



HAL
open science

Learning higher-order functions for computation, memorization and control with artificial neural networks

Mathieu Chalvidal

► **To cite this version:**

Mathieu Chalvidal. Learning higher-order functions for computation, memorization and control with artificial neural networks. Library and information sciences. Université Paul Sabatier - Toulouse III, 2023. English. NNT : 2023TOU30256 . tel-04502217

HAL Id: tel-04502217

<https://theses.hal.science/tel-04502217>

Submitted on 13 Mar 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)

Présentée et soutenue par :

Mathieu Chalvidal

le 17 Juillet 2023

Titre :

Apprentissage de fonctions d'ordre supérieur pour le calcul, la memorisation et le contrôle par réseaux de neurones artificiels.

École doctorale et discipline ou spécialité :

ED MITT : Domaine Mathématiques : Mathématiques appliquées

Unité de recherche :

CerCo (CNRS) / Carney Institute for Brain Sciences (Brown University)

Directeur/trice(s) de Thèse :

Ruffin VANRULLEN
Thomas SERRE

Jury :

Céline HUDELOT - Professeure des universités - CentraleSupélec
Hachem KADRI - Maître de conférences - Université Aix Marseille
Dmitry KROTOV - Docteur - IBM Labs
Mengye REN - Professeur adjoint - New York University
Ruffin VanRullen - CNRS
Thomas Serre - Brown University

Abstract

After decades of research, the question of how biological neural networks synthesize experience to serve higher-level cognitive processes (concept acquisition, systematic decision making, evaluative thinking, creativity...) still resists to a complete scientific understanding. At the same time, advances in large scale computation have enabled the development of their artificial counterparts, which currently power a revolution in machine intelligence. These simplified neural models developed through a global scientific trial-and-error process, present however evident behavioral differences and limitations compared to biological neural networks and provide unsatisfactory to little insight into the computational principles that evolution condensed in biological neurons. One evident limitation is the lack of compositionality and systematicity in artificial neural computation, determining their slow learning process, their poor robustness to contextual perturbations and revealing their current failure as a true functional programming paradigm. In this work, we propose to tackle these weaknesses from the problem perspective of learning higher-order functions (i.e functions of functions) with neural networks. Namely, we propose to leverage modern machine learning techniques while considering neural networks as controllable *functionals*, unearthing original computational mechanisms able to support complex cognitive faculties such as memorization or attention, in turn serving more abstractive machine intelligence. In particular, building on the clear epistemological trend that hand-designed methods are eventually replaced by computerized and self-executing solutions, we will consider the specific higher-order problem of learning to learn, i.e *meta-learning*. We show that granting neural networks the aptitude to design learning strategies with minimal solution constraints as a function of data, activity or choice history, can help explore new forms of adaptive programs, with application to low-shot learning, operator regression, robotic control or controllable generative modeling.

Keywords : Meta-learning, neural networks, higher-order functions, control theory, synaptic plasticity, reinforcement learning, neural operators, working memory

Résumé

Malgré des décennies de recherche, la compréhension des mécanismes d'apprentissage permettant aux réseaux de neurones biologiques d'articuler l'expérience vécue dans l'acquisition et l'exécution de fonctions cognitives supérieures (formation de concepts, décision systématique, pensée critique, créativité...) résiste encore à une description scientifique complète. Au même moment, les avancées du calcul informatique à grande échelle ont permis le développement de leurs homologues artificiels, qui sous-tendent actuellement une révolution en intelligence machine. Ces modèles simplifiés présentent toutefois d'importantes différences d'apprentissage en comparaison des réseaux naturels et apportent des explications insatisfaisantes sur les principes computationnels que l'évolution a favorisé au sein des réseaux biologiques. Le manque de compositionnalité et de systématisme constitue une limitation évidente des réseaux de neurones artificiels, et détermine leur lenteur d'apprentissage et leur faible robustesse face à des perturbations contextuelles, révélant leur échec en tant que paradigme de programmation fonctionnelle viable. Dans cette thèse, nous explorons ces faiblesses sous la perspective du problème d'apprentissage de fonction d'ordre supérieur (i.e fonctions de fonctions) par réseaux de neurones artificiels. Nous proposons de marier les techniques d'apprentissage actuelles à ces réseaux vus comme fonctionnelles contrôlables, proposant des mécanismes calculatoires originaux capables de supporter des fonctions cognitives complexes comme la mémorisation ou l'attention, qui servent à leur tour une intelligence machine plus abstraite. En particulier, nous considérons le problème du méta-apprentissage, consistant à remplacer les méthodes manuelle d'ajustement fonctionnel par des méthodes d'adaptation elle-même apprises. En donnant aux réseaux de neurones la possibilité de construire leur propre stratégies d'apprentissage en fonction des données, de leur activité et de leur historique décisionnel, nous explorons de nouvelles formes de programme adaptatifs, trouvant des applications diverses en apprentissage "low-shot", en régression d'opérateurs fonctionnels, en robotique ou pour le contrôle de modèles génératifs.

Mots-clés : Méta-apprentissage, réseaux de neurones artificiels, fonctions d'ordre supérieur, théorie du contrôle, plasticité synaptique, modèles énergétiques, opérateur neuronaux, mémoire de travail.

Contents

Abstract	1
Résumé	3
Aknowledgments	5
Contents	9
Originality	10
List of figures	13
Notations	16
1 Introduction	18
1.1 Artificial neural networks	19
1.1.1 Definition	19
1.1.2 Energetic formulation	22
1.1.3 Learning paradigms of artificial neural networks	24
1.1.4 Historical perspective	26
1.2 Relation to Neuroscience	28
1.2.1 Bio-inspiration for artificial neural networks	28
1.2.2 Reverse-engineering the brain	29
1.2.3 Missing pieces in artificial neural learning	31
1.3 Relation to control theory	31
1.3.1 Conceptual convergence	31
1.3.2 Control theoretical origin of deep learning	32
1.3.3 Gradient descent as neural energy control	33
1.3.4 Artificial networks as functionals	35
1.4 Challenges and stakes	37
1.4.1 Societal impact of deep learning	37
1.4.2 Current limitations of artificial networks	38
1.5 Contributions and Thesis organization	38
I Learning higher-order functions	42
2 Neurally controlled ODEs	45
2.1 Introduction	45
2.2 Neural ODEs	46

2.2.1	Background	46
2.2.2	Limit of NODE to function approximation	47
2.3	Neurally Controlled Ordinary Differential Equation	48
2.3.1	Motivation	48
2.3.2	General formulation	49
2.3.3	Open and closed-loop controllers	51
2.3.4	Training	53
2.4	Experimental results	57
2.4.1	Flows for annuli dataset	57
2.4.2	Real-time pattern memorization	57
2.4.3	Oscillatory dynamics: Van der Pol and Kuramoto	60
2.4.4	Supervised image classification	62
2.4.5	Implementation of generic NCODE module	64
2.5	Controlled Normalizing flows	65
2.6	Discussion	67
3	Learning Functional transduction	70
3.1	Context	71
3.2	Related work	73
3.3	The functional regression problem	75
3.3.1	Notations	75
3.3.2	The meta-regression problem	75
3.4	Reproducing Kernel Banach Space regression	76
3.5	Transducer	78
3.5.1	Discretization	78
3.5.2	Training	80
3.6	Applications to finite-dimensional meta-learning	83
3.6.1	MNIST-like datasets classification	83
3.7	Applications to ∞ -dimensional problems	83
3.7.1	Regression of Advection-Diffusion Reaction PDEs	83
3.7.2	Burger's equation in 2D	86
3.7.3	Climate modeling	88
3.8	Experimental details	90
3.8.1	Advection-Diffusion-Reaction operators	90
3.8.2	Burger's equation	93
3.8.3	Climate modeling	94
3.9	Discussion	94
II	On Meta-learning	97
4	Meta-learning and the free-energy principle	99
4.1	The free-energy principle	99

4.1.1	Hierarchy of inter-related learning systems	101
4.1.2	Inference and control	103
4.2	Current approaches of meta-learning	104
4.3	Potential directions and challenges	106
4.3.1	Explorable forms of neural control	106
4.4	Transition	110
5	Self-modifying networks	112
5.1	Introduction	112
5.2	Meta-Reinforcement Learning	113
5.2.1	Background	113
5.2.2	Connection to optimal transport	113
5.3	Synaptic plasticity	116
5.3.1	Artificial fast plasticity:	116
5.3.2	Potential mechanism for working memory	117
5.4	Reinforcement with meta-plasticity	117
5.4.1	Learning weights update rules	117
5.5	Experiments	119
5.5.1	Efficiency: One-shot reinforcement learning	120
5.5.2	Harlow task	120
5.5.3	Gym Mujoco directional robot control	122
5.5.4	Capacity : Maze exploration task	124
5.5.5	Generality : Motor control	126
5.6	Optimization	128
5.6.1	Gradient policy update	129
5.6.2	Discrete adjoint system	130
5.7	Experimental details	131
5.7.1	METHODS Optimization	132
5.7.2	Gradient policy update	133
5.7.3	Discrete adjoint system	133
5.8	Discussion	135
5.8.1	Synthesis	135
5.8.2	Outlook	136
6	Discussion	138
6.1	Synthesis	138
6.2	Looking forward	140
7	Résumé de la thèse en Français	142
7.1	Chapitre 1: Introduction	142
7.2	Chapitre 2 : Méta-apprentissage et le principe de l'énergie libre	144

7.3	Chapitre 3: Equations differentielles ordinaires neurales controlées	146
7.3.1	Formulation générale	147
7.4	Chapitre 4: Meta-apprentissage par renforcement de réseaux self-référentiels	147
7.5	Chapitre 5: Meta-apprentissage par transduction fonc- tionnelle	148
	Bibliography	152

Originality

Statement

This manuscript is the compilation of the work produced towards the obtention of a doctoral degree from *Université Toulouse III - Paul Sabatier* over the last three years, between the *Brain and Cognition Research Center* at CNRS, Toulouse and the *Carney Institute for Brain Sciences* at Brown University, US. It is an original and personal work. The writing is my own and material used to illustrate discussions is either my production or drawn from relevant prior work with corresponding references. Additionally, I list below the different written contributions (either published or publicly archived) produced over the course of this PhD.

First author papers

This thesis manuscript contains material taken from the following personal research, ordered in chronological order:

Go with the Flow: Adaptive control for Neural ODEs

Mathieu Chalvidal, Matthiew Ricci, Rufin VanRullen, Thomas Serre

International Conference on Learning Representation 2021

<https://openreview.net/pdf?id=giit4HdDNa>

Meta-Reinforcement Learning with Self-modifying Networks

Mathieu Chalvidal, Thomas Serre, Rufin VanRullen

Neural Information Processing Systems 2022

<https://openreview.net/pdf?id=cYeYzaP-5AF>

Learning Functional Transduction

Mathieu Chalvidal, Thomas Serre, Rufin VanRullen

Neural Information Processing Systems 2023 - Spotlight paper

<https://arxiv.org/abs/2302.00328>

Additional contributions

The following list regroups additional pieces of work for which I contributed either theoretically or experimentally and that concern indirectly this thesis discussion:

Look at the variance! Efficient black-box explanations with Sobol-based sensitivity analysis

Thomas Fel, Rémi Cadène, **Mathieu Chalvidal**, Matthieu Cord, David Vigouroux, Thomas Serre

Neural Information Processing Systems, 2021

<https://openreview.net/pdf?id=hA-PHQG0jqQ>

KuraNet: systems of coupled oscillators that learn to synchronize

Matthew Ricci, Minju Jung, Yuwei Zhang, **Mathieu Chalvidal**, Aneri Soni, Thomas Serre

arXiv:2105.02838, 2021

<https://arxiv.org/abs/2105.02838>

Accurate implementation of computational neuroscience models through neural ODEs

Sabine Muzellec, **Mathieu Chalvidal**, Thomas Serre, Rufin VanRullen
Conference on Cognitive Computational Neuroscience, 2022

<https://2022.ccneuro.org/proceedings/0000090.pdf>

Block Delayed Majorize-Minimize Subspace Algorithm for Large Scale Image Restoration

Mathieu Chalvidal, Emilie Chouzenoux, Jean-Baptiste Fest and Claire Lefort

Inverse Problems, 2023

<https://hal.science/hal-03764677/>

FAMOUS: a fast instrumental and computational pipeline for multiphoton microscopy applied to 3D imaging of muscle ultrastructure

Claire Lefort, **Mathieu Chalvidal**, Alexis Parenté, Véronique Blanquet, Henri Massias, Laetitia Magnol, Emilie Chouzenoux

Journal of Physics D: Applied Physics, 2021

<https://hal.science/hal-03764677/>

Invited talks

- *Brown Unconference* - Virtual 29-30/06/2020
- *ANITI Seminar* - Virtual - 29/10/2020
- *Int. Conf. on Mathematical Neuroscience* - Virtual - 29/06/2021
- *Lab. of Neural Computation and Cognition* - Brown University - 15/02/2022
- *Crunch Group* - Brown University - 06/04/2022
- *Torus AI* - Toulouse - 09/02/2023

Ressources

Github repositories containing the implementation of published papers

- <https://github.com/mathieuchal/NCODE>
- <https://github.com/mathieuchal/METHODS>
- <https://github.com/mathieuchal/LFT>

List of Figures

1.1	Comparison between a cortical pyramidal neuron and an artificial neuron	19
1.2	Schema of activation propagation in neural networks	21
1.3	Varying trajectories of neurons activations in a simple 2 neurons network	22
1.4	Variational inference as a network energy control problem	26
1.5	Synergies between Neuroscience and Deep learning	29
1.6	Grid-like representations similar to mammalian-brain emerging in artificial neural agents	30
1.7	Evolution of a neural network energy profile over training	35
1.8	Energy as a function of gradient iterations	36
1.9	Perturbation of the energy function with a single neuron	37
2.1	2D vector field defined by a parameterized by a one-layer perceptron	48
2.2	Diagram of a general N-CODE module	50
2.3	Idealized dynamical computations of neural modules governed by ODEs	51
2.4	Three types of continuous-time neural networks trajectories	54
2.5	The concentric annuli problem	54
2.6	Time-varying flows for two points in different classes (inner: top row; outer: bottom row) of the annuli dataset. Time advances from left to right.	57
2.7	Diagram of an episode of the few-shot memorization	58
2.8	Error-rate of NCODE on the 1000-bit reconstruction task	59
2.9	Example of dynamical control augmentation: The Van der Pol oscillator	61
2.10	Example of dynamical control augmentation: The Van der Pol oscillator	62
2.11	Accuracy, Train and test losses over training iterations and Number of function evaluations (NFE) vs loss for several versions of NCODE on CIFAR-10.	63

2.12	Two-dimensional vector fields and energetic probabilities of normalizing flow	67
3.1	Schema of a <i>Transducer</i> layer	79
3.2	Experimental results for in-context regression of ADR equations	81
3.3	Comparison of meta-test accuracies of MNIST-like datasets classification task presented in ? against the Transducer.	83
3.4	Example of the extrapolated trajectory of the Transducer on ADR regression	86
3.5	Illustrative example of initial, target and estimation of the vector field for the Burger’s equation	86
3.6	Up: Meta-test regression and outlier detection results at two target times. RMSEs on Burgers’ equations averaged over 200 different parameter conditions $\nu \in [0.1, 0.5]$ each with 100 train examples. Precision/Recall in outlier detection of the Transducer versus Spectral clustering. Bottom: RMSE distributions of each element in the contaminated dataset over the 5000 regressions. Outliers are clearly identified.	88
3.7	Illustrative example for the climate prediction experiment	89
3.8	Examples of sampled functions $\delta(x)$ and ν	91
3.9	Examples of advection-diffusion-reaction datasets	92
3.10	Examples of sampled function for the spatial coefficients	93
3.11	Examples of evolution of state for the ADR operator	93
3.12	Magnitude of the complex coefficients of the Fourier transform	93
3.13	Illustrative examples of initial ($t = 0$), target ($t = 10$) and <i>Transducer</i> estimation	95
5.1	Meta-Reinforcement Learning as a transport problem and schema of MetODS synaptic adaptation	114
5.2	Summary of experiments on METODS efficiency	121
5.3	Schema of the Harlow task	122
5.4	Reward profiles of MAML, RL ² and MetODS over 5 consecutive episodes in the Harlow task	123
5.5	Examples of procedurally generated maze configurations	124
5.6	Performance of Meta-RL models for the maze experiment	125
5.7	Heatmap of activation rate of three neurons of a trained model across 10 different maze maps	126
5.8	Meta-training results for MetaWorld and Mujoco benchmarks	128

7.1	Meta-training results for MetaWorld and Mujoco benchmarks	149
7.2	Résultats expérimentaux pour la régression contextuelle des équations ADR	151

Notations

Numbers and arrays

a A scalar value from a field \mathbb{K}

\mathbf{a} A vector

\mathbf{A} A matrix

\mathbf{I}_n The identity matrix with n rows and n columns

$\text{diag}(\mathbf{a})$ A diagonal square matrix with diagonal entries given by \mathbf{a}

Sets & spaces

\mathcal{S} A set

\mathcal{X} A vector space

$\mathbf{x} \in \mathcal{X}$ An element of \mathcal{X}

$[a, b]$ The real interval including a and b

$[a, b[$ The real interval including a and excluding b

$\{0, \dots, n\}$ The set of all integers between 0 and n

$\mathcal{A} \setminus \mathcal{B}$ The restriction set of elements of \mathcal{A} not in \mathcal{B}

Indexing

\mathbf{a}_i An element at indices i of vector \mathbf{a}

$\mathbf{A}_{i,j}$ An element at indices i, j of matrix \mathbf{A}

\mathbf{a}^l A vector of activation of l th layer of a neural network

Functions

$f : \mathcal{X} \mapsto \mathcal{Y}$ A function between \mathcal{X} and \mathcal{Y}

- $f \circ g$ Composition of function f and g
- f_θ A function parameterized by θ
- $\log(\mathbf{x}), e^{\mathbf{x}}$ Natural logarithm and exponential of \mathbf{x}
- $\log(\mathbf{x}), e^{\mathbf{x}}$ L^p norm of \mathbf{x}
- $\mathbf{1}_S$ The indicator function of the set S
- \mathcal{C}^p The set of p times differentiable functions.

Calculus

- $\frac{\partial \mathbf{x}}{\partial \mathbf{y}}$ The partial derivative of \mathbf{x} with respect to \mathbf{y}
- $\frac{d\mathbf{x}}{d\mathbf{y}}$ The derivative of \mathbf{x} with respect to \mathbf{y}
- $\nabla_{\mathbf{x}} f$ The gradient of f with respect to \mathbf{x}
- $\mathcal{J}(f)(\mathbf{x})$ The Jacobian of f with respect to \mathbf{x}
- $\int_S f(\mathbf{x}) d\mathbf{x}$ The Lebesgue integral of f over S .
- $\dot{\mathbf{x}} = \frac{d\mathbf{x}}{dt}$ the temporal derivative of \mathbf{x}

Probability and Information Theory

- ν A probability measure
- $\nu(\mathbf{x})$ The probability of event \mathbf{x}
- $p_\nu(\mathbf{x})$ The probability density function of measure ν applied to \mathbf{x}
- $\mathbf{x} \sim F$ Variable \mathbf{x} follows distribution F
- $\mathbb{E}_\nu[f(\mathbf{x})]$ Expectation of $f(\mathbf{x})$ with respect to measure ν
- $V_\nu(f(\mathbf{x}))$ Variance of $f(\mathbf{x})$ with respect to measure ν
- $Cov_\nu(f(\mathbf{x}), g(\mathbf{y}))$ Covariance of $f(\mathbf{x}), g(\mathbf{y})$ with respect to measure ν
- $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ Multivariate Gaussian dist. of mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$

Chapter 1

Introduction

“The universe is built on a plan the profound symmetry of which is somehow present in the inner structure of our intellect.”

