitive Sciences, 8(1), 26 – 32. 2004.
- [77] Baron-Cohen S. *Autism: The Empathizing-Systemizing (E-S) Theory*. Annals of the New York Academy of Sciences, 1156(The Year in Cognitive Neuroscience 2009), 68–80. 2009.
- [78] Frith U. *Autism: Explaining the enigma*. Wiley-Blackwell, 2 ed. 2003.
- [79] Hallmayer J. *Genetic Heritability and Shared Environmental Factors Among Twin Pairs With Autism*. Archives of General Psychiatry, 68(11), 1095. 2011.
- [80] Lu ATH, Dai X, Martinez-Agosto JA et al. *Support for calcium channel gene defects in autism spectrum disorders*. Molecular Autism, 3(1), 18. 2012.
- [81] Vargas DL, Nascimbene C, Krishnan C et al. *Neuroglial activation and neuroinflammation in the brain of patients with autism*. Annals of Neurology, 57(1), 67–81. 2005.
- [82] Poggi A, Rubartelli A & Zocchi MR. *Involvement of dihydropyridine-sensitive calcium channels in human dendritic cell function. Competition by HIV-1 Tat*. J Biol Chem, 273(13), 7205–7209. 1998.
- [83] Zocchi MR, Rubartelli A, Morgavi P et al. *HIV-1 Tat inhibits human natural killer cell function by blocking L-type calcium channels*. J Immunol, 161(6), 2938–2943. 1998.
- [84] Cheshenko N, Del Rosario B, Woda C et al. *Herpes simplex virus triggers activation of calcium-signaling pathways*. J Cell Biol, 163(2), 283–293. 2003.

- [85] Misra UK, Gawdi G, Pizzo SV et al. *Exposure of cultured murine peritoneal macrophages to low concentrations of beryllium induces increases in intracellular calcium concentrations and stimulates DNA synthesis.* J Leukoc Biol, 65(6), 786–791. 1999.
- [86] Palmieri L, Papaleo V, Porcelli V et al. *Altered calcium homeostasis in autism-spectrum disorders: evidence from biochemical and genetic studies of the mitochondrial aspartate/glutamate carrier AGC1.* Molecular Psychiatry, –. 2008.
- [87] Napolioni V, Persico AM, Porcelli V et al. *The Mitochondrial Aspartate/Glutamate Carrier AGC1 and Calcium Homeostasis: Physiological Links and Abnormalities in Autism.* Molecular Neurobiology, 44(1), 83–92. 2011.
- [88] McEnery MW, Vance CL, Begg CM et al. *Differential expression and association of calcium channel subunits in development and disease.* J Bioenerg Biomembr, 30(4), 409–418. 1998.
- [89] Gargus JJ. *Genetic Calcium Signaling Abnormalities in the Central Nervous System: Seizures, Migraine, and Autism.* Annals of the New York Academy of Sciences, 1151(1), 133–156. 2009.
- [90] Dolphin AC. *A short history of voltage-gated calcium channels.* Br J Pharmacol, 147 Suppl 1, S56–S62. 2006.
- [91] Splawski I, Timothy KW, Sharpe LM et al. *CaV1.2 Calcium Channel Dysfunction Causes a Multisystem Disorder Including Arrhythmia and Autism.* Cell, 119(1), 19 – 31. 2004.
- [92] Lozac NBS. *Central Role of Voltage Gated Calcium Channels and Intercellular Calcium Homeostasis in Autism.* 2009. First version published on-line on www.autismcalciumchannelopathy.com.
- [93] Krey JF & Dolmetsch RE. *Molecular mechanisms of autism: a possible role for Ca²⁺ signaling.* Current Opinion in Neurobiology, 17(1), 112 – 119. 2007. Development.
- [94] Turiel A & Parga N. *Role of statistical symmetries in sensory coding: an optimal scale invariant code for vision.* J Physiol Paris, 97(4-6), 491–502. 2003.
- [95] Elsabbagh M, Volein A, Holmboe K et al. *Visual orienting in the early broader autism phenotype: disengagement and facilitation.* Journal of Child Psychology and Psychiatry, 50(5), 637–642. 2009.
- [96] Townsend J, Westerfield M, Leaver E et al. *Event-related brain response abnormalities in autism: evidence for impaired cerebello-frontal spatial attention networks.* Cognitive Brain Research, 11(1), 127–45. 2001.