Paul Valery

Contents

1.1	Artificial neural networks	19
1.2	Relation to Neuroscience	28
1.3	Relation to control theory	31
1.4	Challenges and stakes	37
1.5	Contributions and Thesis organization	38

This thesis aims at proposing a coherent perspective for different projects exploring original computational principles for the manipulation of higher-order functions (i.e functions of functions) with artificial neural networks, in particular, regarding the problem of learning to learn. This approach lies at the interface of **deep learning**, **dynamical systems** and **meta-learning**, while taking inspiration from neuroscience and control theory. I start by presenting past and current research in artificial neural networks in the form of a historical note as well as its interaction with these two fields. The remaining chapters are dedicated to the theoretical and experimental contributions of this thesis. The first part presents two original forms of functional regression with neural networks. The second part is dedicated more specifically to the problem of neural meta-learning and its connection with the free-energy minimization principle. I hope that this document will prove useful, not only personally for graduation purpose, but also as inspiration towards new approaches for designing and controlling artificial neural networks. Along the way, I shall con-

nect my discussion with various applications from generative modeling, robotics or physical science to contextualize and motivate this work.

1.1 Artificial neural networks

1.1.1 Definition

Artificial neural networks refers to a broad class of dynamic graphs that can be generally described as sequences of 3-tuples $(\mathcal{G}_t)_{t \in \mathcal{T}} = (V_t, E_t, s_t)_{t \in \mathcal{T}}$ indexed over an ordered set \mathcal{T} , where V_t correspond to the set of processing units at index t (or vertices V_t) linked together by set of edges E_t (i.e a set of pairs of vertices in V_t). Taken collectively, such units are able to carry and transform information, in the form of an analogic and distributed code of "activations" $\mathbf{x}_t = s_t(V_t)$ given by the functions $s_t : \mathcal{V} \mapsto \mathcal{X}$ which associates to vertices V_t a value in a set \mathcal{X} according to particular message-passing processes between units and as a function of some parameters $\boldsymbol{\theta}_t$ belonging to a space Θ . Because such networks mimic the organization of nervous systems found in nature with local computation rules that are inspired from biology, their processing units are called "neurons".

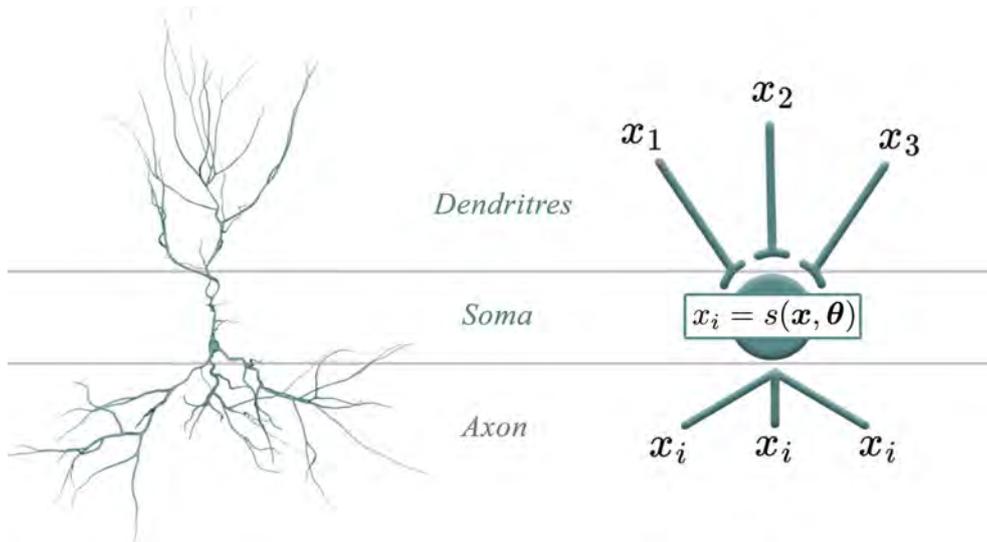


Figure 1.1 – **Comparison of biological pyramidal neurons and its artificial idealization.** Pyramidal neurons possess identifiable body structures that allows for information integration through dendritic and somatic bodies and transfer through axonal connections. This structure is replicated in artificial neurons in the form of a parametric transfer function $s(\mathbf{x}, \boldsymbol{\theta})$ relating inputs stimuli to an activation state \mathbf{x}_i .

The precise structural organization of networks (connectivity and hierarchy of neurons determined by E_t) as well as the principles governing message aggregation and passing in the network which determines the expression of functions $(s_t)_{t \in \mathcal{T}}$ can vary widely and describe vastly different types of dynamics (see fig 1.3 for a simple example). In this work, we will mostly consider artificial neural networks as dynamical systems of a potentially large number of variables referred to as the state $\mathbf{h}_t = (\mathbf{x}_t, \boldsymbol{\theta}_t)$ of the network over time, such that for all $t \in \mathcal{T}$, \mathbf{h}_t is a multidimensional vector belonging to a state space $\mathcal{H} = \mathcal{X} \times \Theta$ most often identified to the real-valued euclidean space \mathbb{R}^p for $p \in \mathbb{N}$. We will show below that this generic definition will allow to unify our treatment of several temporal processes of interest such as network inference and learning, as well as different network typologies such as deep feed-forward and recurrent network. Specifically, we will show below that the partial dynamics governing the evolution of neural activations \mathbf{x}_t can be seen as particular arrangement preferences in the space \mathcal{X} , which can in turn serve to form complex and distributed computational programs relating pieces of information by encoding them in the form of subsets of neurons activations at particular times.

Definition 1 *The states of an artificial neural network $(\mathcal{G}_t)_{t \in \mathcal{T}}$ can be seen as a dynamical system governed by a function $\Phi : \mathcal{H} \times \mathcal{T} \mapsto \mathcal{H}$ defined over a vector space \mathcal{H} and a monoid \mathcal{T} (usually real positive numbers \mathbb{R}^+ or natural numbers \mathbb{N}) such that:*

$$\forall \mathbf{h} \in \mathcal{H}, \quad \Phi(\mathbf{h}, 0) = \mathbf{h} \quad (1.1)$$

$$\forall t, t' \in \mathcal{T}^2, \quad \Phi(\mathbf{h}, t + t') = \Phi(\Phi(\mathbf{h}, t), t') \quad (1.2)$$

Φ is called the **flow** or **evolution map** of the dynamic system. Furthermore, we define the state of the system as $\mathbf{h}_t = \Phi(\mathbf{h}, t)$. In many cases, when $\mathcal{T} = \mathbb{N}$, Φ is implicitly defined through the following iteration map F

$$\forall t \in \mathbb{N}, \quad \mathbf{h}_{t+1} = F(\mathbf{h}_t, t) \quad (1.3)$$

Remark: Defining neural networks as dynamical systems allows us to consider a wide range of previously defined networks with the same theoretical treatment. For instance, 'classic' feedforward computation in neural networks (as represented in figure 1.2) can be thought as dynamical system where neurons activations are partially updated following a specific sequence of message-passing operations until we reach neurons

considered as output of the system. Recurrent neural networks can be directly associated to this definition. (see in particular chapter 2 and 5)

Remark II: Although we will mostly consider discrete sequences of operations, this definition will also let us manipulate continuous-time neural networks in chapter 2. Note that this work will focus more specifically on the class of "rate-based" neural networks, which does not consider the exact temporality of signal passing at the neuron level as in spiking neural networks but rather consider sequences of synchronized operations between neurons.

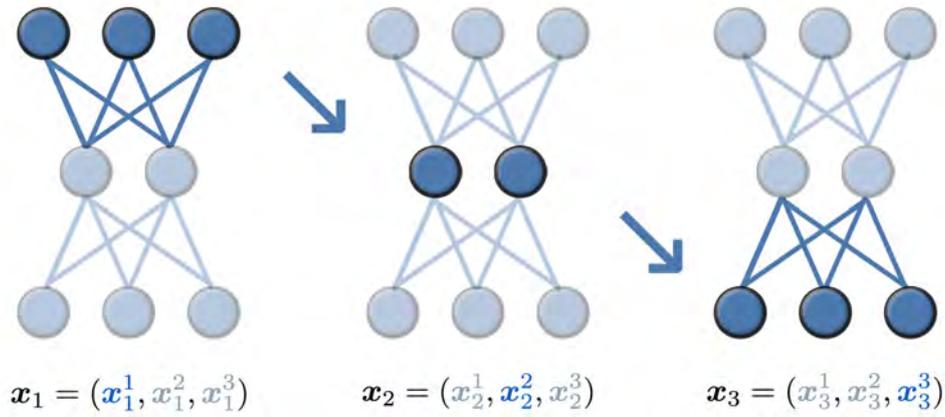


Figure 1.2 – **Schema of activation propagation between three subsets of neurons.** Our definition encompass several forms of neural network architectures as message passing dynamics between neurons. Here is depicted a feedforward network with two sequential operations affecting alternatively three neurons subsets $\{\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3\}$. At each state, neurons with states varying from previous time-step are shown in blue.

Often, the state of a finite subset of neurons at time t is identified to that of variables of the external world that act as stimuli over the network, and form a vector \mathbf{x}_t^{in} belonging to a multidimensional *input* space \mathcal{X}_{in} . Additionally, a (potentially overlapping) subset of neurons state at a subsequent time t' can be identified as a vector $\mathbf{x}_{t'}^{out}$ in the *output* space \mathcal{X}_{out} and represent the influence of the network over its environment. As stated above, neurons connections and transfer functions defining F in equation 1.3, are potentially defined as parametric objects, most simply depending on a vector-valued parameters $\boldsymbol{\theta} = \{\theta_1, \theta_2, \dots, \theta_n\}$ where n is

a finite natural number. From a learning theory perspective, considering various ways in which networks structure and parametrisation can vary amounts to describe an *hypothesis class* $\mathcal{F}(\mathcal{X}_{in}, \mathcal{X}_{out})$ of possible network response functions $\mathbf{x}_t^{in} \mapsto \mathbf{x}_t^{out}$ between input and output spaces. Because

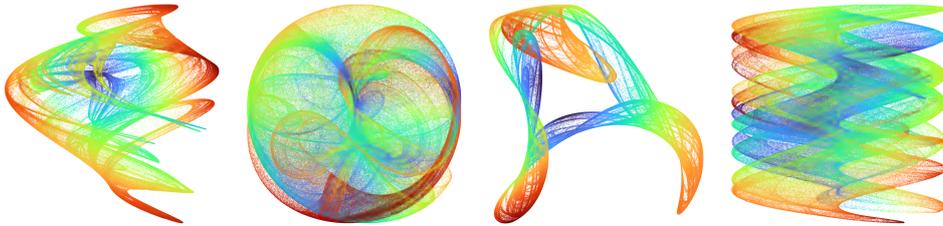


Figure 1.3 – **Varying trajectories of neurons activations in a simple 2 neurons network.** Neurons state evolution is specified by the equation $\dot{x}_1 = \sin(ax_2) + c\cos(ax_1)$ and $\dot{x}_2 = \sin(bx_1) + d\cos(bx_2)$ with varying random parameters a, b, c, d for the 4 cases. Small parameters variations generates large attractor differences in the network state space.

of their highly combinatorial and parametric nature, neural networks have high algorithmic complexity and theory or intuition fail to describe precisely the influence of specific network configurations and parameters to the expressed function. Hence, finding automatic and efficient ways to endow the network response function with desired properties, a process that we will refer to as *neural network functional control*, will be the principal problem explored in this work. Specifically, we will see that such problem can be seen as building higher-order functions operating on the class of applications defined by the network dynamics and that such function can be interpreted from an energetic perspective to enforce a desired functional behavior.

Research question 1

What factors drives the emergence of coherent functional associations in apparently chaotic artificial neural dynamics? Can we build higher-order processes able to orchestrate and compose such dynamics in order to model general cognitive functions?

1.1.2 Energetic formulation

Energetic descriptions of relations and interactions between physical systems are ubiquitous in science, and artificial neural networks make no exception. Most generally, the dynamics governing the evolution of states

\mathbf{h}_t in the network can be characterized from the perspective of a defined *energy function* E dependant on parameters $\boldsymbol{\mu}$ and mapping different neurons states over time $\mathbf{h}_t, \dots, \mathbf{h}_{t'}$ to real numbers:

$$E(\mathbf{h}_t, \dots, \mathbf{h}_{t'}, \boldsymbol{\mu}) \in \mathbb{R} \tag{1.4}$$

Defining such energy function allows in turn to describe a probability distribution via the Boltzmann distribution $p_{\boldsymbol{\theta}}(\mathbf{h}_t, \dots, \mathbf{h}_{t'})$ over the cardinal space $H = \mathcal{H} \times \dots \times \mathcal{H}$ with partition function $Z(\boldsymbol{\mu})$, assuming existence of the integral:

$$p_{\boldsymbol{\theta}}(\mathbf{h}_t, \dots, \mathbf{h}_{t'}) = \frac{e^{-E(\mathbf{h}_t, \dots, \mathbf{h}_{t'}, \boldsymbol{\mu})}}{Z(\boldsymbol{\theta})} \tag{1.5}$$

$$Z(\boldsymbol{\mu}) = \int e^{-E(\mathbf{h}_t, \dots, \mathbf{h}_{t'}, \boldsymbol{\mu})} d_H \tag{1.6}$$

As we will see particularly in chapter 4, these two definitions are not only powerful tools to understand the mechanics of neural networks information processing by connecting our discussion with statistical physics, they also offer techniques for "aligning" networks dynamics with variables and phenomenons observed in reality by allowing to define a classic optimization problem allowing to form a model selection rule within an hypothesis space of dynamics. More precisely, they can also describe the particular aspect of information encoding from a statistical physics perspective. There are many different ways of defining such neural energy function depending on the task and network definition. We show below, two particular examples:

◇ **Example 1:** A very simple neural energy model can be formed by defining an energy associated to the network as the state of a single neuron activation x_t^{out} at time t , after initially forcing the network activations \mathbf{x}_0 (see figure 1.4 for an illustration):

$$E\left(\underbrace{\mathbf{x}_0}_{\text{stimuli}}, \dots, \underbrace{\mathbf{x}'_t}_{\text{internal states}}, \dots\right) = \underbrace{x_t^{out}}_{\text{output}} \tag{1.7}$$

◇ **Example 2:** Autoencoding networks form a second class of energy models whose energy can be written as the distance between inputs and outputs neurons after a sequence of internal representations:

$$E(\mathbf{x}_0, \dots, \mathbf{x}'_t, \dots) = \|\mathbf{x}_0 - \mathbf{x}_t\|_{\mathcal{X}} \tag{1.8}$$

This formulation has been particularly explored in the context of *generative modeling*.

1.1.3 Learning paradigms of artificial neural networks

Although not explicitly framed in the above-mentioned energetic formulation, many research fields in machine learning are nowadays dominated by neural energy control approaches. *Supervised learning* (Cunningham u. a., 2008), which consists in optimizing explicitly a certain energy function through the parameters θ in order to approximate of a specific reference function $f : \mathcal{X} \mapsto \mathcal{Y}$ given available examples pairs $(\mathbf{x}^i, f(\mathbf{x}^i))_{i \in I}$, has been a prominent paradigm and have led to breakthrough in different domains, for instance in computer vision for face, objects, scene or activity recognition (Krizhevsky u. a., 2012; Simonyan und Zisserman, 2014; He u. a., 2016).

Supervised learning

Let us define \mathcal{X} and \mathcal{Y} two vector spaces and let us further assume that \mathcal{X} is a measure space $(\mathcal{X}, \mathcal{A}, \mathbb{P}_{\mathbf{x}})$ with measure $\mathbb{P}_{\mathbf{x}}$ for the σ -algebra \mathcal{A} . We also define an objective function $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}$ and a class $\mathcal{F}_{\Theta} \subset \mathcal{F}(\mathcal{X}, \mathcal{Y})$ of functions f_{θ} from \mathcal{X} to \mathcal{Y} and parameterized by vectors $\theta \in \Theta$, where Θ represent a set of admissible parameters.

Supervised learning of a target function f consists in the optimization problem that finds the best approximation of f in the sense of \mathcal{L} and $\mathbb{P}_{\mathbf{x}}$, (assuming the existence of the following integral):

$$\inf_{\theta \in \Theta} \mathbb{E}_{\mathbb{P}_{\mathbf{x}}} [\mathcal{L}(f_{\theta}(\mathbf{x}), f(\mathbf{x}))] = \inf_{\theta \in \Theta} \int_{\mathcal{X}} \mathcal{L}(f_{\theta}(\mathbf{x}), f(\mathbf{x})) \mathbb{P}_{\mathbf{x}}(\mathbf{x}) \quad (1.9)$$

In practice, given the intractability of the integral, problem 1.9 is approximated through a Monte-Carlo approximation, provided a countable collection of available reference pairs $(\mathbf{x}_i, f(\mathbf{x}_i))_{i \in I}$ and assuming existence of a minimizer:

$$\min_{\theta \in \Theta} \sum_{i \in I} [\mathcal{L}(f_{\theta}(\mathbf{x}_i), f(\mathbf{x}_i))] \quad (1.10)$$

In the context of supervised learning with neural networks defining functions $f_{\theta} : \mathcal{X} \mapsto \mathcal{Y}$ by associating neurons states $\mathbf{x}_{in} \mapsto f_{\theta}(\mathbf{x}_{in}) = \mathbf{x}_{out}$, this problem amounts to find a network such that the following

energy function E_{θ} is minimized over examples:

$$E(\mathbf{x}_{in}, \mathbf{x}_{out}, \theta) = \mathcal{L}(f_{\theta}(\mathbf{x}), f(\mathbf{x})) \quad (1.11)$$

Remark: Interestingly, this framework called “empirical risk minimization” goes beyond neural supervised learning and encompasses any methodology aiming at inducing a particular function from a parametric class of models \mathcal{F}_{Θ} . This framework find its origin in the method of least squares, which introduced by Legendre in 1805 (Farebrother, 2001) and have further been developed for linear or kernel regression, naives Bayes classifiers or polynomial interpolation.