- [97] Zwaigenbaum L, Bryson S, Rogers T et al. *Behavioral manifestations of autism in the first year of life*. International Journal of Developmental Neuroscience, 23(2-3), 143 – 152. 2005. Autism: Modeling Human Brain Abnormalities in Developing Animal Systems.
- [98] Schenk T & McIntosh RD. *Do we have independent visual streams for perception and action?* Cognitive Neuroscience, 1(1), 52–62. 2010.
- [99] Wu W. *What is Conscious Attention?* Philosophy and Phenomenological Research, 82(1), 93–120. 2011.
- [100] Posner M, Walker J, Friedrich F et al. *Effects of parietal injury on covert orienting of attention*. Journal of Neuroscience, 4(7), 1863–1874. 1984.
- [101] Knudsen EI. *Fundamental Components of Attention*. Annual Review of Neuroscience, 30(1), 57–78. 2007.
- [102] Korsten NJH, Fragopanagos N, Hartley M et al. *Attention as a controller*. Neural Networks, 19(9), 1408–21. 2006.
- [103] Taylor J. *CODAM: A neural network model of consciousness*. Neural Networks, 20(9), 983–992. 2007.
- [104] Taylor JG. *The CODAM Model and Deficits of Consciousness I: CODAM*. Springer-Verlag. 2003.
- [105] Taylor JG. *CODAM and Deficits of Consciousness II: Schizophrenia and Neglect/Extinction*. Springer-Verlag. 2003.
- [106] Tsotsos JK & Rothenstein A. *Computational models of visual attention*. Scholarpedia, 6(1), 6201. 2011.
- [107] Koch C & Ullman S. *Shifts in selective visual attention: towards the underlying neural circuitry*. Hum Neurobiol, 4(4), 219–227. 1985.
- [108] Desimone R & Duncan J. *Neural mechanisms of selective visual attention*. Annual review of neuroscience, 18(1), 193–222. 1995.
- [109] Itti L & Koch C. *Computational modelling of visual attention*. Nature Reviews Neuroscience, 2(3), 194–203. 2001.
- [110] Kirchner H, Barbeau E, Thorpe S et al. *Ultra-rapid sensory responses in the human frontal eye field region*. Journal of Neuroscience, 29(23), 7599. 2009.
- [111] Hamker FH & Zirnsak M. *V4 receptive field dynamics as predicted by a systems-level model of visual attention using feedback from the frontal eye field*. Neural Networks, 19(9), 1371–82. 2006.

- [112] Hommel B. *On the social psychology of modelling*. Neural Networks, 19(9), 1455–7. 2006.
- [113] Tsotsos JK, Culhane SM, Wai WYK et al. *Modeling visual attention via selective tuning*. Artificial Intelligence, 78(1–2), 507 – 545. 1995. Special Volume on Computer Vision.
- [114] Kreiter AK. *How do we model attention-dependent signal routing?* Neural Networks, 19(9), 1443–4. 2006.
- [115] Niebur E, Koch C & Rosin C. *An oscillation-based model for the neuronal basis of attention*. Vision Research, 33(18), 2789–2802. 1993.
- [116] Niebur E. *Temporal tagging of attended objects*. Proceedings of the National Academy of Sciences, 106(8), 2479. 2009.
- [117] Crick F & Koch C. *Towards a neurobiological theory of consciousness*. Seminars in the Neurosciences, 2(263–275), 203. 1990.
- [118] Borisyuk RM & Kazanovich YB. *Oscillatory neural network model of attention focus formation and control*. Biosystems, 71(1–2), 29–38. 2003.
- [119] Borisyuk RM & Kazanovich YB. *Oscillatory model of attention-guided object selection and novelty detection*. Neural Networks, 17(7), 899–915. 2004.
- [120] Borisyuk R, Kazanovich YB, Chik D et al. *A neural model of selective attention and object segmentation in the visual scene: An approach based on partial synchronization and star-like architecture of connections*. Neural Networks, 22(5–6), 707–719. 2009.
- [121] Mishra J, Fellous J & Sejnowski T. *Selective attention through phase relationship of excitatory and inhibitory input synchrony in a model cortical neuron*. Neural Networks, 19(9), 1329–1346. 2006.
- [122] Crick F & Koch C. *Some reflections on visual awareness*. Cold Spring Harb Symp Quant Biol, 55, 953–962. 1990.
- [123] Shadlen MN & Movshon JA. *Synchrony unbound: a critical evaluation of the temporal binding hypothesis*. Neuron, 24(1), 67–77, 111–25. 1999.
- [124] Reynolds JH, Chelazzi L & Desimone R. *Competitive mechanisms subserve attention in macaque areas V2 and V4*. J Neurosci, 19(5), 1736–1753. 1999.
- [125] Chelazzi L, Miller EK, Duncan J et al. *A neural basis for visual search in inferior temporal cortex*. Nature, 363(6427), 345–347. 1993.

- [126] Corchs S & Deco G. *A neurodynamical model to simulate neural activities in visual attention experiments*. *Neurocomputing*, 44, 759–768. 2002.
- [127] Deco G & Zihl J. *A neurodynamical model of visual attention: Feedback enhancement of spatial resolution in a hierarchical system*. *Journal of Computational Neuroscience*, 10(3), 231–253. 2001.
- [128] Deco G, Pollatos O & Zihl J. *The time course of selective visual attention: theory and experiments*. *Vision Research*, 42(27), 2925–2945. 2002.
- [129] Deco G & Rolls E. *A neurodynamical cortical model of visual attention and invariant object recognition*. *Vision research*, 44(6), 621–642. 2004.
- [130] Rolls ET & Deco G. *Attention in natural scenes: Neurophysiological and computational bases*. *Neural Networks*, 19(9), 1383–94. 2006.
- [131] Posner MI & Cohen Y. *Components of visual orienting*. *Attention and performance X: Control of language processes*, 32, 531–556. 1984.
- [132] Olshausen B, Anderson C & Van Essen D. *A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information*. *J. Neurosci.*, 13(11), 4700–4719. 1993.
- [133] McClelland JL. *The basis of hyperspecificity in autism: A preliminary suggestion based on properties of neural nets*. *Journal of Autism and Developmental Disorders*, 30(5), 497–502. 2000.
- [134] Cohen IL. *An artificial neural network analogue of learning in autism*. *Biological Psychiatry*, 36(1), 5–20. 1994.
- [135] Bednar J, Kelkar A & Miikkulainen R. *Scaling self-organizing maps to model large cortical networks*. *Neuroinformatics*, 2(3), 275–301. 2004.
- [136] Gustafsson L & Papliński A. *Self-organization of an artificial neural network subjected to attention shift impairments and familiarity preference, characteristics studied in autism*. *Journal of Autism and Developmental Disorders*, 34(2), 189–198. 2004.
- [137] Courchesne E, Townsend JP & Akshoomoff NA. *A new finding: Impairment in shifting attention in autistic and cerebellar patients*. In: *Atypical Cognitive Deficits in Developmental Disorders: Implications for Brain Function*, 101–137. Lawrence Erlbaum. 1994.
- [138] Paplinski AP & Gustafsson L. *An attempt in modelling early intervention in autism using neural networks*. In: *Neural Networks*, 2004.