The realization that neural networks representations could be effectively bootstrapped to form feedback training loop also determined the rise of another paradigm called *unsupervised learning*. This paradigm has found important developments for learning useful representations and transformations with auto-encoding (Kane, 1994; Salakhutdinov u. a., 2007; Mikolov u. a., 2013a) or adversarial networks (Goodfellow u. a., 2020). In particular, an important field where unsupervised neural learning has been highly successful is natural language processing with applications in language modeling (Jozefowicz u. a., 2016; Sutskever u. a., 2014), parsing (Vinyals u. a., 2015) or translation (Bahdanau u. a., 2014). Another important development of learning with neural networks which we already hinted at with the energetic formulation consists in its combination with tools from Bayesian inference (Kingma und Welling, 2013) allowing to extract and manipulate the generative structure of high-dimensional complex data distributions. This area of research is nowadays highly active with different emerging techniques such as score-matching (Hyvärinen, 2005; Song u. a., 2020) or likelihood-based modeling (auto-regressive models (Van Den Oord u. a., 2016), normalizing flows (Dinh u. a., 2016; Grathwohl u. a., 2019) or explicit energy-based modeling (LeCun u. a., 2006; Song und Kingma, 2021)). Current state of research is exploring how such representations can emerge from self-referential problem formulations as in *self-supervised learning* or how they could usefully *transfer* to new downstream tasks. Finally, another highly important development in artificial neural learning has been *deep reinforcement learning* Mnih u. a. (2013). The combination of neural networks with the theory of Markov decision processes has enabled great progress to study decision making and enabled the creation of autonomous agents able tackle automatically hard sequential tasks in complex environments.

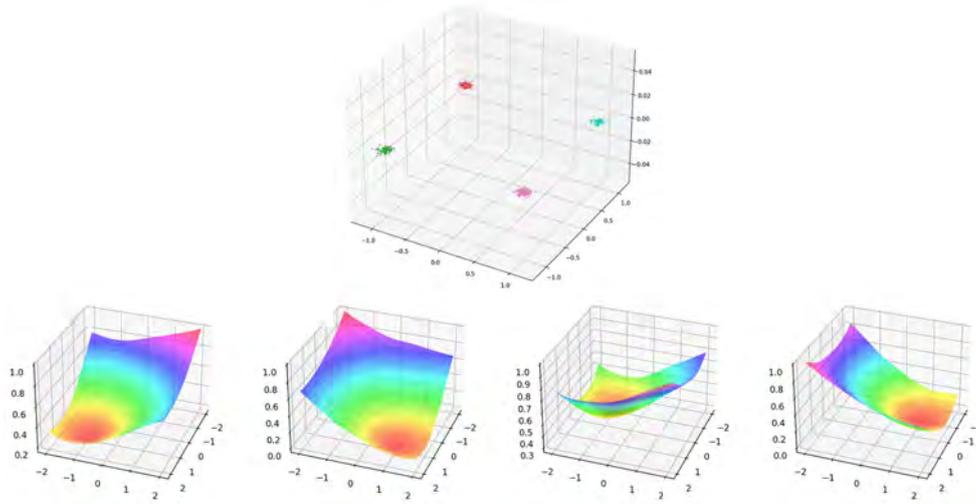


Figure 1.4 – **Variational inference as a network energy control problem.** Learning a probability distribution of alternative external events (here alternatives are represented by 4 different clusters of points) with a neural network as defined in equation (1.7) amounts to control the energy landscape expressed by the network. One important question is how can we leverage data regularities (symmetries in this case) to learn efficient control over the network belief.

1.1.4 Historical perspective

Pioneering work that led the way towards the introduction of artificial neural networks and underscore their physical energetic interpretation can be dated back from the study of thermodynamic systems of interacting particles such as the Ising model in the 1920's. Later work from (McCulloch und Pitts, 1943) formally introduced artificial networks while the idea of learning functions first appeared in the form of local update rule referred as hebbian learning (Hebb, 1949). At the end of the 1950's, Rosenblatt (1958) considered more complex multi-layered systems. The increase in networks depth called for more efficient methods of training (Rosenblatt, 1961; Ivakhnenko, 1971), which determined the emergence of backpropagation, that is the automatic propagation and accumulation of errors to internal nodes with respect to specific network target state (Linnainmaa, 1976; Werbos, 1982; Rumelhart u. a., 1985) which was further coupled with stochastic gradient descent (Robbins und Monro, 1951) by Amari (1967). This technique constitutes nowadays the most popular approach for controlling networks, due to its stability and the strong theory of gradient-based optimization to which

it is associated. Since then, many coinciding factors have determined a steady and branching out development of artificial neural networks into the full-fledged research field of machine intelligence called *Deep learning* (LeCun u. a., 2015). A first factor has consisted in the technological progress in computers power and memory (see for example Moore’s law Moore (2006), culminating nowadays in GPU/TPU-accelerated computing (Garland, 2011). A second factor is accessibility and open-sourcing of resources, from data availability such as normative large-scale datasets like the *UC Irvine public dataset repository* and later *Image-Net* for vision (<https://www.image-net.org/>), to the creation of widely accessible public knowledge consolidation platforms such as *arXiv* (<https://arxiv.org/>) , or more recently *Kaggle* (<https://www.kaggle.com/>) and *PaperWithCode* (<https://paperswithcode.com/>), as well as modern highly documented and modular public tensor programming libraries such as *Tensorflow*, *Pytorch* or *JAX* allowing to perform large scale automatic differentiation. Finally, key computational mechanisms and methodologies have been added to the deep learning bestiary, from architectural hypothesis such as recurrence (Hopfield, 1982b; Rumelhart u. a., 1985; Hochreiter und Schmidhuber, 1997a), convolutions ((Fukushima, 1980; LeCun u. a.)), residual connections (He u. a., 2016) and attentional modulation (Vaswani u. a., 2017), or advances in gradient-based optimization (see Sun (2019a) for a recent overview). All these factors contributed to the unmatched abilities of current networks for data modeling, which determined the growth of scientific and engineering interest for neural-based solutions as well as a rapid expansion into the different form of discussed artificial neural learning frameworks

While these frameworks have produced powerful solutions to highly sophisticated problems throughout science and industry such as protein folding prediction (Jumper u. a., 2021), playing Go at super-human level (Silver u. a., 2017), nuclear reaction control (Degraeve u. a., 2022), modeling 3D scenes from sparse viewpoints observations (Mildenhall u. a., 2020) or generating photo-realistic images (Ramesh u. a., 2021), research in artificial neural networks has also unveiled open problems that remain to be solved: A synthetic question could be *whether these current purely connectionist approaches will be able to reach the cognitive flexibility that remain a hallmark of human intelligence*. Theoretical results delineating the class of solvable problems with neural networks have been proposed: For instance, pieces of work proved universal approximation theorems for functions and operators (Hornik u. a., 1989; Cybenko, 1989; Chen und Chen, 1995; Lu u. a., 2017) or showed the Turing-completeness of recurrent neural networks (Siegelmann und Sontag, 1995). Many differ-

ent approaches have notably tried to build a theory of deep learning, either in terms of kernel machines (Jacot u. a., 2018), information theory, (Hjelm u. a., 2018; Tishby und Zaslavsky, 2015), control theory (Liu und Theodorou, 2019), or topological and computational complexity theory with Vapnik–Chervonenkis dimension (Bartlett und Maass, 2003) or persistent homology (Bianchini und Scarselli, 2014; Hofer u. a., 2017). Additionally, much remain to be understood regarding the influence of neural networks parametrization regimes over functions learnability. For instance, the double descent phenomenon (Poggio u. a., 2020) as well as emerging properties recently observed in very-large scale models (Wei u. a., 2022) suggest that neural networks escape understanding of conventional tools from statistical learning. Finally, from an optimization perspective, better describing the interaction between various classes of neural networks and objective functions determining specific loss landscapes, as well as their relation with stochastic gradient methods is also a crucial research direction to characterize the properties and performance of the found solutions. Finally, one important perspective that we further develop below is to qualify the difference of artificial networks with respect to their biological counterparts.

1.2 Relation to Neuroscience

1.2.1 Bio-inspiration for artificial neural networks

This work is also directly inspired by the research program defended by Rufin VanRullen and Thomas Serre regarding the necessary dialogue between Neuroscience and Deep Learning (see Hassabis u. a. (2017) for a recent review). Since early work (Hinton, 1984), artificial neural networks have been noted to account well for previously unexplained human physiological behaviors, which determined the key influence of neuroscientific insights on their subsequent development. Many aspects of the current techniques used for building artificial networks have been motivated by neuroscientific considerations; for instance, the replication of known structures of biological connectomes (Fukushima, 1980; Serre u. a., 2007; Yamins und DiCarlo, 2016; Krizhevsky u. a., 2017), the explicit modeling of particular cognitive functions such as attention (Larochelle und Hinton, 2010; Gregor u. a., 2015; Reed u. a., 2016; Vaswani u. a., 2017) and memory (Hopfield, 1982b; Hochreiter und Schmidhuber, 1997a; Mnih u. a., 2014), the addition of natural regularization constraints Lyu und Simoncelli (2008); Srivastava u. a. (2014); Kubilius u. a. (2019) or even the embedding of broad computational theories of natural learning (Oord u. a., 2018). Furthermore, neuroscience not only actively helps inform the

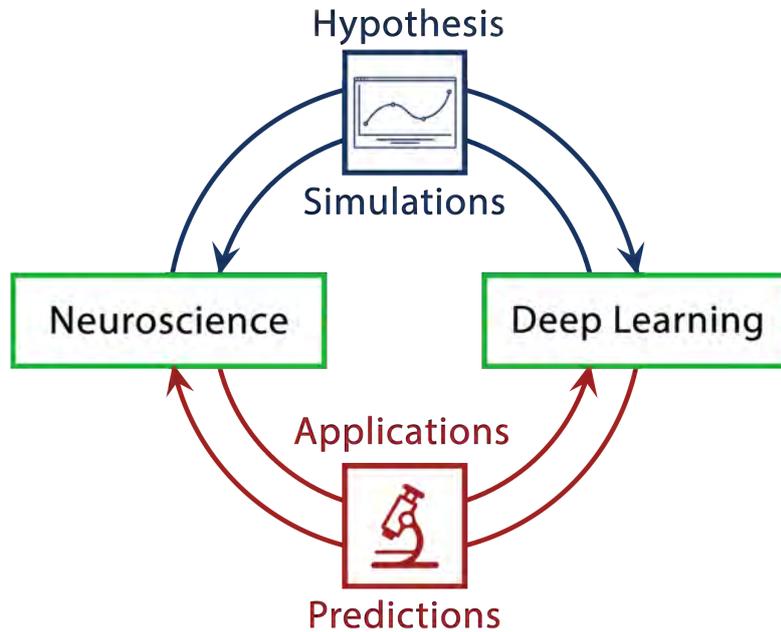


Figure 1.5 – **Synergies between Neuroscience and Deep learning.** The two fields interact in two distinct aspects: Neuroscience provides hypothesis and inspiration to build artificial neural networks that can in turn be criticized through deep learning experimentation. From an empirical perspective, Neuroscience data also represent a privileged field to deploy deep learning models.

development of recent generations of neural networks but also provides tools for analysing and comparing them (Schrimpf u. a., 2018; Geirhos u. a., 2020).

1.2.2 Reverse-engineering the brain

Moreover, various emerging features discovered in artificial networks suggest a deeper conceptual connection between artificial and biological networks: Recent discoveries in both representation and reinforcement learning (Botvinick u. a., 2020), such as the existence of hierarchically complex feature selectivity in vision networks (Yamins u. a., 2014; Khaligh-Razavi und Kriegeskorte, 2014; Güçlü und van Gerven, 2015) or the spatial selectivity profile of artificial neurons trained for navigation tasks (Banino u. a. (2018), see figure 1.6) are placing deep neural networks as an important modeling tool to understand their biological cousins. These synthetic neural models might shed light on specific phenomena that would

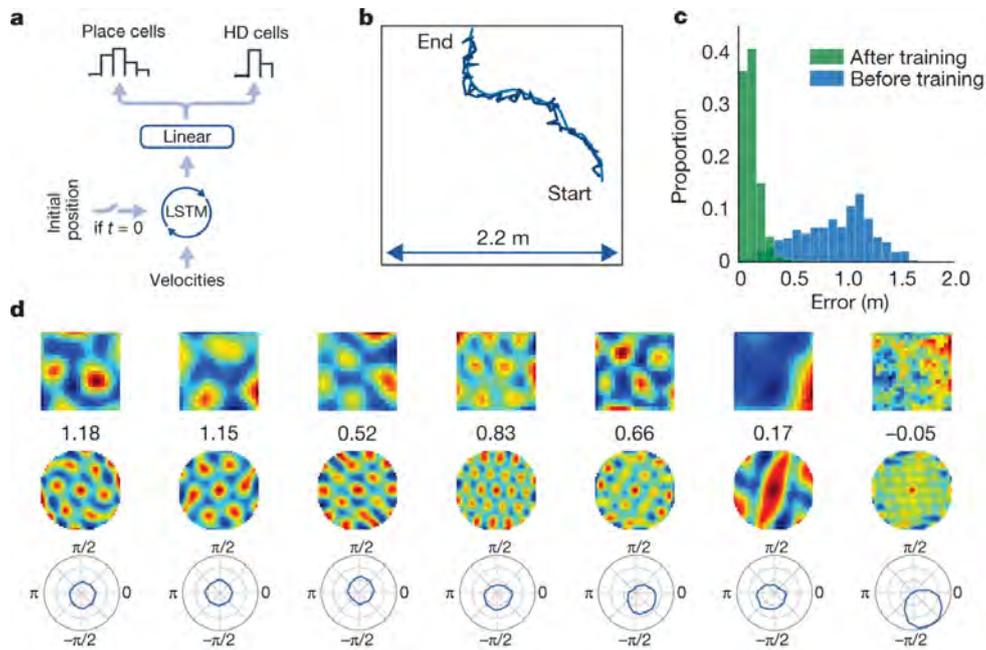


Figure 1.6 – **Example of grid-like representations similar to mammalian-brain emerging in artificial neural agents** a) A linear read-out of recurrent neural network trained for path-integration reveals activation profiles similar to those found in mammalian entorhinal cortex. b) An example of the path-integration task. c) Accuracy of decoded location before (blue) and after (green) training. d) Linear layer units exhibit spatially tuned responses resembling grid, border, and head direction cells. Top, ratemap shows activity over location; middle, spatial autocorrelogram of ratemap with gridness indicated; bottom, polar plot show activity versus head direction. Adapted from (Banino u. a., 2018)

otherwise be experimentally impossible to probe.

Research question 2

How can the emerging dynamical properties of trained artificial neural networks help understand the learning mechanisms of biological neural networks?

Hence, this duality currently strengthen both fields through a scientific loop: neuroinspiration accelerates research in deep learning which in turn help test the hypothesis from neuroscience (Storrs und Kriegeskorte, 2019). But it is also functioning on more experimental level: As neural-based machine intelligence becomes more an more precise, experimental

neuroscience becomes a privileged field for the application of modern artificial networks (Mathis und Mathis, 2020).

1.2.3 Missing pieces in artificial neural learning

Finally, although rapidly growing in classes, current artificial neural implementations are rough approximation of their biological counterparts and remains highly restricted with respect to the definition on how neurons interacts with each other. Arguably, it can be hypothesized that precious computational mechanisms entailing high representational and decisional capabilities might be lost through this blunt simplification. Layers of artificial neurons are highly parallel systems connected with static parametric pathways or *synapses* describing their interactions which map to operations from linear algebra and do not account for phenomena such as oscillations (VanRullen, 2016), or neuroplasticity and remapping (Wittenberg, 2010; Costandi, 2016). While universal approximation theorems exist under such constructive hypothesis, they only hold in some asymptotic sense and do not fully account for the current observed behaviors of such systems in terms of training and *in situ* dynamics in terms of robustness and generalization. Hence, augmenting current networks with bio-inspired computational mechanisms represents an active line of research with strong potential for understanding and improving artificial neural networks abilities.

1.3 Relation to control theory

1.3.1 Conceptual convergence

This work also advocates for a reinvestment of control theoretical approaches into deep learning and computational neuroscience research: As these two fields progress towards more and more advanced descriptions of the mechanisms supporting intelligence, there is no doubt that a science providing a principled framework to describe the interaction of actions and effects in complex dynamical systems will be of ever greater value for the two fields. Going further, if we accept a basic definition of intelligence as the capacity of an agent to identify and react to environmental novelty, we must agree that intelligence carries intrinsically a notion of *variation*, that is, of change in reaction to some cause. Hence, control theory, as a proposition to 1) generically model sensitivity and effects between variables of a system and 2) build autonomous processes that perpetually adapt to these, shares intrinsically common objectives

with Artificial Intelligence. While this conceptual contiguity was initially evident especially regarding the field of reinforcement learning and dynamic programming (Bellman, 1966), control and machine learning have evolved towards different approaches for studying complex autonomous systems that we can roughly interpret as a varied affinity between interpretable model-based solutions and model-free solutions built from available data. In recent years, as problems dimensionality increased and analytical solution were not available, methods from machine learning has taken over control theoretical solutions for applications were the available numerical schemes and control regulators were not sufficient (Han u. a., 2018; Degraeve u. a., 2022).

1.3.2 Control theoretical origin of deep learning

Hence, while leveraging neural networks as controllers has been an active area of scientific investigation, the reciprocal perspective about how control theory can inform deep learning systems has only been recently reconsidered. Beyond their converging objectives, control and deep learning share important theoretical descriptions that can shed a new light on modern artificial neural techniques. For instance, fundamental concepts that powers neural networks training such as backpropagation of errors (Rumelhart und McClelland, 1987) are actually derived from the Pontryagin Maximum Principle (Pontryagin Lev Semyonovich ; Boltyanskii V G und F, 1962). The powerful stochastic gradient descent scheme powering deep neural networks training have only recently received an extensive treatment from the control perspective (Liu und Theodorou, 2019; Li u. a., 2019). Moreover, the interpretation of deep residual networks as discrete dynamical systems (Chang u. a., 2017; Haber und Ruthotto, 2017; Ruthotto und Haber, 2018; Chang u. a., 2018; Ruthotto u. a., 2019) has allowed to introduce new optimization methods such as in Li u. a. (2017); Chen u. a. (2018); Böttcher u. a. (2022). Beyond optimization, control theory still heavily influences recent developments in reinforcement learning such as the description of optimal exploration-exploitation trade-off (Wang u. a., 2018a). This reconsidered interaction of model-based control theory with data-driven deep learning is still very recent. We can anticipate that more research at this control-learning interface could yield important theoretical and practical development for Artificial Intelligence at large. One contemporary central problem laying ahead consists in building a satisfactory system theory for neural networks. Current theoretical results considers specific networks classes but a systematic descriptions of how theses classes vary as a function of their

architectural choices and training might unveil generic motifs and shed light on the inner working of these currently mystifying objects.

1.3.3 Gradient descent as neural energy control

Virtually all recent approaches discussed in section 1.1.4 consist in controlling simultaneously thousands to billions parameters in large networks with a single approach called *gradient descent*. Formally attributed to Cauchy in his *Compte Rendu à l'Académie des Sciences* in 1847, gradient descent is fundamentally an iterative method that build a sequence of parameters $(\boldsymbol{\theta}_n)_{n \in \mathbb{N}}$ converging to some “stable” configuration in the parameter space Θ . One interpretation of defining a specific energy function over the network state as discussed in section 1.1.2 is that it creates a gradient field over the parameter space Θ allowing to reach specific parametric states in this space with respect to the energy E , that we discuss below.