- Proceedings. 2004 IEEE International Joint Conference on*, vol. 1. IEEE. 2004.
- [139] Noriega G. *Self-organizing maps as a model of brain mechanisms potentially linked to autism*. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 15(2), 217–226. 2007.
  - [140] Noriega G. *A neural model for compensation of sensory abnormalities in autism through feedback from a measure of global perception*. IEEE Transactions on Neural Networks, 19(8), 1402–1414. 2008.
  - [141] Noriega G. *Modeling propagation delays in the development of SOMs - a parallel with abnormal brain growth in autism*. Neural Networks, 21(2-3), 130–139. 2008.
  - [142] Rossman I & DiCicco-Bloom E. *ENGRAILED2 and Cerebellar Development in the Pathogenesis of Autism Spectrum Disorders*. In: *Autism: Current Theories and Evidence*, chap. 1, 3–40. Springer. 2008.
  - [143] Dawson M, Mottron L & Gernsbacher M. *Learning in autism*. JH Byrne (Series Ed.) & HL Roediger (Vol. Ed.), *Learning and memory: A comprehensive reference*, 759–772. 2008.
  - [144] Oliver A, Johnson MH, Karmiloff-Smith A et al. *Deviations in the emergence of representations: A neuroconstructivist framework for analysing developmental disorders*. Developmental Science, 3(1), 1–40. 2000.
  - [145] Price C & Friston K. *Functional ontologies for cognition: The systematic definition of structure and function*. The Multiple Functions of Sensory-Motor Representations, 22(3/4), 262. 2005.
  - [146] Neumann D, Spezio M, Piven J et al. *Looking you in the mouth: abnormal gaze in autism resulting from impaired top-down modulation of visual attention*. Social cognitive and affective neuroscience, 1(3), 194. 2006.
  - [147] Brock J, Brown CC, Boucher J et al. *The temporal binding deficit hypothesis of autism*. Development and Psychopathology, null, 209–224. 2002.
  - [148] Heimburg T & Jackson A. *On soliton propagation in biomembranes and nerves*. Proceedings of the National Academy of Sciences of the United States of America, 102(28), 9790. 2005.
  - [149] Das P & Schwarz WH. *Solitons in cell membranes*. Phys. Rev. E, 51(4), 3588–3612. 1995.
  - [150] Heimburg T & Jackson AD. *The Thermodynamics of General Anesthesia*. Biophysical Journal, 92(9), 3159–3165. 2007.

- [151] Heilman AD & Quattrochi J. *Computational models of epileptiform activity in single neurons*. Biosystems, 78(1-3), 1 – 21. 2004.
- [152] Leach A. *Molecular Modelling: Principles and Applications*. Prentice Hall, Englewood Cliffs. 2001.
- [153] Itti L, Koch C & Niebur E. *A model of saliency-based visual attention for rapid scene analysis*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 20(11), 1254–1259. 1998.
- [154] O'Reilly R. *Biologically plausible error-driven learning using local activation differences: The generalized recirculation algorithm*. Neural computation, 8(5), 895–938. 1996.
- [155] Goldberg MC, Landa R, Lasker A et al. *Evidence of Normal Cerebellar Control of the Vestibulo-Ocular Reflex (VOR) in Children with High-Functioning Autism*. Journal of Autism and Developmental Disorders, 30(6), 519–524. 2000.
- [156] McClelland J & Rumelhart D. *Explorations in Parallel Distributed Processing Handbook*. The MIT Press. 1987.
- [157] Sporns O, Honey CJ & Kötter R. *Identification and Classification of Hubs in Brain Networks*. PLoS ONE, 2(10), e1049. 2007.
- [158] Kandel E, Schwartz J, Jessell T et al. *Principles of neural science*. Elsevier New York, 4 ed. 1991.
- [159] Oja E. *Simplified neuron model as a principal component analyzer*. Journal of Mathematical Biology, 15(3), 267–273. 1982.
- [160] Rumelhart DE, Hinton GE & Williams RJ. *Learning representations by back-propagating errors*. Nature, 323(6088), 533–536. 1986.
- [161] Hinton G & McClelland J. *Learning representations by recirculation*. Neural information processing systems: Denver, CO, 1987, 358. 1988.
- [162] Limpert E, Stahel WA & Abbt M. *Log-normal Distributions across the Sciences: Keys and Clues*. BioScience, 51(5), 341. 2001.
- [163] Baayen RH & Milin P. *Analyzing Reaction Times*. International Journal of Psychological Research, 3(2). 2010.
- [164] Saslow MG. *Effects of Components of Displacement-Step Stimuli Upon Latency for Saccadic Eye Movement*. J. Opt. Soc. Am., 57(8), 1024–1029. 1967.
- [165] Adams P & Snutch T. *Calcium Channelopathies: Voltage-Gated Calcium Channels*. Springer-Verlag. 2007.

- [166] Sharma A, Thakker RV & Jüppner H. *Genetic Disorders of Calcium and Phosphate Homeostasis*. In: *Pediatric Nephrology*, vol. 2, chap. 11, 267–305. Springer Berlin Heidelberg. 2009.
- [167] Simpson HD, Mortimer D & Goodhill GJ. *Theoretical models of neural circuit development*. *Current topics in developmental biology*, 87, 1–51. 2009.
- [168] Casanova MF, Kooten IAJ, Switala AE et al. *Minicolumnar abnormalities in autism*. *Acta Neuropathologica*, 112(3), 287–303. 2006.
- [169] Vuilleumier P & Pourtois G. *Distributed and interactive brain mechanisms during emotion face perception: Evidence from functional neuroimaging*. *Neuropsychologia*, 45(1), 174–194. 2007.
- [170] Hildenbrand T, Rothlauf F, Geisser M et al. *Approaches to collaborative software development*. In: *Complex, Intelligent and Software Intensive Systems*, 2008. *CISIS 2008. International Conference on*, 523–528. IEEE. 2008.
- [171] Abrahamsson P, Salo O, Ronkainen J et al. *Agile software development methods: Review and analysis*. 2002.
- [172] Gleeson P, Crook S, Cannon RC et al. *NeuroML: A Language for Describing Data Driven Models of Neurons and Networks with a High Degree of Biological Detail*. *PLoS Computational Biology*, 6(6), e1000815. 2010.
- [173] Raikov I, Cannon R, Clewley R et al. *NineML: the network interchange for neuroscience modeling language*. *BMC Neuroscience*, 12(Suppl 1), P330. 2011.
- [174] Hucka M, Finney A, Sauro HM et al. *The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models*. *Bioinformatics*, 19(4), 524–531. 2003.
- [175] Demir E, Cary MP, Paley S et al. *The BioPAX community standard for pathway data sharing*. *Nature Biotechnology*, 28(9), 935–942. 2010.
- [176] Djurfeldt M, Hjorth J, Eppler JM et al. *Run-Time Interoperability Between Neuronal Network Simulators Based on the MUSIC Framework*. *Neuroinformatics*, 8(1), 43–60. 2010.
- [177] Sanner MF et al. *Python: a programming language for software integration and development*. *J Mol Graph Model*, 17(1), 57–61. 1999.
- [178] Markram H, Lübke J, Frotscher M et al. *Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs*. *Science*, 275(5297), 213–215. 1997.