Gradient descent

Let Θ be a finite-dimensional Hilbert space (most often, the network parameter space correspond to \mathbb{R}^d for $d \in \mathbb{N}$ a natural number) and a sequence $(\boldsymbol{\theta}_n)_{n \in \mathbb{N}}$ of elements in Θ . In this case, weak and strong convergence of $(\boldsymbol{\theta}_n)_{n \in \mathbb{N}}$ towards a point $\hat{\boldsymbol{\theta}} \in \Theta$ for the norm induced by the scalar product on Θ are equivalent and corresponds to the following property:

$$\lim_{n \rightarrow +\infty} \|\boldsymbol{\theta}_n - \hat{\boldsymbol{\theta}}\|_{\Theta} = 0 \quad (1.12)$$

We define the notion of differentiability that will be of interest for our discussion: A function E from Θ to a normed vector space \mathcal{V} is said to be Fréchet differentiable if there exists a bounded linear operator $F : \Theta \mapsto \Theta$ such that

$$\forall \boldsymbol{\theta} \in \Theta, \quad \lim_{\|\mathbf{h}\| \rightarrow 0} \frac{\|E(\boldsymbol{\theta} + \mathbf{h}) - E(\boldsymbol{\theta}) - F(\boldsymbol{\theta})\|}{\|\mathbf{h}\|} = 0 \quad (1.13)$$

We usually note this operator $\nabla_{\boldsymbol{\theta}} E$ and called gradient, which represent the direction of highest variation of the function E with respect to $\boldsymbol{\theta}$. If we now assume that an energy $E : \hat{\boldsymbol{\theta}} \mapsto \mathbb{R}$ is Fréchet differentiable such that $\nabla_{\boldsymbol{\theta}} E$ is ν -Lipschitz continuous:

$$\forall (\boldsymbol{\theta}, \boldsymbol{\theta}') \in \Theta^2, \quad \|\nabla_{\boldsymbol{\theta}} E(\boldsymbol{\theta}) - \nabla_{\boldsymbol{\theta}} E(\boldsymbol{\theta}')\|_{\Theta} \leq \nu \|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_{\Theta} \quad (1.14)$$

then we can define the gradient descent operator $Id - \gamma \nabla E : \Theta \mapsto \Theta$ which is a $\frac{\gamma\nu}{2}$ -averaged operator (?), such that for any sequence $(\lambda_n)_{n \in \mathbb{N}}$ such that $\sum \lambda_n (1 - \frac{\gamma\nu}{2} \lambda_n) > +\infty$, the sequence $(\theta_n)_{n \in \mathbb{N}}$ defined by

$$\theta_{n+1} = \theta_n - \lambda_n \gamma \nabla E(\theta_n), \quad \theta_0 \in \Theta \quad (1.15)$$

converges to a fixed point θ^* of $Id - \gamma \nabla E$, at rate $o(\frac{1}{n})$. Furthermore θ^* verifies $\nabla E(\theta^*) = 0$, a necessary condition for θ^* to be at least a local optimum of the function E .

This fixed-point perspective allows to see gradient descent as straightforward network control method that allow to reach useful attractor states in the parameter space Θ . (See (Bauschke u. a.) for a more complete description). However, a direct limitation of this numerical framework is the *generalization error* entailed by such formulation. Since the optimization problem defined in equation (1.9) is only defined over a finite set I of atoms, there is no explicit control over the entire support of the target data distribution. One central question of artificial network control is to understand the precise mechanism by which different hypothesis over network dynamics (architecture, optimization...) yield regularities with respect to the energy function in "unseen" cases. We illustrate this question by showing how the energy function of a neural network in a simple supervised learning problem evolves over the whole domain during gradient-based optimization in figure 1.9.

Moreover, this method can be compute and memory expansive since the energy can be defined on a large quantity of network states which are themselves high-dimensional vectors. A large body of work has attached to accelerate convergence in gradient fields, from preconditionning (sophistication of the step-size γ), averaging (adding momentum to the dynamics) or line-search (projection of the gradient update to specific subspaces). These techniques are out of the scope of this discussion but we must note that they all influence the dynamics of gradient descent and the properties of the found local minimum (see (Sun, 2019b) for a review).

One particular technique that is however worth mentioning is *stochastic gradient descent* (Robbins und Monro, 1951), which replace the deterministic update $\nabla E(\theta_n)$ in the gradient descent sequence with an unbiased estimator of the gradient $\widehat{\nabla E(\theta_n)}$. Apart from the computational efficiency of this method, which relaxes the need for computing exactly the gradient of the network energetic state, research has shown that it could benefit the generalization properties of the solutions (Keskar u. a.,

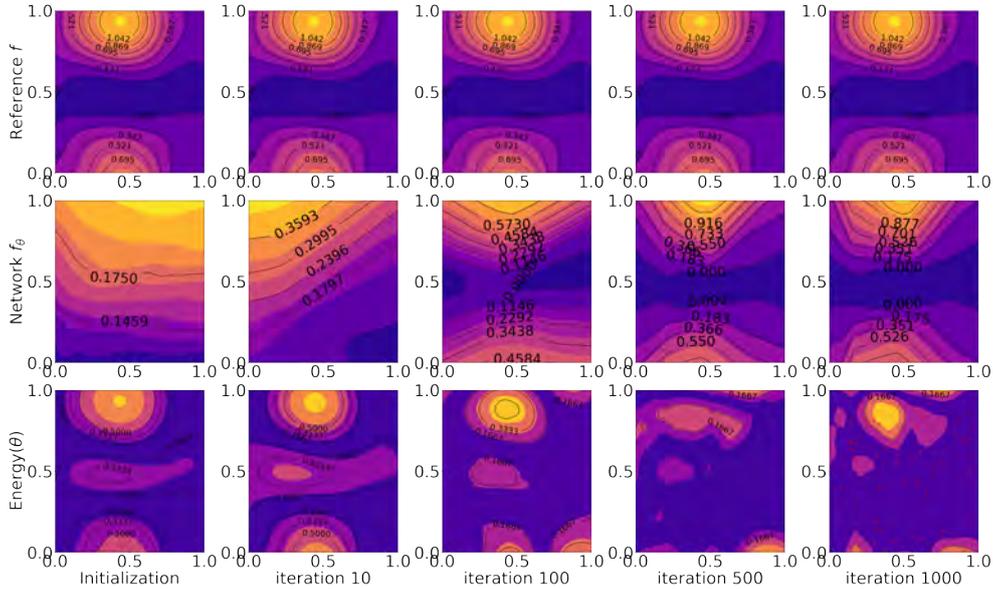


Figure 1.7 – **Evolution of a neural network energy profile over training represented by contour plots** Each row represent a distinct function: The upper row is the target function $f : [0, 1] \mapsto \mathbb{R}$, the second represents the network association between input and output neurons, the last one represent the energy $E(\theta)$ described in equation 1.11. As training progress from left to right, the network expressed function coincide with target and the energy decreases. Note however that the energy at points outside of the training set (represented by red crosses in last contour plot) is not controlled by definition.

2016; Pesme u. a., 2021) by influencing the type of reached energetic minimum in the parameter space. As an example, we show in figure 1.8 an illustration of the influence of the estimator size used to compute $\widehat{\nabla E(\theta_n)}$ over the network parameter trajectory under energy gradient control.

1.3.4 Artificial networks as functionals

Finally, control theory holds also potential for reconciling ambivalent views regarding neurons themselves, that is, they are either *functionals* whose state modulates information processing in downstream areas, or whether they are vectors collections of neurons are encoding simultaneously some information content in the form of a multi-dimensional vectors, a notion denoted as *population coding* (Panzeri u. a., 2010). This ambivalence is important because neuroscience theories constantly alter-

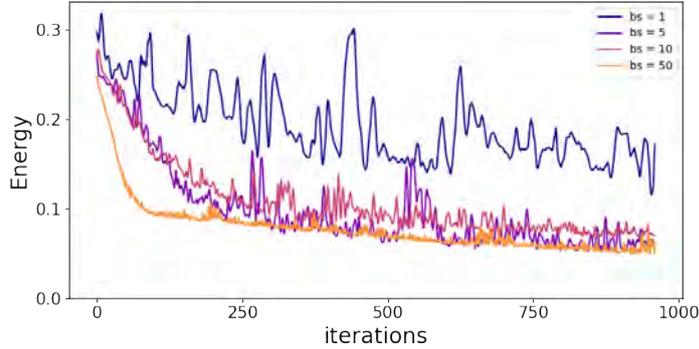


Figure 1.8 – **Evolution of the same network energy of the previous experiment as a function of number of gradient steps for different batch size** Each color represent a distinct batch size for estimating the energy. Note that this is not the training loss function but also measure the network alignment at unseen coordinates. Fewer example points determine a larger fluctuation of the energy during training.

nate between these dual conceptions and the same goes unsurprisingly for artificial neural networks: Researchers substitute often the perspective of neurons as filtering functions describing collectively a particular association from one space to another, and neurons activations as *embeddings* constituted by the collective activation of some specific population of neurons. From the energy perspective, this alternatives consist informally in considering the “marginalized” energy function with respect to specific state $\mathbf{x}_s^{\mathcal{S}}$ of a subset \mathcal{S} of neurons at time s :

$$E(\mathbf{x}_t, \dots, \mathbf{x}_s^{\mathcal{S}}, \dots, \mathbf{x}_{t'}, \boldsymbol{\theta}) = E_{\mathbf{x}_s^{\mathcal{S}}}(\mathbf{x}_t, \dots, \mathbf{x}_{t'}, \boldsymbol{\theta}) \quad (1.16)$$

Reconciling these two views by showing how they are two face of the same neural process is one central challenge for neuroscience and deep learning. In this work, we will adopt a different angle and argue that neuron activations are only to be taken as “encoding” the presence of some features in the data in the sense that they induce global transformations of the neural network expressed functions by affecting the entire chain of subsequent neural excitation/inhibitions. This view will also be extended by exploring new forms of such self-control, notably through principles of plasticity and meta-parameterization.

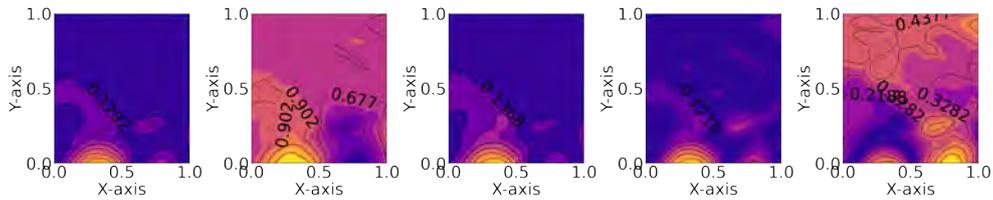


Figure 1.9 – **Influence of perturbing a single neuron state in the network over the energy function** We define a feedforward neural network with two hidden layers of width $w = 32$ and Rectified linear unit activation function (Fukushima, 1980) trained on the same target function as in figure 1.9. After training, we clamp at random one neuron of the last layer $x = 1$ and show the network energy map.

Research question 3

Can we build numerical methods to automatically discover the computational principles that determine learning in artificial networks? Are there efficient and robust alternatives to gradient descent control?

1.4 Challenges and stakes

1.4.1 Societal impact of deep learning

Because of its ubiquitous modeling abilities and high adjustability to constraints and hypothesis, deep learning powers an information processing revolution that currently percolates throughout science and industry. In natural sciences, it constitutes an original approach for modeling complex physical phenomena: On the one hand, the underspecified and automatic nature of neural learning potentially allows for finding original solutions that will complement and go beyond man-made approaches. On the other hand, the efficiency of network computation allows for new numerical simulation and computing methods that allows to tackle larger problems and iterate faster. In social sciences and humanities, deep learning offers altogether a new medium for experimentation. For instance, current advances in language and graph modeling allow to capture and interrogate collective or implicit phenomena in unprecedented ways (Mikolov u. a., 2013b; Qiu u. a., 2018; Fan u. a., 2019). In parallel, the originality and sophistication allowed by the integration of artificial neural productions in artistic creations is progressively calling into question what were taken as human prerogative skills, such as intuition and creativity. Altogether, deep learning transformative features have

the potential to change radically our lives at many levels ranging from mundane activities (transportation (Nguyen u. a., 2018), industry automation (Kotsiopoulos u. a., 2021), material and structure design (Zhou u. a., 2019), healthcare (Esteva u. a., 2019)) to more elaborated cultural and social interactions with new forms of production, communication and consumption of information and entertainment (Tsalakanidou, 2020; Sousa u. a., 2022)).

1.4.2 Current limitations of artificial networks

Yet for all these potential applications, the cognitive abilities exhibited the current generation of artificial neural networks remains highly primitive and brittle compared to the versatility and flexibility found in biological neural intelligence. Arguably, there are two aspects that can account for these systems limitations. First, the "genotype" of these systems, which correspond to the set of high-level architectural and computational principles governing information traffic between neurons is limited. As stated previously, many biological principles such as neural synchronicity, plasticity or neuromodulation remains under-explored in artificial deep networks. Similarly, architectural principles such as hierarchy, feedback or the diversity of biological neurons populations seems currently overlooked. Second, the training curriculum is also stereotypical and presumably jointly contributes to an artificial neural "phenotype" with well-known limitations that we will discuss over the following chapters: They suffer from a high sample complexity for learning (Xu u. a.) and show poor robustness to out-of-training-distribution cases and adversarial attacks (Szegedy u. a., 2013), or anomalous data contamination (Zhang u. a., 2021). In addition they are plagued with "catastrophic forgetting" of previous solution when retrained (McCloskey und Cohen, 1989) and have poor transfer abilities (Jin u. a., 2020). A question that this work is raising is whether these limitations can be put down to the current rudimentary control framework used to train them. By adopting the energy perspective, we can strive to build general behaviors and escape from the narrow train/test machine learning problem to build more adaptive and general neural agents, by binding together different aspect of neural computation, memory, learning into the language of optimality.

1.5 Contributions and Thesis organization

Below are listed the notable contributions of this thesis. Specifically, we split our discussion in two distinct parts. The first part explore the

general problem of learning functionals with neural networks in several contexts such as generative modeling and operator regression, while the second part is particularly dedicated to the notion of meta-learning neural network control rules.

- **Part I** presents two original approaches for learning of functionals with neural network which correspond to two different forms of neural network control: *parametric* and *data-based* control.
 - **Chapter 2** deals with a generalization of the class of time-continuous neural networks (Neural Ordinary Differential Equations) with dynamic parametrization associated to classical notions of open-loop and closed-loop control.
 - **Chapter 3** proposes an interpretation of the "in-context" learning abilities of recent Transformers-like architecture from the perspective of vector-valued Reproducing Kernel Banach Space theory which leads to the construction of a meta-learning framework based on original transduction principles to perform ultra-fast finite or infinite-dimensional functional regression.
- **Part II** particularly focuses on a specific form of functional learning which consists in the problem of learning-to-learn or meta-learning.
 - In **Chapter 4**, we propose a review of recent meta-learning approach organised in relation to the notion of free-energy minimization as a general principle for learning in natural and artificial intelligence. We further outline different potential research direction and challenges for neural meta-learning system that will be discussed in the next chapter.
 - **Chapter 5** presents an original class of neural network with tunable fast Hebbian synaptic plasticity inspired from neuroscience that describe a form of dynamic modern Hopfield networks. We show that such model can implement a self-contained Reinforcement Learning program through its weights dynamics with application to one-shot learning, spatial navigation and robotic motor control.

- Finally, **Chapter 6** concludes our discussion and outlines potential future continuation of this work.

Part I

Learning higher-order functions

Introduction

In essence, computer science is striving towards building and understanding automatic machines able to manipulate ever higher-level abstract concepts as well as being endowed with more and more autonomous behavior. In recent years, this aspiration has materialized into a clear conceptual shift going from human-specified algorithm design towards computational solutions automatically unfolding from loosely specified principles. Machine learning has been at the forefront of this transformation by building loosely specified data-centric approaches culminating in the unmatched modeling abilities characterizing *deep learning* (LeCun u. a., 2015). This algorithmic shift from hand-designed to learned features characterizing modern deep learning has been transformative, allowing to solve complex problems ranging from video games (Mnih u. a., 2013, 2015) to multiplayer contests (Jaderberg u. a., 2019) or motor control (Levine u. a., 2016; Lillicrap u. a., 2016). Yet while deep learning has shown, as previously discussed, astonishing results for machine intelligence, current generations of neural networks operate at fixed time-scales and low abstraction level lacking compositionality to form symbolic structures which presumably determines the high sample complexity required for training as well as unintuitive decision rules. Furthermore, they remains weakly adaptive to new contexts with static properties over their lifetime deployment, depending on specific human intervention for tuning to specific applications. Recent work has shown that deep neural networks could learn higher-order functions (i.e functions of functions) (Lu u. a., 2019; Li u. a., 2020a), showing potential to form algorithmic and computational principles that will lead artificial neural networks towards representing and manipulating more sophisticated and conceptual representations of information that remain currently a hallmark of biological networks. In this regard, this first part explores two particular models of neural network as functionals, one is a generalization of the recent class of continuous-time networks called *Neural ODEs*, the second propose a reinterpretation of recent attentional models as function approximators in particular Banach spaces of functions.

Chapter 2

Neurally controlled ordinary differential equations

“Among all of the mathematical disciplines the theory of differential equations is the most important... It furnishes the explanation of all those elementary manifestations of nature which involve time.”

Sophus Lie

Contents

2.1	Introduction	45
2.2	Neural ODEs	46
2.3	Neurally Controlled Ordinary Differential Equation	48
2.4	Experimental results	57
2.5	Controlled Normalizing flows	65
2.6	Discussion	67

2.1 Introduction

This chapter covers the presentation of a theoretical generalization of the recently proposed *Neural ODE* (Chen u. a., 2018) framework, which build on a close connection between residual networks and the theory of ordinary differential equations. We show that such models can be extended to form open and closed-loop control systems where networks parameters constitute the control variables. In turns, these augmented systems endow *Neural ODEs* with increased representation capacity and can serve different purpose from generative modeling with *normalizing*

flows, complex dynamics stabilization or few-shot adaptation with associative memory networks.

2.2 Neural ODEs

2.2.1 Background

The interpretation of artificial neural networks as discretizations of continuous-time ordinary differential equations has led to both theoretical and practical advances in representation learning (Lu u. a., 2018; Chang u. a., 2018). According to this interpretation, the separate layers of a deep neural network are understood to be a discretization of a continuous-time dynamical system. *Neural ODEs* (NODEs) adopt this perspective by directly considering neural non-linear functions as vector fields that can be integrated over time using classic solver methods, so that, in the limit, an infinitely deep network is defined. Because of their ability to take advantage of such black-box differential equation solvers and their correspondence to dynamical systems in nature, NODEs have found natural applications from physical system identification (Köhler u. a., 2019; Ruthotto u. a., 2020) to generative modeling with free-form jacobian normalizing flows (Grathwohl u. a., 2019) such as the recent and powerful class of diffusion models (Song u. a., 2021). They can be generically described as dynamical systems of the following form:

$$\frac{d\mathbf{x}}{dt} = f(\mathbf{x}, \boldsymbol{\theta}, t), \quad (2.1)$$

where \mathcal{X} is a space of features (such as the vector space \mathbb{R}^N), $\boldsymbol{\theta} \in \Theta$ is a collection of learnable parameters, and $f : \mathcal{X} \times \Theta \times \mathbb{R} \mapsto \mathcal{X}$ is an equation of motion which we take to be differentiable on its whole domain, such as a neural network with differentiable activation function. Each f defines a *flow*, i.e. a triple $(\mathcal{X}, \mathbb{R}, \Phi_{\boldsymbol{\theta}})$ with $\Phi_{\boldsymbol{\theta}} : \mathcal{X} \times \mathbb{R} \mapsto \mathcal{X}$ defined by:

$$\Phi_{\boldsymbol{\theta}}(\mathbf{x}(0), T) = \mathbf{x}(0) + \int_0^T f(\mathbf{x}(t), \boldsymbol{\theta}, t) dt \quad (2.2)$$

which relates an initial point $\mathbf{x}(0)$ to an orbit of points $\{\mathbf{x}(t) = \Phi_{\boldsymbol{\theta}}(\mathbf{x}(0), t), t \in \mathbb{R}\}$. For a fixed T , the map $x \mapsto \Phi_{\boldsymbol{\theta}}(x, T)$ is a homeomorphism from \mathcal{X} to itself that is parametrized by $\boldsymbol{\theta}$. Several properties of such flows make them appealing for machine learning. As mentioned, the ODEs that govern them can be solved with off-the shelf solvers and they can potentially model data irregularly sampled in time. Moreover, such flows

are reversible maps by construction whose inverse is just the system integrated backward in time, $\Phi_\theta(\cdot, t)^{-1} = \Phi_\theta(\cdot, -t)$. This property enables depth-constant memory cost of training thanks to the adjoint sensitivity method (Pontryagin Lev Semyonovich ; Boltyanskii V G und F, 1962) and the modeling of continuous-time generative normalizing flow algorithms (Grathwohl u. a., 2019). Interestingly, discretizing Eq. 2.2 yields the recursive formulation of a residual network (He u. a., 2015) with a single residual operator f_θ :

$$\Phi_{ResNet}(\mathbf{x}(0), T) = \mathbf{x}(0) + \sum_{t=1}^T f_\theta(\mathbf{x}_{t-1}) \quad (2.3)$$

In this sense, NODEs with time-independent (autonomous) equations of motion, f , are the infinitely-deep limit of weight-tied residual networks. Relying on the fact that every non-autonomous dynamical system with state $\mathbf{x} \in \mathbb{R}^d$ is equivalent to an autonomous system on the extended state $(\mathbf{x}, t) \in \mathbb{R}^{d+1}$, Eq. 2.2 can also be used to model general, weight-untied residual networks. However it remains unclear how dependence of f in time should be modeled in practice and how their dynamics relate to their discrete counterparts with weights evolving freely across blocks through gradient descent.

2.2.2 Limit of NODE to function approximation

Using NODEs for representation learning suffer from known limitations regarding the class of expressible functions. By definition, the feature transformation created by Neural ODEs are homeomorphic, that is, they are continuous, bijective, with inverse continuous. Hence they cannot change the topology of the feature space.

Researchers have tried to alleviate either by lifting the feature space into an higher dimensional one or by allowing the vector field to change in time, making the system non-autonomous (Dupont u. a., 2019). For example, Zhang u. a. (2020) showed that NODEs can arbitrarily approximate maps from \mathbb{R}^d to \mathbb{R} if NODE dynamics operate with an additional time dimension in \mathbb{R}^{d+1} and the system is affixed with an additional linear layer. The same authors showed that NODEs could approximate homeomorphisms from \mathbb{R}^d to itself if the dynamics were lifted to \mathbb{R}^{2d} . Yet, the set of homeomorphisms from \mathbb{R}^d to itself is in fact quite a conservative function space from the perspective of representation learning, since these mappings preserve topological invariants of the data space, preventing them from “disentangling” data classes like those of the annulus data that we present in section 2.3.3 (lower left panel). In general,

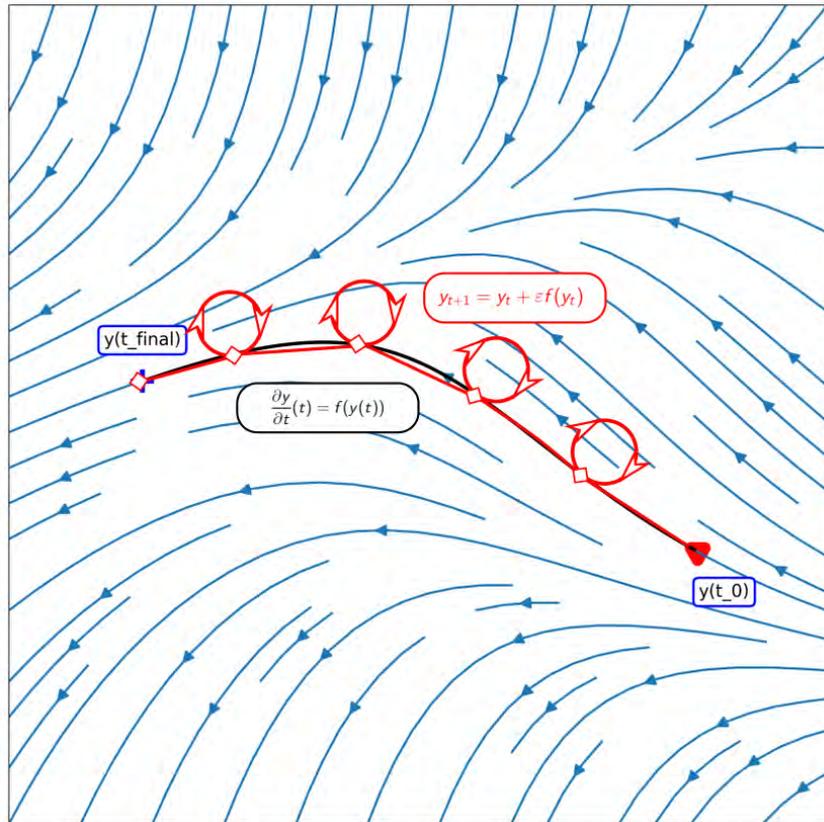


Figure 2.1 – Simple example of a vector field (in blue) parameterized by a one-layer perceptron acting on a 2-dimensional feature space. A particular *orbit* (in black) is approximated through an explicit Euler integration scheme with constant step-size (in red). The application $\Phi_{\theta}(\mathbf{y}(0), T)$ defines in turn a particular feature transformation that can be trained with gradient descent.

understanding the expressive capabilities of the continuous-time neural networks and how they relate to their deep discrete counterparts is an active area of research. Ott u. a. (2021)

2.3 Neurally Controlled Ordinary Differential Equation

2.3.1 Motivation

In this subsection, we present the generic formulation of Neurally Controlled Ordinary Differential Equations (NCODE). The main idea of NCODE is to consider the parameters, θ , in Eq. 2.1 as dynamic con-

trol variables, $\boldsymbol{\theta}(\mathbf{x}, t)$, for the dynamical state, $\mathbf{x}(t)$. This generalization is inspired by several observations. First, generic problems in optimal control theory (Liberzon, 2011), similar to the learning objective we will introduce in section 2.3.4, are formulated over dynamic variables $\boldsymbol{\theta}(\mathbf{x}, t)$ instead of static parameters $\boldsymbol{\theta}$, which allows to tackle more general control problems and most notably allows to build the notion of feedback, which plays a central role for learning. Second, this development echoes recent research regarding meta-parametrisation of deep networks through *hyper-networks* (Ha u. a., 2017) as well as self-referential networks (Irie u. a., 2022b). These ideas explore a blind spot of gradient-based deep learning, by singularly reconsidering networks connection parameters as dynamical process that are governed by trainable hierarchical mechanisms. Lastly, it also echoes a known principle from computational neuroscience, *short-term synaptic plasticity* which describes electro-chemical processes determining fast variations in synaptic efficacy between biological neurons, thus creating fluctuations in the transfer function expressed by biological networks at fast time scales (Wu u. a., 2013). We will see that NCODE provides a generic framework to embed and explore such ideas into continuous-time dynamics. We note also that recent developments proposed for Neural ODES such as delays (Zhu u. a., 2021), polymorphic parameterization (Yoon u. a., 2022), gradient flows (Masaroli u. a., 2020a) or Riemannian manifold parametrization (Choromanski u. a., 2020b) can be combined with our general discussion.

2.3.2 General formulation

In NCODE, model dynamics are governed by a coupled system of equations on the extended state $\mathbf{z}(t) = (\mathbf{x}(t), \boldsymbol{\theta}(t))$. Throughout, we assume that the initial time point is $t = 0$. The initial value of the control weights, $\boldsymbol{\theta}(0)$, is given by a mapping $\gamma : \mathcal{X} \rightarrow \Theta$. The full trajectory of control weights, $\boldsymbol{\theta}(t)$, is then output by a controller, g , given by another differentiable equation of motion $g : \Theta \times \mathcal{X} \times \mathbb{R} \mapsto \Theta$ with initial condition $\gamma(\mathbf{x}_0)$. Given an initial point, $\mathbf{x}(0)$, we can solve the initial value problem (IVP)

$$\begin{cases} \frac{d\mathbf{z}}{dt} = h(\mathbf{z}, t) \\ \mathbf{z}(0) = \mathbf{z}_0 \end{cases} = \begin{cases} \left(\frac{d\mathbf{x}}{dt}, \frac{d\boldsymbol{\theta}}{dt} \right) = (f(\mathbf{x}, \boldsymbol{\theta}, t), g(\boldsymbol{\theta}, \mathbf{x}, t)) \\ (\mathbf{x}(0), \boldsymbol{\theta}(0)) = (\mathbf{x}_0, \gamma(\mathbf{x}_0)) \end{cases} \quad (2.4)$$

where $h = (f, g)$. We may think of g and γ as a controller of the dynamical system with equation of motion, f . We model g and γ as neural

networks parameterized by $\boldsymbol{\mu} \in \mathbb{R}^{n_\mu}$ and use gradient descent techniques where the gradient can be computed by solving an adjoint problem (Pontryagin Lev Semyonovich ; Boltyanskii V G und F, 1962) that we describe in the next section. Our goal here is to use the meta-parameterization in the space Θ to create richer dynamic behavior for a given f than directly optimizing fixed weights $\boldsymbol{\theta}$.

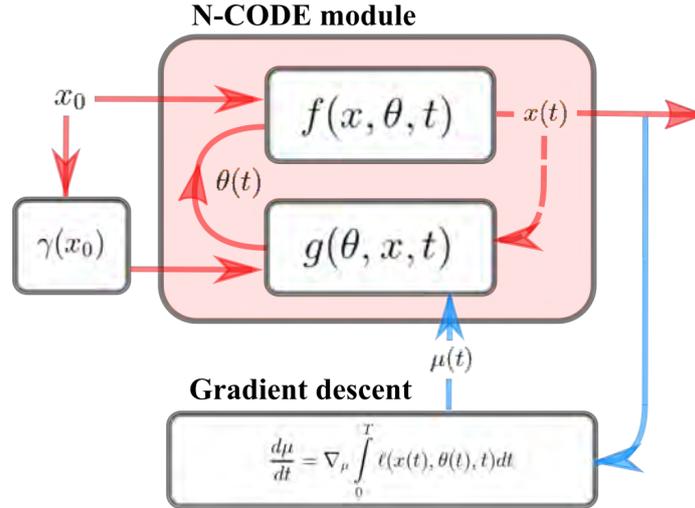


Figure 2.2 – Diagram of a general N-CODE module: Given a initial state $\boldsymbol{x}(0)$, the module consists of an augmented dynamical system that couples activity state \boldsymbol{x} and weights $\boldsymbol{\theta}$ over time (red arrows). A mapping γ infers initial control weights $\boldsymbol{\theta}_0$ defining an initial flow (open-loop control). This flow can potentially evolve in time as $\boldsymbol{\theta}$ might be driven by a feedback signal from \boldsymbol{x} (closed-loop, dotted line). This meta-parameterization of f can be trained with gradient descent by solving an augmented adjoint sensitivity system (blue arrows).

Well-posedness - If f and g are continuously differentiable with respect to \boldsymbol{x} and $\boldsymbol{\theta}$ and continuous with respect to t , then, for all initial conditions $(\boldsymbol{x}(0), \boldsymbol{\theta}(0))$, there exists a unique solution \boldsymbol{z} for Eq. 7.3 by the Cauchy-Lipschitz theorem. This result leads to the existence and uniqueness of the augmented flow $(\mathcal{X} \times \Theta, \mathbb{R}, \Phi_{\boldsymbol{\mu}})$ with $\Phi_{\boldsymbol{\mu}} : (\mathcal{X} \times \Theta) \times \mathbb{R} \mapsto \mathcal{X} \times \Theta$. Moreover, considering the restriction of such a flow on \mathcal{X} , we are now endowed with a universal approximator for at least the set of homeomorphisms on \mathcal{X} given that this restriction constitutes a non-autonomous system. We discuss below how g and γ affect the evolution of the variable $\boldsymbol{x}(t)$, exhibiting two forms of control and noting how they relate to previous extensions of NODEs.

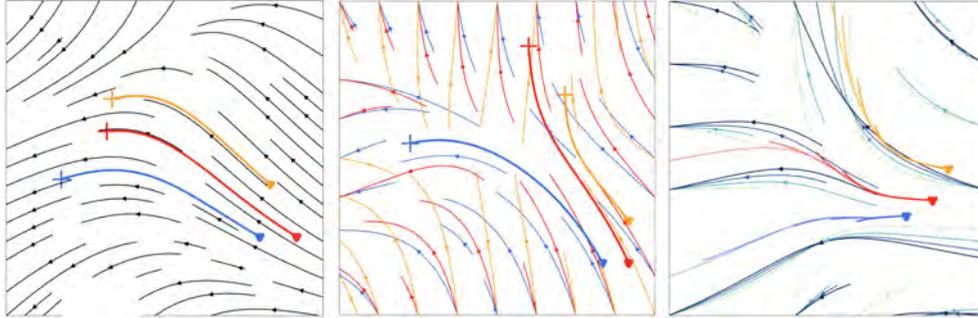


Figure 2.3 – Idealized dynamical computations of neural modules governed by ODEs and interpreted as particle trajectories in a vector field over time (from triangles to crosses.) Left: A single vector field is learnt for the whole data-distribution. Middle: An open-loop control enables the system to flexibly adjust the trajectory for every data point. Right: A closed-loop control continuously transform the form of vector field at every state of the trajectory, making it a non-autonomous system.

2.3.3 Open and closed-loop controllers

If the controller outputs control weights as a function of the current state, $\mathbf{x}(t)$, then we say it is a *closed-loop* controller. Otherwise, it is an *open-loop* controller.

Open-loop control: First, we consider the effect of using only the mapping γ in Eq. 7.3 as a controller. Here, γ maps the input space \mathcal{X} to Θ so that f is conditioned on $\mathbf{x}(0)$ but not necessarily on $\mathbf{x}(t)$ for $t > 0$. In other words, each initial value $\mathbf{x}(0)$ evolves according to its own learned flow $(\mathcal{X}, \mathbb{R}, \Phi_{\gamma(\mathbf{x}(0))})$. This allows for trajectories to evolve more freely than within a single flow that must account for the whole data distribution and resolves the problem of non-intersecting orbits (see Figure 2.4). Recently, (Massaroli u. a., 2020b) proposed a similar form of data-conditioned open-loop control with extended state $(\mathbf{x}(t), \mathbf{x}(0))$. This is a version of our method in which γ is of the form $\gamma(\mathbf{x}) = \theta(0) = [C : id]$ with C a constant vector, id is the identity function, and $:$ denotes concatenation. Our open-loop formulation makes an architectural distinction between controller and dynamics and is consequently generalizable to the following closed-loop formulation.

Closed-loop control: Defining a differentiable mapping, g , which outputs the time-dependent control weights $\theta(t)$ given the state of the variable $\mathbf{x}(t)$ yields a non-autonomous system on \mathcal{X} (see Fig. 2.12). This can be seen as a specific dimension augmentation technique where additional

variables correspond to the parameters $\boldsymbol{\theta}$. However, contrary to appending extra dimensions which does not change the autonomous property of the system, this augmentation results in a module describing a time-varying transformation $\Phi_{\boldsymbol{\theta}(t)}(\mathbf{x}_0, t)$. Note that this formulation generalizes the functional parameterization proposed in recent non-autonomous NODES systems (Choromanski u. a., 2020b; Massaroli u. a., 2020b; Zhang u. a., 2019), since the evolution of $\boldsymbol{\theta}(t)$ depends on $\mathbf{x}(t)$. Much like in the case of classical control theory, we hypothesized that the use of dynamic control weights would be of particular use in reacting to a non-stationary stimulus.

The expressivity of NCODE compared to other continuous-time neural networks is encapsulated in the following proposition. The result shows that both open-loop and closed-loop control systems overcome NODEs' expressivity constraint with two distinct strategies, data-conditioning and state-space augmentation.

Proposition 1 *There exists a function $\phi : \mathbb{R}^d \rightarrow \mathbb{R}^d$ which can be expressed by NCODE but not by NODEs. In particular, ϕ is not a homeomorphism.*

Proof of Proposition 1: In order to show that the set of functions defined by NCODE are not necessarily homeomorphisms on \mathcal{X} , we show that they are not generally injective.

Open-loop: Consider the one-dimensional system $\Phi : (\mathcal{X}, \mathbb{R}) \mapsto \mathcal{X}$ with equation of motion $f(\mathbf{x}, \boldsymbol{\theta}) = -\boldsymbol{\theta}$ where $\boldsymbol{\theta} = \gamma(\mathbf{x}(0)) = \mathbf{x}(0)$. This system has solution:

$$\Phi(\mathbf{x}(0), T) = \mathbf{x}(0) + \int f(\mathbf{x}, \boldsymbol{\theta}, t) dt = \mathbf{x}(0) - \mathbf{x}(0)T. \quad (2.5)$$

However, this implies $\Phi(0, 1) = \Phi(1, 1) = 0$.

Closed-loop: Similarly, we can show that the following 2D oscillatory system:

$$\begin{cases} \dot{\mathbf{x}}(t) = \boldsymbol{\theta}(t) \\ \dot{\boldsymbol{\theta}}(t) = -\mathbf{x}(t) \end{cases} \implies \mathbf{x}(t) = \alpha \cos(t) + \beta \sin(t) \quad (2.6)$$

If $\boldsymbol{\theta}(0) = 0$, we have that $\mathbf{x}(t) = \mathbf{x}(0) \cos(t)$, which implies that $\Phi(\mathbf{x}, \pi/2) = 0$, so $\mathbf{x} \mapsto \Phi(\mathbf{x}, t) = \mathbf{x}(t)$ is not injective.

In particular, the transformation of the input space is no longer constrained to be a homeomorphism, since the flows associated with each datum are specifically adapted to that point. Consequently, our system can easily “tear” apart the two annulus classes in Fig. 2.5 without directly lifting the data space to a higher dimension. Moreover, when control weights are allowed to vary in time, they can play the role of fast, plastic synapses which can adapt to dynamic model states and inputs.