- [179] Morrison A, Diesmann M & Gerstner W. *Phenomenological models of synaptic plasticity based on spike timing*. Biol Cybern, 98(6), 459–478. 2008.
- [180] Reymann KG & Frey JU. *The late maintenance of hippocampal LTP: requirements, phases, 'synaptic tagging', 'late-associativity' and implications*. Neuropharmacology, 52(1), 24–40. 2007.
- [181] Clopath C. *Synaptic consolidation: an approach to long-term learning*. Cogn Neurodyn, 6(3), 251–257. 2012.
- [182] Bednar JA & Mäkelä R. *Joint maps for orientation, eye, and direction preference in a self-organizing model of V1*. Neurocomputing, 69(10–12), 1272–1276. 2006.
- [183] Hebb DO. *The organization of behavior: A neuropsychological approach*. John Wiley & Sons. 1949.
- [184] VanRossum G & Drake FL. *The Python Language Reference*. Python Software Foundation. 2010.
- [185] Oliphant TE. *Python for Scientific Computing*. Computing in Science & Engineering, 9(3), 10–20. 2007.
- [186] Müller K & Vignaux T. *SimPy: Simulating Systems in Python*. ONLamp.com Python DevCenter. 2003.
- [187] Gravier A. *PyCogMo: a neural network learning framework for Computational Neuroscience*. 2013. github.com/agravier/pycogmo.
- [188] Bohte SM, Kok JN & Poutré HL. *Error-backpropagation in temporally encoded networks of spiking neurons*. Neurocomputing, 48(1–4), 17 – 37. 2002.
- [189] Hodgkin AL & Huxley AF. *A quantitative description of membrane current and its application to conduction and excitation in nerve*. J Physiol, 117(4), 500–544. 1952.
- [190] Fitzhugh R. *Impulses and Physiological States in Theoretical Models of Nerve Membrane*. Biophys J, 1(6), 445–466. 1961.
- [191] Izhikevich E & FitzHugh R. *FitzHugh-Nagumo model*. Scholarpedia, 1(9), 1349. 2006.
- [192] Zador AM, Agmon-Snir H & Segev I. *The morphoelectrotonic transform: a graphical approach to dendritic function*. J Neurosci, 15(3 Pt 1), 1669–1682. 1995.
- [193] Dayan P & Abbott LF. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. The MIT Press. 2005.

- [194] Bower JM & Beeman D. *The book of GENESIS*. Springer-Verlag. 1998 (out of print), 2003 (Internet edition).
- [195] Ermentrout B. *Simulating, analyzing, and animating dynamical systems: a guide to XPPAUT for researchers and students*. Society for Industrial Mathematics. 2002.
- [196] Cannon R, Hasselmo M & Koene R. *From biophysics to behavior: Catacomb2 and the design of biologically-plausible models for spatial navigation*. *Neuroinformatics*, 1, 3–42. 2003. 10.1385/NI:1:1:003.
- [197] Versace M, Ames H, Léveillé J et al. *KInNeSS: A modular framework for computational neuroscience*. *Neuroinformatics*, 6(4), 291–309. 2008.
- [198] Rochel O, Martinez D et al. *An event-driven framework for the simulation of networks of spiking neurons*. In: *Proc. 11th european symposium on artificial neural networks*, 295–300. Citeseer. 2003.
- [199] Wilson E, Goodman P & Harris Jr F. *Implementation of a biologically realistic parallel neocortical-neural network simulator*. In: *Proc. of the 10th SIAM Conf. on Parallel Process. for Sci. Comput.* Citeseer. 2001.
- [200] Eliasmith C & Stewart T. *Nengo and the Neural Engineering Framework: Connecting Cognitive Theory to Neuroscience*. 2005.
- [201] Eppler J, Helias M, Muller E et al. *PyNEST: a convenient interface to the NEST simulator*. *Frontiers in neuroinformatics*, 2. 2008.
- [202] Weitzenfeld A, Arbib M & Alexander A. *The neural simulation language: a system for brain modeling*. The MIT Press. 2002.
- [203] Delorme A & Thorpe S. *SpikeNET: an event-driven simulation package for modelling large networks of spiking neurons*. *Network: Computation in Neural Systems*, 14(4), 613–627. 2003.
- [204] Vibert J, Alvarez F & Kosmidis E. *XNBC V9: A user friendly simulation and analysis tool for neurobiologists*. *Neurocomputing*, 38, 1715–1723. 2001.

COLOPHON

This document was typeset using $\text{\LaTeX}$ and written using *emacs* and *TeXnicle*. The typographical look-and-feel and the overall style make use of the *classicthesis* package developed by André Miede, and available for both $\text{\LaTeX}$ and $\text{\LyX}$ at <http://code.google.com/p/classicthesis/>. The style was inspired by Robert Bringhurst’s seminal book on typography “*The Elements of Typographic Style*”. Most graphs were plotted with *Knitr*, using a *tikzDevice* backend and *R* or *Python* as frontend. Others were output by the *Emergent* enural simulator. Schemas were designed using *Inkscape*, with the *inkscapeztikz* output converter.

Final Version as of 9th December 2013.