Two examples : Reflection and Concentric Annuli

We illustrate our previous point with a 1-dimensional and a 2-dimensional problem previously discussed in (Dupont u. a., 2019; Massaroli u. a., 2020b), consisting of learning either the reflection map $\varphi(\mathbf{x}) = -\mathbf{x}$ or a linear classification boundary on a data-set of concentric annuli. Earlier results have shown that vanilla NODEs cannot learn these functions since NODEs preserve the topology of the data space, notably its linking number, leading to unstable and complex flows for entangled classes. However, we will show that both the open and closed loop formulations of our model easily fit these functions, as shown in Figs 2.4 and 2.5. The adaptive parameterization allows our model to learn separate vector fields for each input, allowing for simpler paths that ease model convergence. Informally, we may think of the classification decisions as being mediated by a bifurcation parameterized by the data space. Transiting across the true classification boundary switches the vector field from one pushing points to one side of the classification boundary to the other. Moreover, the construction of γ and g as a neural network guarantees a smooth variation of the flow with respect to the system state, which can potentially provide interesting regularization properties on the family learned by the model.

2.3.4 Training

In a standard control problem, dynamics are evaluated according to a generalized loss function that integrates a cost over some interval $[0, T]$:

$$l(\mathbf{z}) := \int_0^T \ell(\mathbf{z}(t), t) dt = \int_0^T \ell(\mathbf{x}(t), \boldsymbol{\theta}(t), t) dt \quad (2.7)$$

The loss in Eq. 2.7 is more general than in Chen u. a. (2018) since ℓ can be any Lebesgue-measurable function of both states, \mathbf{x} , and control parameters, $\boldsymbol{\theta}$. In particular, this includes penalties at discrete time points or over the entire trajectory (Massaroli u. a., 2020b) but also regularizations on the weights or activations over the entire trajectory rather than

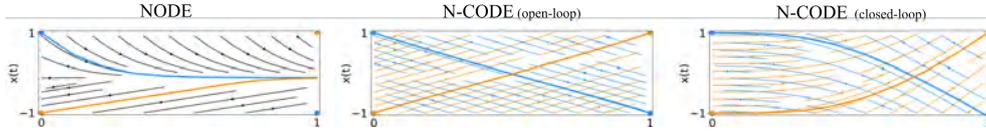


Figure 2.4 – Trajectories over time for three types of continuous-time neural networks learning the 1-dimensional reflection map $\varphi(\mathbf{x}) = -\mathbf{x}$. **Left:** (*NODE*) Irrespective of the form of f , a NODE module cannot learn such a function as trajectories cannot intersect (Dupont u. a., 2019). **Middle:** (*Open-loop NCODE*). The model is able to learn a family of vector fields by controlling a single parameter of f conditioned on $x(0)$. **Right:** (*Closed-loop NCODE*) With a fixed initialization, $x(0)$, the controller model still learns a deformation of the vector field to learn φ . Note that the vector field is time-varying in this case, contrary to the two other versions.

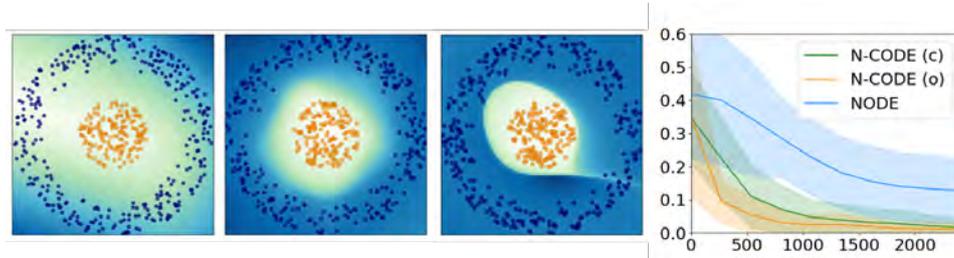


Figure 2.5 – The *concentric annuli* problem. A classifier must separate two classes in \mathbb{R}^2 , a central disk and an annulus that encircles it. **Left:** Soft decision boundaries (blue to white) for NODE (*First*), NCODE open-loop (*Second*) and closed-loop (*Third*) models. **Right:** Training curve for the three models as a function of gradient steps

the final state $\mathbf{z}(T)$ of the system. In order to estimate a control function $\boldsymbol{\theta}(t)$ that is optimal with respect to Eq. 2.7, we invoke Pontryagin’s maximum principle (PMP) (Pontryagin Lev Semyonovich ; Boltyanskii V G und F, 1962), which only requires mild assumptions on the functional control space Θ and applies to functions f that are non-smooth in $\boldsymbol{\theta}$. The PMP gives necessary conditions on $\boldsymbol{\theta}(t)$ at optimality via the augmented adjoint variable $\mathbf{a}(t)$. This quantity is the Jacobian of ℓ with respect to both $\mathbf{x}(t)$ and $\boldsymbol{\theta}(t)$. In the case of $\boldsymbol{\theta}(t)$ being differentiable with respect to the meta-parameters $\boldsymbol{\mu}$, solving for the augmented adjoint state $\mathbf{a}(t)$ as in Chen u. a. (2018) allows us to compute the gradient of the loss with respect to $\boldsymbol{\mu}$ thanks tthe following theorem.

Theorem 1 *Augmented adjoint method:* Given the IVP of (7.3) and for ℓ defined in (2.7), we have:

$$\frac{\partial \ell}{\partial \boldsymbol{\mu}} = \int_0^T \mathbf{a}(t)^T \frac{\partial h}{\partial \boldsymbol{\mu}} dt, \quad \text{such that } \mathbf{a} \text{ satisfies } \begin{cases} \frac{d\mathbf{a}}{dt} = -\mathbf{a}^T \cdot \frac{\partial h}{\partial \mathbf{z}} - \frac{\partial \ell}{\partial \mathbf{z}} \\ \mathbf{a}(T) = \mathbf{0} \end{cases} \quad (2.8)$$

where $\frac{\partial h}{\partial \mathbf{z}}$ is the Jacobian of h with respect to \mathbf{z} :
$$\frac{\partial h}{\partial \mathbf{z}} = \begin{pmatrix} \frac{\partial f}{\partial \mathbf{x}} & \frac{\partial f}{\partial \boldsymbol{\theta}} \\ \frac{\partial g}{\partial \mathbf{x}} & \frac{\partial g}{\partial \boldsymbol{\theta}} \end{pmatrix}.$$

Proof of Theorem 1 We place our analysis in the space \mathbb{R}^n . Since ℓ and h are continuously differentiable, let us form the *Lagrangian* functional \mathcal{L} with \mathbf{a} the lagrange multiplier, which is an element of the dual space of $\mathbf{z} = \begin{bmatrix} \mathbf{x} \\ \boldsymbol{\theta} \end{bmatrix}$:

$$\mathcal{L}(\mathbf{z}, \mathbf{a}, T) := \int_0^T [\ell - \langle \mathbf{a}, \frac{\partial \mathbf{z}}{\partial t} - h \rangle] dt \quad (2.9)$$

The constraint $\langle \mathbf{a}, \frac{\partial \mathbf{z}}{\partial t} - h \rangle$ is always *active* on the admissible set by construction of \mathbf{z} in (7.3), such that we have $\forall \mathbf{z}, \mathbf{a}, T$ $\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}}(\mathbf{z}, \mathbf{a}, T) = \frac{\partial \ell}{\partial \boldsymbol{\mu}}(\mathbf{z}, \mathbf{a}, T)$. Moreover, integrating the left part of the integral in (2.9) gives:

$$\int_0^T \langle \mathbf{a}, \frac{\partial \mathbf{z}}{\partial t} \rangle dt = \langle \mathbf{a}, \mathbf{z} \rangle \Big|_0^T - \int_0^T \langle \frac{\partial \mathbf{a}}{\partial t}, \mathbf{z} \rangle dt \quad (2.10)$$

Now, given that $\ell - \langle \frac{d\mathbf{a}}{dt}, \mathbf{z} \rangle$ is differentiable in $\boldsymbol{\mu}$ for any $t \in [0, T]$, Leibniz integral rule allows to write

$$\frac{\partial l}{\partial \boldsymbol{\mu}} = \int_0^T \frac{\partial}{\partial \boldsymbol{\mu}} [\ell + \langle \frac{d\mathbf{a}}{dt}, \mathbf{z} \rangle + \langle \mathbf{a}, h \rangle] dt + \langle \mathbf{a}(T), \frac{\partial \mathbf{z}(T)}{\partial \boldsymbol{\mu}} \rangle - \langle \mathbf{a}(0), \frac{\partial \mathbf{z}(0)}{\partial \boldsymbol{\mu}} \rangle \quad (2.11)$$

$$\frac{\partial l}{\partial \boldsymbol{\mu}} = \int_0^T \left[\frac{\partial \ell}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \boldsymbol{\mu}} + \langle \frac{d\mathbf{a}}{dt}, \frac{\partial \mathbf{z}}{\partial \boldsymbol{\mu}} \rangle + \langle \mathbf{a}, \frac{\partial h}{\partial \boldsymbol{\mu}} \rangle \right] dt + \langle \mathbf{a}(T), \frac{\partial \mathbf{z}(T)}{\partial \boldsymbol{\mu}} \rangle \quad (2.12)$$

$$\frac{\partial l}{\partial \boldsymbol{\mu}} = \int_0^T \left[\frac{\partial \ell}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \boldsymbol{\mu}} + \langle \frac{d\mathbf{a}}{dt}, \frac{\partial \mathbf{z}}{\partial \boldsymbol{\mu}} \rangle + \langle \mathbf{a}, \frac{\partial h}{\partial \boldsymbol{\mu}} + \frac{\partial h}{\partial t} \frac{\partial t}{\partial \boldsymbol{\mu}} + \frac{\partial h}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \boldsymbol{\mu}} \right] dt + \langle \mathbf{a}(T), \frac{\partial \mathbf{z}(T)}{\partial \boldsymbol{\mu}} \rangle \quad (2.13)$$

The last equation can be reordered as:

$$\frac{\partial l}{\partial \boldsymbol{\mu}} = \int_0^T \left\langle \frac{d\mathbf{a}}{dt} - \mathbf{a} \frac{\partial h}{\partial \mathbf{z}} - \frac{\partial \ell}{\partial \mathbf{z}}, \frac{\partial \mathbf{z}}{\partial \boldsymbol{\mu}} \right\rangle dt + \int_0^T \langle \mathbf{a}, \frac{\partial h}{\partial \boldsymbol{\mu}} \rangle dt + \langle \mathbf{a}(T), \frac{\partial \mathbf{z}(T)}{\partial \boldsymbol{\mu}} \rangle \quad (2.14)$$

Posing that $\mathbf{a}(T) = \mathbb{O}_{|\mathcal{X}|+|\Theta|}$ and $\langle \frac{d\mathbf{a}}{dt} - \mathbf{a} \frac{\partial h}{\partial \mathbf{z}} - \frac{\partial \ell}{\partial \mathbf{z}}, \frac{\partial \mathbf{z}}{\partial \boldsymbol{\mu}} \rangle = 0$, the result follows.

Remark 1: This result allows to compute gradient estimate assuming that $\boldsymbol{\mu}$ directly parametrizes h such that the gradient $\partial h / \partial \boldsymbol{\mu}$ are straightforward to compute. However, in the case of optimizing the initial mapping γ that infer the variables $\boldsymbol{\theta}$, this result can be combined with usual chain rule to estimate the gradient:

$$\frac{\partial l}{\partial \boldsymbol{\mu}_\gamma} = \frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\gamma}} \frac{\partial l}{\partial \boldsymbol{\theta}} \quad (2.15)$$

In practice, we compute the Jacobian for this augmented dynamics with open source automatic differentiation libraries using Pytorch (Paszke u. a., 2019), enabling seamless integration of NCODE modules in bigger architectures. Furthermore, our implementation is based on the remarkable `torchdiffeq` package that augment Pytorch with NODE adjoint sensitivity training method, we provide a commented generic PyTorch implementation for the NCODE module in the open-loop setting in Annex and show examples of such modules in section 2.4 below.

2.4 Experimental results

Resources - Our experiments were run on a 12GB NVIDIA® Titan Xp GPUs cluster equipped with CUDA 10.1 driver. Neural ODEs were trained using the `torchdiffeq` (Chen u. a., 2018) PyTorch package.

2.4.1 Flows for annuli dataset

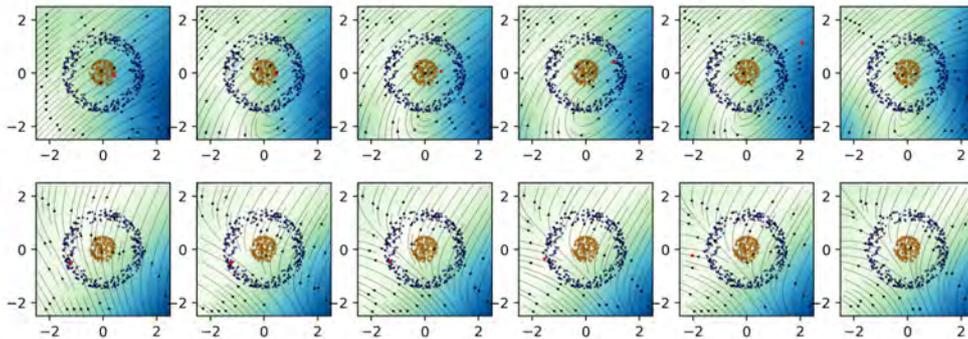


Figure 2.6 – Time-varying flows for two points in different classes (inner: top row; outer: bottom row) of the annuli dataset. Time advances from left to right.

2.4.2 Real-time pattern memorization

To illustrate the idea that closed-loop network control is particularly valuable on problems involving flexible adaptation to non-stationary stimuli, we trained NCODE on a dynamic, few-shot memorization task originally proposed in (Miconi u. a., 2018b). Here, the model is cued to quickly memorize sets of sequentially presented n -dimensional binary patterns and to reconstruct one of these patterns when exposed to a degraded version. For each presentation episode, we exposed the model sequentially to $m = 2, 5$, or 10 randomly generated n -length ($n = 100, 1000$) bit patterns with $\{-1, 1\}$ -valued bits. Then, we presented a degraded version of one of the previous patterns in which half of the bits were zeroed, and we tasked the system with reconstructing this degraded pattern via the dynamic state, $\mathbf{x}(t)$. This is a difficult task since it requires the memorization of several high-dimensional data in real-time. Miconi u. a. (2018b) have shown that, although recurrent networks are theoretically able to solve this task, they struggle during learning, while models

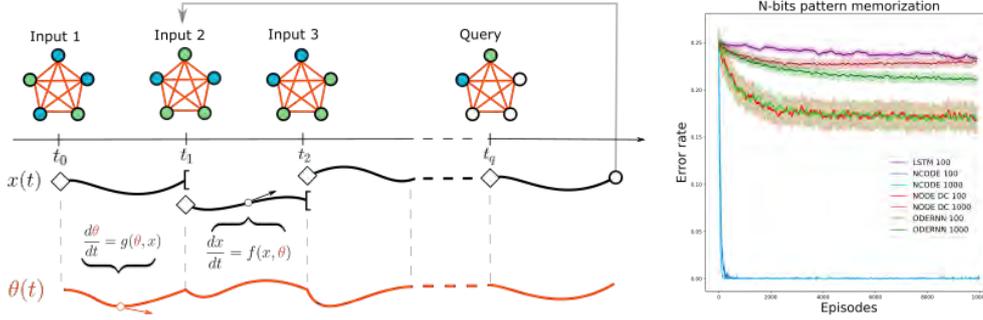


Figure 2.7 – **Left**: Diagram of an episode of the few-shot memorization task initially presented in [Miconi u. a. \(2018b\)](#). Three, 5-length $-1, 1$ -valued bit patterns ($n = 5$ here for visualization) are presented to the system at regular intervals. At query time, t_q , a degraded version of one of the patterns is presented and the task of the model is to complete this pattern. **Right**: Performance of a closed-loop N-CODE, an LSTM, a continuous-time ODE-RNN [Rubanova u. a. \(2019\)](#) and the data-conditioned NODE of [Massaroli u. a. \(2020b\)](#) averaged over 3 runs. All models besides N-CODE plateau at high values (chance is .25). N-CODE, on the other hand, learns immediately and with extremely high accuracy across all n .

storing transient information in dynamic synapses tend to perform better. Following this result, we evaluated the performance of a closed-loop N-CODE model in which the output of the control function takes the form of dynamic synapses (Eq. 2.16). Here, the control parameters are a matrix of dynamic weights, $\boldsymbol{\theta}(t) \in \mathbb{R}^{m \times m}$, which are governed by the outer product $\boldsymbol{\mu} \odot \mathbf{x}\mathbf{x}^T$,

$$\begin{cases} \left(\frac{d\mathbf{x}}{dt}, \frac{d\boldsymbol{\theta}}{dt} \right) = (\rho(\boldsymbol{\theta}\mathbf{x}), \boldsymbol{\mu} \odot \mathbf{x}\mathbf{x}^T) \\ (\mathbf{x}(0), \boldsymbol{\theta}(0)) = (\mathbf{x}_{\text{stim}}, \boldsymbol{\theta}_0) \end{cases} \quad (2.16)$$

where $\boldsymbol{\mu}$ is matrix of learned parameters, $\boldsymbol{\theta}_0$ is a learned initial condition, ρ is an element-wise sigmoidal non-linearity, \odot is element-wise multiplication, and x_{stim} is the presented stimulus. We have suppressed t for concision.

We ran this experiment on several recurrent neural networks for comparison. The different considered models are listed below:

- *Open-loop control*: We adopt the implementation of ([Massaroli u. a., 2020b](#)) by appending the presented pattern \mathbf{x}_{stim} to the dynamic variable $\mathbf{x}(t)$ such that $\boldsymbol{\theta}(t) = \boldsymbol{\theta}$ is linear layer $L(\mathbb{R}^{2 \times N}, \mathbb{R}^N)$.
- *Closed-loop control*: The weights are fully dynamic and the influ-

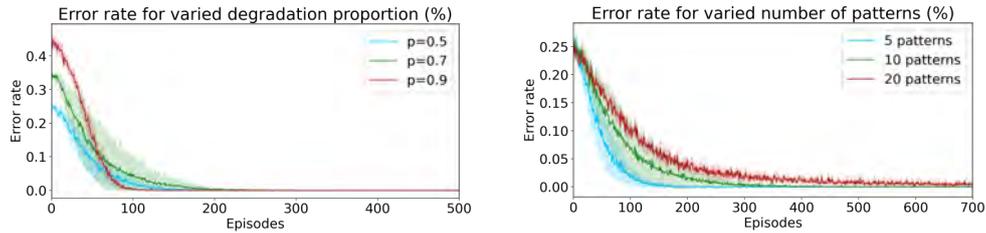


Figure 2.8 – Error rate of NCODE on the 1000-bit reconstruction task as a function of episodes for **(Left)** different proportions of degraded bits and **(Right)** different number of patterns presented in one episode. In all cases, the model learns in several hundreds episodes.

ence of the plastic evolution is tuned by learning the components μ that apply an element-wise gain for every weight velocity, according to Eq. 2.16.

- *ODE-RNN*: Here, a static-weight NODE models the evolution of a continuous-time single-cell recurrent network whose hidden state is carried over from one presentation to the next. A supplementary linear read-out $L(\mathbb{R}^N, \mathbb{R}^N)$ outputs the model’s guess and balances the number of parameters with other models.
- *LSTM*: A vanilla LSTM cell (Hochreiter und Schmidhuber, 1997b). In coherence with (Miconi u. a., 2018b) results, the hidden state dimension need to be greatly increased for the module to start memorizing. We tested a 5000-dimensional hidden state for the reported results.

All systems had the same number of learnable parameters, except for the LSTM which was much larger. For each model, we learn the model weights with gradient descent using an Adam optimizer with a learning rate $\lambda = 3e - 4$ and use the L_2 distance between the reconstruction and objective pattern as our objective function. For continuous models, each episode consists of a sequential presentation of each pattern for 0.5 sec followed by a query time of 0.5 sec. For the LSTM, we adopted the setting of (Miconi u. a., 2018b) where the sequence of presentation is discretized into 5 time-steps of presentations. We found that all considered models, except closed-loop N-CODE, struggled to learn this task (Fig. 2.7). Note that chance is .25 since half the bits are degraded. A discrete LSTM endowed with a much larger hidden state of 5000 neurons learns marginally better than chance level. ODE-RNN and data control versions plateau at 17.5% error rate. For both n tested, NCODE learned the problem not

only much better, but strikingly faster. We tested for episodes of 3,5 and 10 patterns with 2 presentations in random order and different proportions of degradation (0.5,0.7 and 0.9). Static-weight model performance rapidly deteriorated in more challenging settings, whereas N-CODE reconstruction converged to a residual error below 1% in all cases.

2.4.3 Oscillatory dynamics: Van der Pol and Kuramoto

Networks of coupled oscillators are some of the most studied objects in the theory of dynamical systems and are of particular interest for computational neuroscience. Two important areas of current interest are the study of synchrony in highly disordered systems and the modeling of systems with adaptive network structures. NCODE represents a single approach to both of these problems by offering the possibility to learn how to synchronize across a distribution of disordered network conditions. The key feature of the model is the replacement of the traditionally static couplings with a coupling function which can learn optimal interactions within heterogeneous oscillator populations.

Kuramoto model of synchronization

Networks of coupled oscillators represent a class of objects among most studied objects in the theory of dynamical systems. Two important questions concern the study of synchrony in highly disordered systems and the modeling of networks with adaptive coupling structures. While previous literature has mostly focus on designing analytical descriptions to these two questions, we showed that we can learn to replace traditionally static couplings with a learned coupling function which can associate to oscillatory states, interactions specifically designed to determine synchronization coherence within the network. Specifically, we examined the popular Kuramoto model (Kuramoto, 1975) of n oscillators, whose dynamics are governed by the following equation:

$$\dot{\omega}_i = \mu_i + \sum_{j=1}^n K_{ij} \sin(\omega_j(t - \tau_j) - \omega_i(t)) + b_i \sin(\omega_i). \quad (2.17)$$

We consider the problem of learning a function $\gamma : \mathcal{X} \mapsto \mathcal{K}$ from a feature space \mathcal{X} (here we take $\mathcal{X} = \mathbb{R}$) to the couplings $(K_{ij})_{i,j} \in \mathcal{K}$ defined in equation (2.17), such that a function $\mathcal{L} : \Omega \mapsto \mathbb{R}$ is minimized for a certain state of the vector $\omega_i(t)$. In other word, we consider the problem of learning to leverage the oscillatory dynamics defined by equation (2.17) to perform a task specified by the loss function \mathcal{L} . This problem is exactly

an instance of the open-loop control system discussed in section 2.3.3. While a particular discussion on the Kuramoto model is out of the scope of this thesis, we propose an application for machine learning in the form of learning a clustering function from a set of points $(\mathbf{x}_i)_{i \leq I}$ in the feature space that we describe in figure 2.9.

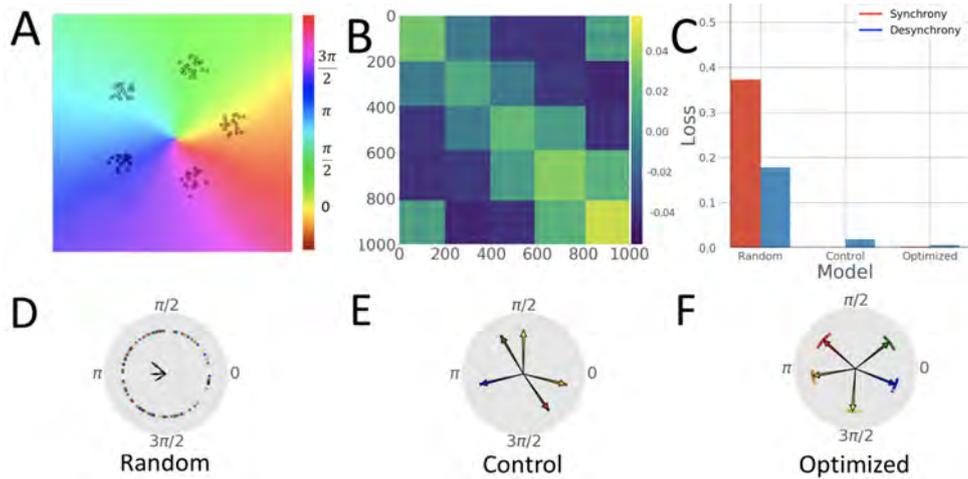
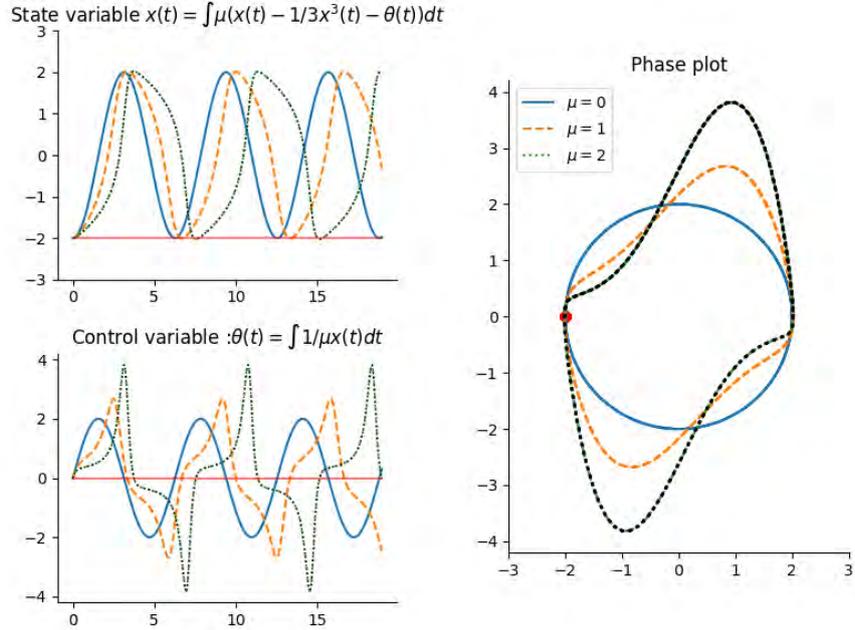


Figure 2.9 – Cluster synchrony for a 5-component gaussian mixture model. (A) The phase landscape in $\mathcal{X} = R^2$ is plotted by recording the asymptotic phase of a single oscillator, i , in a fixed batch (small circles) as its 2-D feature was varied over a grid of 100 x 100 values. (B) The trained system learns to produce coupling matrices with a clear block-diffusive structure whose cells correspond to the true labeling function on X . (C) However, not just any block coupling structure can produce this effect, since the optimized model (right bars) achieves a lower total loss than both a randomized model (left bars) and a block control model (middle bars) with positive intra-cluster couplings and negative inter-cluster couplings. (D,E,F) The differences among these models is all the clearer when plotting terminal phases (small dots) and cluster mean fields (colored arrows). Here, phases and mean fields are colored according to ground truth clusters, not angle. While the control model can easily exhibit strong intra-group synchrony (E, only the balance of positive and negative weights in the learned diffusive couplings of the optimized model (F) can also produce reliable splay configurations. Taken from Ricci u. a. (2021).

We additionally show that augmentation of NODEs through closed-loop control also enhance the class of dynamics that can be represented. In particular, we show through the Van der Pol oscillator example that higher order differential equations can be represented.

Van der Pol oscillator



$$\begin{aligned} \frac{d^2 \mathbf{x}}{dt^2} &= \boldsymbol{\mu}(1 - \mathbf{x}^2) \frac{d\mathbf{x}}{dt} - \mathbf{x} \\ \Leftrightarrow \begin{cases} \dot{\mathbf{x}} &= \boldsymbol{\mu}(\mathbf{x} - \frac{1}{3}\mathbf{x}^3 - \boldsymbol{\theta}) \\ \dot{\boldsymbol{\theta}} &= \frac{1}{\boldsymbol{\mu}}\mathbf{x} \end{cases} \end{aligned}$$

Figure 2.10 – Example of dynamical control augmentation: The Van der Pol oscillator: As a second order differential equation, the dynamics cannot be approximated by a single 1-dimensional NODE with a constant control $\boldsymbol{\theta}$ (degenerate solution in red). However, if the dynamics are decomposed into a planar system with a dynamic control variable $\boldsymbol{\theta}(t)$, then the parameter $\boldsymbol{\mu}$ can be adapted to fit a particular oscillatory regime (in black). This is a particular form of augmentation discussed in (Dupont u. a. (2019); Norcliffe u. a. (2020)) that showcases the benefit of using additional variables as evolving parameters of the dynamical system.

2.4.4 Supervised image classification

For MNIST and CIFAR-10 classification, we defined the equation of motion f as sequence of convolutional filters replicating a ResNet block architecture with padding to conserve dimensionality and non-linear ReLU

activation. The base architecture takes one of the forms

- 1×1 , k filters, 1 stride, 0 padding
- 3×3 , k filters, 1 stride, 1 padding
- 1×1 , $c + a$ filters, 1 stride, 0 padding

with k the number of filters, c the number of channels (1 for MNIST and 3 for CIFAR10) and a the augmentation channel (Dupont u. a., 2019). The open-loop controller is a very simple linear transformation of the data into 10 hidden units followed by multiple heads outputting the vector of weights for each convolution kernel. In order to examine the effect of the size of dimensionality of the equation of motion on learnability, we tried different parameterizations for f :

- *NCODE* : The three kernels are conditioned by the controller γ .
- *NCODE* 3×3 : The inner 3×3 kernels are dynamically conditioned.
- *NODE DC* : This corresponds to the form of data-control proposed by (Massaroli u. a., 2020b) where the image data is concatenated with state $\mathbf{x}(t)$ at each evaluation of the first 1×1 convolution kernel.

All parameters not adaptively controlled are simply learned as fixed parameters. For all models we used the adaptative solver Dormund-Prince with tolerance of $1e-3$.

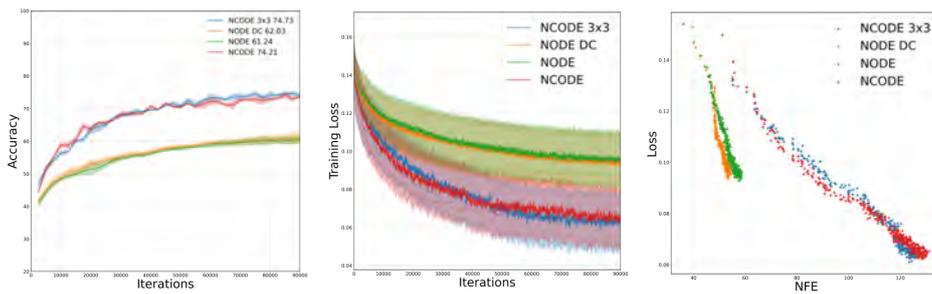


Figure 2.11 – Accuracy, Train and test losses over training interations and Number of function evaluations (NFE) vs loss for several versions of NCODE on CIFAR-10.

The results show that N-CODE models achieve higher accuracy in fewer epochs and overall lower loss (Fig. 2.11, top panels). However, the

simplicity of the mapping γ in this experiment and the absence of regularization of the induced map Φ_γ (Finlay u. a., 2020) yields a complex flow requiring a higher total number of function evaluations (Fig. 2.11, lower panels). This potentially prevents NCODE from achieving even better accuracy. Interestingly, the results for the partially controlled systems suggest that gradient descent accommodates the learning of hybrid modules f with both static and dynamic convolution parameters. These results advocate for the exploration of specific closed-loop formulations and regularization of control for convolution layers which we leave for future work.

2.4.5 Implementation of generic NCODE module

```
class NCODE(torch.nn.Module):
    def __init__(self, *args):
        super(NCODE, self).__init__()

        #Definition of the ODE system
        self.h = ...

    def forward(self, t, z):
        #Unpack variables (theta_t is potentially a tuple of tensors)
        x_t, *theta_t = z

        #Compute evolution od the system
        d_x_t, d_theta_t = self.f_g(x_t,theta_t)

        return (d_x_t, *d_theta_t)

class NCODE_wrapper(torch.nn.Module):
    def __init__(self, *args):
        super(NCODE_wrapper, self).__init__()
        #Definition of module and controller
        self.gamma = ...
        self.NCODE = NCODE

    def forward(self, x_0):
        #Set initial conditions
        self.theta_0 = self.gamma(x_0)
        z_0 = (x_0,*self.theta_0)

        #Run the system forward with defined settings
```

```
x_t, *theta_t = odeint(self.NCODE, z_0, t, *options,
adjoint_params=self.NCODE.parameters())

return (x_t, *theta_t)
```

2.5 Controlled Normalizing flows

As mentioned previously, an important application of Neural ODEs are continuous normalizing flows, which comes from a physical interpretation of Neural ODEs flows as defining an invertible transformation of densities. As such, they are able to define expressive energy-based expressions of multivariate distributions. More specifically, a continuous transformation of $\mathbf{x}(t)$ described by the ODE “ $d\mathbf{x}/dt = f$ ” defines a deterministic flow for each data particle. Let $p(\mathbf{x}, t)$ be the probability density of the continuous random variable $\mathbf{x}(t)$. The change of this density is completely determined by f as p follows the *continuity equation* (Batchelor, 2000):

$$\frac{\partial p(\mathbf{x}, t)}{\partial t} = -\nabla_{\mathbf{x}} \cdot (pf) \tag{2.18}$$

where $\nabla_{\mathbf{x}}$ is the divergence operator over \mathbf{x} . From this result can be derived a change in log-density also referred to as the "Instantaneous Change of Variables" that is easier to manipulate:

Theorem 2 *Instantaneous Change of Variables, (Chen u. a., 2018).*
 Defining the material derivative $\frac{dp}{dt} = \frac{\partial p}{\partial t} + \nabla_{\mathbf{x}} p \cdot \frac{d\mathbf{x}}{dt}$, we have:

$$\frac{d \log p(\mathbf{x}, t)}{dt} = -\nabla_{\mathbf{x}} \cdot f \tag{2.19}$$

The proof of this result is a mere application of Leibniz rule of calculus. The continuity equation $\frac{\partial p(\mathbf{x}, t)}{\partial t} = -\nabla_{\mathbf{x}} p \cdot f - p \cdot \nabla_{\mathbf{x}} f$ implies that $\frac{dp(\mathbf{x}, t)}{dt} = -p \cdot \nabla_{\mathbf{x}} f$. Hence, we can write $\frac{d \log p(\mathbf{x}, t)}{dt} = \frac{1}{p} \frac{dp(\mathbf{x}, t)}{dt} = -\frac{p}{p} \nabla_{\mathbf{x}} f$.

This result allows to train neural ODEs to perform variational inference by relating some data distribution p_d to some reference distribution p_0 for which the log-probability is computable such as a multivariate gaussian distribution. The total change in logprobability after time T can be written:

$$\log p(\mathbf{x}, 0) = \log p(\mathbf{x}, t) - \int_0^T \nabla_{\mathbf{x}} \cdot f dt \tag{2.20}$$

With this theoretical framework, continuous normalizing flows offer nice properties to model, evaluate and sample arbitrary distributions provided the underlying vector field f is powerful enough to fit the geometry of the target distribution in the feature space.

Controlled flows

Depending on the context and available information, the beliefs of an agent over the state of the observed world can vary. One way to model this variability is namely to define a probability distribution that can be conditioned by available information. In figure , we show that we can learn an "adaptive" energy function of the latent space and that by the same token as previously, NCODE can enhance the continuous normalizing flows by specifying an explicit control signal. An important application of this general framework is recent generative diffusion models. [Song u. a.](#) showed that stochastic denoising diffusion models define an implicit normalizing flow ([Song u. a.](#)).

$$f(\mathbf{x}, \mathbf{y}) \stackrel{\text{def}}{=} \begin{cases} \mathcal{X} \times \mathcal{Y} \mapsto \mathcal{X} \\ (\mathbf{x}, \mathbf{y}) \mapsto f(\mathbf{x}, \boldsymbol{\theta}(y)) \end{cases} \quad (2.21)$$

2-dimensional example of energy interpolation

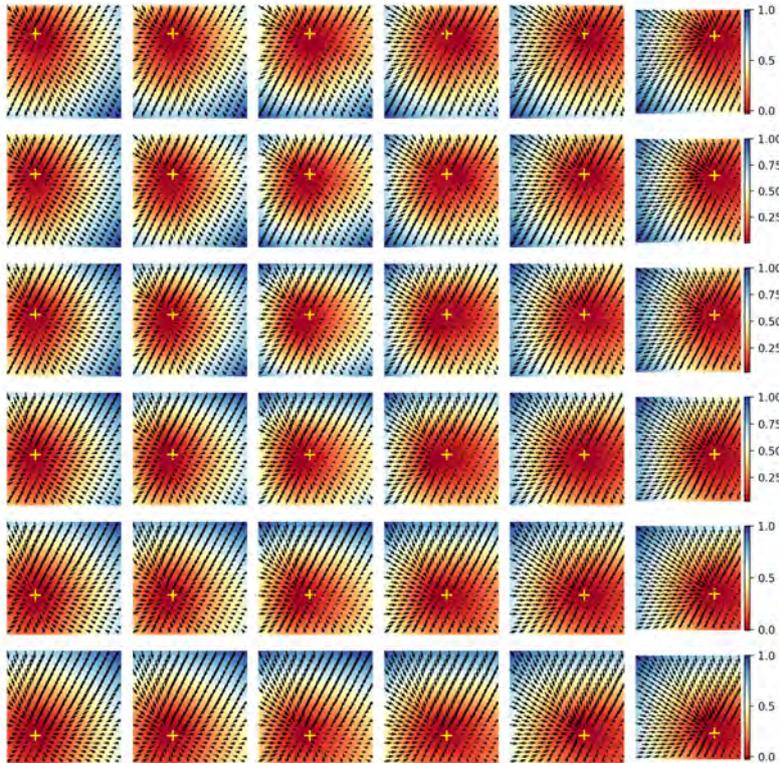


Figure 2.12 – Example of two-dimensional vector fields (arrows) and energetic probabilities (colors) of normalizing flow controlled with targets combinations $(x, y) \in [-1, 1]^2$. Despite being trained on punctual control targets combinations $(x, y) \in \{-1, 1\}^2$ (i.e corresponding to corners of the panel.), the vector field flexibly interpolates new targets. This regularity is certainly one factor explaining the ability of diffusion models to converge towards satisfying samples on the image manifold as they define an implicit normalizing flow (Song u. a.)

2.6 Discussion

In this work, we have presented an original control formulation for continuous in time parametric neural feature transformations (NODEs). We have shown that it is possible to dynamically shape the trajectories of a continuous-time neural network by augmenting the network with a trained control mechanism operating on the network connection space, and further demonstrated that this can be applied in the context of supervised and unsupervised representation learning. Although the general idea is simple, this generalization has founds echoes in recent research (Choromanski u. a., 2020b; Irie u. a., 2022a). This work calls for the investigation the robustness and generalization properties of such con-

trolled models as well as their similarities with fast-synaptic modulation systems observed in neuroscience, and test this on natural applications such as vision and robotics. An additional avenue for further research is the connection between our system and the theory of bifurcations in dynamical systems and neuroscience.

Chapter 3

Learning Functional transduction

“One can imagine that the ultimate mathematician is one who can see analogies between analogies.”

Stephan Banach

Contents

3.1	Context	71
3.2	Related work	73
3.3	The functional regression problem	75
3.4	Reproducing Kernel Banach Space regression	76
3.5	Transducer	78
3.6	Applications to finite-dimensional meta-learning	83
3.7	Applications to ∞-dimensional problems . .	83
3.8	Experimental details	90
3.9	Discussion	94

Research in Machine Learning has polarized into two general regression approaches: Transductive methods derive estimates directly from available data but are usually problem unspecific. Inductive methods can be much more particular, but generally require tuning and compute-intensive searches for solutions. In this chapter, we adopt a hybrid approach: We leverage the theory of Reproducing Kernel Banach Spaces (RKBS) and show that transductive principles can be induced through gradient descent to form efficient *in-context* neural approximators. We apply this approach to RKBS of function-valued operators and show that

once trained, our *Transducer* model can capture on-the-fly relationships between infinite-dimensional input and output functions, given a few example pairs, and return new function estimates. We demonstrate the benefit of our transductive approach to model complex physical systems influenced by varying external factors with little data at a fraction of the usual deep learning training computation cost for partial differential equations and climate modeling applications.

3.1 Context

In statistical learning, transduction refers to the process of reasoning directly from observed (training) cases to new (test) cases, and contrasts with induction, which amounts to extracting general rules from observed training cases. [Vapnik \(2006\)](#) condensed this philosophy into an imperative principle for statistical learning: *"When solving a problem of interest, do not solve a more general problem as an intermediate step. Try to get the answer that you really need but not a more general one."* Transductive machine learning has yielded some of the most useful regression algorithms, from K -nearest neighbors ([Cover und Hart, 1967](#)) to support vector machines ([Boser u. a., 1992](#)) or Gaussian processes ([Williams und Rasmussen, 1995](#)). A major advantage of such systems is their wide applicability and the straightforward construction of estimates. Research in deep learning, however, has mostly endeavored to find inductive solutions by crafting networks that generalize from large datasets, relying on the empirical evidence that stochastic gradient descent is a powerful control rule faithfully encoding functional relationships between input and outputs into the network weights. Nevertheless, this paradigm has shown limits: Gradient-based training is compute-intensive and requires a large amount of data to approximate a single functional map. This may be particularly problematic for real-world applications where data has heterogeneous sources or only a few examples of the relationship of interest are available. Moreover, these solutions generalize poorly outside of the training distribution ([Jin u. a., 2020](#)), and a slight modification of the target function might require expensive retraining as well as forgetting of the previous solution ([McCloskey und Cohen, 1989](#)). Finally, they show poor robustness to adversarial attack ([Szegedy u. a., 2013](#)) or anomalous data contamination of the training distribution ([Zhang u. a., 2021](#)). Ultimately, to become practically useful, such data-driven methods should also perform well outside of the data set they are trained with and they should be able to extrapolate to different parameters and contexts. Still, deep learning systems capable of flexibly and robustly adapting their re-

sponse to small data regimes are yet to come.

In this chapter, we explore a hybrid solution to tackle functional regression problems defined in Banach spaces by meta-learning a deep transductive regression model that can produce relevant estimates given original datasets. This meta-problem represents itself as an infinite-dimensional regression problem, which consists in finding a program associating a target function to its corresponding exemplar dataset. We leverage the theory of Reproducing Kernel Banach Spaces (RKBS) (Zhang, 2013; Lin u. a., 2022) and interpret the recent attention mechanism of the Transformer (Vaswani u. a., 2017) as a parametric form of a reproducing kernel that we use to build functional approximators in various RKBS. Our framework allows us to handle both RKBS defined between finite and infinite-dimensional spaces, allowing us to manipulate functional data, and performing operator regression given a few examples of input-output pairs. While kernel regression is usually plagued with the “curse of dimensionality” (Bellman, 1966; Aggarwal u. a., 2001), we show that our approach escapes this pitfall and manages to learn well-adapted operator-valued RKBS kernels. From a control perspective, our method can be interpreted as building an *open-loop controller* where input-output examples condition a neural operator to produce a desired functional response. From a machine learning perspective, our method can be interpreted as a form of *in-context* learning in infinite-dimensional function spaces.

Contributions \diamond We introduce a novel meta-learning approach leveraging reproducing kernel theory and deep learning methods to perform instantaneous regression of an infinity of functions in reproducing kernel spaces.

- Our model learns an implicit regression program able to identify, in a single feedforward pass, elements of specific functional spaces from any corresponding collection of input-output pairs describing the target function. Such ultra-fast regression program, which bypasses the need for gradient-based training, is also general and can be applied to functions either defined on finite dimensional spaces (scalar-valued function spaces) or infinite dimensional spaces (function-valued operator spaces).
- In particular, we demonstrate the flexibility and efficiency of our framework for fitting function-valued operators in two PDEs and one climate modeling problem. We show that our transductive approach allows for better generalization properties of neural operator regression, better precision when relevant data is available, and can

be combined with iterative regression schemes that are too expensive for previous inductive approaches, thus holding great potential to improve neural operators applicability.

- To the best of our knowledge, our proposal is the first to marry vector-valued RKBS theory with deep meta-learning and might also shed new light on the in-context learning abilities observed in deep attentional architectures.

The remainder of the paper is organized as follows: in Section 3.2 we discuss previous related work. In Section 3.3 we introduce the mathematical formulation of the functional transductive regression problem and describe more formally our model in section 3.5. We report in section 5.5 experimental results in different settings. Finally, in Section 3.9 we summarise our work and outline future research directions.

3.2 Related work

Transductive Machine learning – The idea of determining statistical predictions directly from exemplar data has been more formally described in [Gammerman u. a. \(1998\)](#); [Vapnik \(1999\)](#) with pervasive applications in Machine Learning: Estimators relying on relational structures between data points such as K -nearest neighbors ([Cover und Hart, 1967](#)) and kernels methods ([Nadaraya, 1964](#); [Watson, 1964](#)) build regression estimates by weighting examples with respect to a certain metric space. Further, the “kernel trick” allows to embed possibly infinite-dimensional features ([Ferraty und Vieu, 2006](#)) into finite Gram matrix representations that are also well-suited for multi-task regression ([Evgeniou u. a., 2005](#); [Caponnetto u. a., 2008](#)). Gaussian processes ([Williams und Rasmussen, 1995](#)) combine transduction with Bayesian modeling to regress a posterior distribution over possible functions. However, these techniques might suffer from the so-called “curse of dimensionality”: with growing problem dimensionality, the density of the exemplar point diminishes, which increases their statistical variance. More recent work combining deep learning with transductive inference has shown promising results for few-shot learning even in high-dimensional spaces ([Snell u. a., 2017a](#)) or for sequence modeling ([Jaitly u. a., 2015](#)), but the vast majority of neural networks still remain purely inductive. Finally, modern Hopfield networks ([Ramsauer u. a., 2020](#)) have been shown to implement various dynamic operations on sets of representations related to the Transformer attention, offering the possibility to leverage transductive computation in the deep learning realm.

Neural operator learning – While regression with neural networks has primarily been applied to learning mappings between finite-dimensional spaces, their impressive approximation abilities have been generalized to operator mappings between function spaces. [Chen und Chen \(1995\)](#) showed that finite neural parametrization can approximate well infinite-dimensional functional relationships. More recent work on neural operator regression has shown promising results ([Lu u. a., 2019](#)), especially when mixed with tools from functional analysis and physics ([Raissi u. a., 2017](#); [Li u. a., 2020a](#); [Gupta u. a., 2021](#); [Li u. a., 2020b](#); [Nelsen und Stuart, 2021](#); [Wang u. a., 2021](#); [Roberts u. a., 2021](#)) and constitutes a booming research direction in particular for physical applications ([Goswami u. a., 2022](#); [Pathak u. a., 2022](#); [Vinuesa und Brunton, 2022](#); [Wen u. a., 2022](#); [Pickering u. a., 2022](#)). Such systems are able to approximate infinite-dimensional functional objects in a highly compressed form, and potentially offer a much more powerful substrate for learning and manipulating representations than static vectors. Orthogonally, while the recent ground-breaking attentional computation of the Transformer was interpreted as a Petrov-Galerkin projection ([Cao, 2021](#)) or through Reproducing Kernel Hilbert Space theory ([Kissas u. a., 2022](#)) for building neural operators, these perspectives propose to apply attention to fit a single target operator. We will show the theory of reproducing kernel Banach space suggests that we can take further advantage of kernel modeling together with neural operators in order to flexibly solve infinitely many operator regression problems.

Meta-learning and in-context learning – Research for improving deep learning training has accelerated in recent years. Promising work has regarded this problem from the perspective of "learning to learn" or meta-learning ([Schmidhuber u. a., 1997](#); [Vilalta und Drissi, 2002](#)), by either explicitly treating a gradient descent sequence as an optimizable object ([Finn u. a., 2017a](#)) or modeling an optimizer as a black-box autoregressive model ([Santoro u. a., 2016a](#); [Ravi und Larochelle, 2017](#)). Another orthogonal approach aims at learning a generalizing metric that allows for straightforward object class regression in a latent space ([Snell u. a., 2017b](#); [Sung u. a., 2018](#)). More recently, converging findings in various domains from reinforcement learning ([Mishra u. a., 2018](#); [Laskin u. a.](#)), natural language processing ([Brown u. a., 2020a](#); [Xie u. a., 2021](#); [Olsson u. a., 2022](#)) and functional regression ([Garg u. a., 2022](#)) have established the ability of set-based attentional computation in the Transformer ([Vaswani u. a., 2017](#)) for *in-context* learning, by flexibly extracting functional relationships and performing dynamic association such as linguistic analogy or few-shot behavioral imitation. We show that the theory of RKBS can

help interpret such property and extends it to function-valued operators regression.

3.3 The functional regression problem

3.3.1 Notations

Let \mathcal{V} and \mathcal{U} be two (finite or infinite-dimensional) Banach spaces, respectively referred to as the input and output space, and let \mathcal{B} a Banach space of functions from \mathcal{V} to \mathcal{U} . We also note $L(\mathcal{U}, \mathcal{B})$ (resp. $L(\mathcal{U})$) the set of bounded linear operators from \mathcal{U} to \mathcal{B} (resp. to itself). In this chapter, we leverage the structure of reproducing kernel Banach spaces of functions \mathcal{B} and combine it with the universal approximation abilities of deep networks. As we will see in the experimental section, RKBS are very general spaces occurring in a wide range of machine learning applications. We start by recalling some elements of the theory of vector-valued RKBS developed in [Zhang \(2013\)](#). Namely, we will consider throughout *uniform* Banach spaces \mathcal{S} (such condition guarantees the unicity of a compatible semi-inner product $\langle \cdot, \cdot \rangle_{\mathcal{S}} : \mathcal{S} \times \mathcal{S} \mapsto \mathbb{R}$, i.e. $\forall \mathbf{s} \in \mathcal{S}, \langle \mathbf{s}, \mathbf{s} \rangle_{\mathcal{S}} = \|\mathbf{s}\|_{\mathcal{S}}^2$ and allows to build a bijective and isometric dual space \mathcal{S}^*).

3.3.2 The meta-regression problem

We consider the *meta-learning* problem of creating a function \mathcal{T} able to approximate any functional element \mathcal{O} in the space \mathcal{B} from any finite collection of example pairs $\mathcal{D}_{\mathcal{O}} = \{(\mathbf{v}_i, \mathbf{u}_i) \mid \mathbf{v}_i \in \mathcal{V}, \mathbf{u}_i = \mathcal{O}(\mathbf{v}_i)\}_{i \leq n}$. A prominent approach in statistical learning is *empirical risk minimization* which consists in predefining a class $\tilde{\mathcal{B}} \subset \mathcal{B}$ of computable functions from \mathcal{V} to \mathcal{U} and subsequently selecting a model $\tilde{\mathcal{O}}$ as a minimizer (provided its existence) of a risk function $\mathcal{L} : \mathcal{B} \times \mathcal{D} \mapsto \mathbb{R}$:

$$\mathcal{T}(\mathcal{D}_{\mathcal{O}}) \in \underset{\tilde{\mathcal{O}} \in \tilde{\mathcal{B}}}{\operatorname{argmin}} \mathcal{L}(\tilde{\mathcal{O}}, \mathcal{D}_{\mathcal{O}}) \quad (3.1)$$

For instance, the procedure consisting in performing gradient-based optimization of objective (3.1) over a parametric class $\tilde{\mathcal{B}}$ of neural networks defines implicitly such a function \mathcal{T} . Fundamentally, this technique works by induction: It captures the statistical regularities of a single map \mathcal{O} into the parameters of the neural network $\hat{\mathcal{O}}$ such that $\mathcal{D}_{\mathcal{O}}$ is discarded for inference. Recent examples of gradient-based optimization of neural networks for operator regression (i.e when \mathcal{V} and \mathcal{U} are infinite-dimensional) are DeepOnet ([Lu u. a., 2019](#)) or Fourier Neural Operator (FNO) ([Li](#)

u. a., 2020a). As previously discussed, for every regression problem instance, evaluating \mathcal{T} with these approaches requires a heavy training procedure. Instead, we show in this work that for specific spaces \mathcal{B} , we can meta-learn a parametric map \mathcal{T}_θ that transductively approximates (in a certain functional sense) any target function $\mathcal{O} \in \mathcal{B}$ given a corresponding dataset $\mathcal{D}_\mathcal{O}$ such that:

$$\forall \mathbf{v} \in \mathcal{V}, \mathcal{T}(\mathcal{D}_\mathcal{O})(\mathbf{v}) = \mathcal{T}_\theta(\mathbf{v}_1, \mathcal{O}(\mathbf{v}_1), \dots, \mathbf{v}_n, \mathcal{O}(\mathbf{v}_n), \mathbf{v}) \approx \mathcal{O}(\mathbf{v}) \quad (3.2)$$

3.4 Reproducing Kernel Banach Space regression

We start by recalling some elements of the theory of Reproducing Kernel Banach Spaces (RKBS) developed in Zhang (2013), which will be useful to build our neural kernel regression model. For a more in-depth treatment of RKBS, see also Lin u. a. (2022).

Theorem 3 (Vector-valued RKBS) *A \mathcal{U} -valued reproducing kernel Banach space \mathcal{B} of operators from \mathcal{V} to \mathcal{U} is a Banach space such that for all $\mathbf{v} \in \mathcal{V}$, the point evaluation $\delta_\mathbf{v} : \mathcal{B} \mapsto \mathcal{U}$ defined as $\delta_\mathbf{v}(\mathcal{O}) = \mathcal{O}(\mathbf{v})$ is continuous. In this case, there exists a unique function $\mathcal{K} : \mathcal{V} \times \mathcal{V} \mapsto \mathcal{L}(\mathcal{U})$ such that for all $(\mathbf{v}, \mathbf{u}) \in \mathcal{V} \times \mathcal{U}$:*

$$\begin{cases} \mathbf{v}' \mapsto \mathcal{K}(\mathbf{v}, \mathbf{v}')(\mathbf{u}) \in \mathcal{B} \\ \forall \mathcal{O} \in \mathcal{B}, \langle \mathcal{O}(\mathbf{v}), \mathbf{u} \rangle_{\mathcal{U}} = \langle \mathcal{O}, \mathcal{K}(\mathbf{v}, \cdot)(\mathbf{u}) \rangle_{\mathcal{B}} \\ \forall \mathbf{v}' \in \mathcal{V}, \|\mathcal{K}(\mathbf{v}, \mathbf{v}')\|_{\mathcal{L}(\mathcal{U})} \leq \|\delta_\mathbf{v}\|_{\mathcal{L}(\mathcal{B}, \mathcal{U})} \|\delta_{\mathbf{v}'}\|_{\mathcal{L}(\mathcal{B}, \mathcal{U})} \end{cases} \quad (3.3)$$

Informally, definition (3.3) states that the image of *any* function \mathcal{O} of \mathcal{B} at a given point \mathbf{v} can be expressed in terms of a single function \mathcal{K} . The latter is hence called the *reproducing kernel* of \mathcal{B} and our goal is to leverage the unicity of \mathcal{K} to form a single approximator for all operators in \mathcal{B} . However, the previous theorem does not provide explicit information on how to construct such function \mathcal{K} . Fortunately, the feature maps characterization of RKBS will give us a structure to build \mathcal{K} . We first define for any linear operator $T \in \mathcal{L}(\mathcal{S}_1, \mathcal{S}_2)$ between two Banach spaces $\mathcal{S}_1, \mathcal{S}_2$, the generalized adjoint $T^\dagger \in \mathcal{L}(\mathcal{S}_2, \mathcal{S}_1)$ as the application verifying $\langle T\mathbf{s}, \mathbf{s}' \rangle_{\mathcal{S}_1} = \langle \mathbf{s}, T^\dagger \mathbf{s}' \rangle_{\mathcal{S}_2}$ for all $(\mathbf{s}, \mathbf{s}') \in \mathcal{S}_1 \times \mathcal{S}_2$.

Theorem 4 (Feature map characterization of RKBS) *A function $\mathcal{K} : \mathcal{V} \times \mathcal{V} \mapsto \mathcal{L}(\mathcal{U})$ is the reproducing kernel of some \mathcal{U} -valued RKBS if and only if there exist a Banach space \mathcal{B} and a mapping $\Phi : \mathcal{V} \mapsto \mathcal{L}(\mathcal{B}, \mathcal{U})$ such that:*

$$\begin{cases} \forall (\mathbf{v}, \mathbf{v}') \in \mathcal{V}^2, \mathcal{K}(\mathbf{v}, \mathbf{v}') = \Phi(\mathbf{v}')(\Phi^\dagger(\mathbf{v})) \\ \overline{\text{span}}\{(\Phi(\mathbf{v}))^\dagger(\mathbf{u})\}^*, \mathbf{v} \in \mathcal{V}, \mathbf{u} \in \mathcal{U}\} = \mathcal{B}^* \end{cases} \quad (3.4)$$

with $\Phi^\dagger : \mathcal{V} \mapsto \mathcal{L}(\mathcal{U}, \mathcal{B})$ defined by: $\forall \mathbf{v}, \Phi^\dagger(\mathbf{v}) = (\Phi(\mathbf{v}))^\dagger$.

Theorem (4) reformulates the kernel construction problem into a search for its corresponding feature maps Φ and Φ^\dagger (in RKBS, contrary to RKHS, the adjoint can generally not be directly expressed from the operator itself due to the non-additivity of the semi-inner product), for which we can find parametric approximations. It also constitutes a generalisation of the Transformer attention mechanism. To see this, let us assume that \mathcal{V} and \mathcal{U} are finite-dimensional and define $W_\theta \in \mathcal{L}(\mathbb{R}^p, \mathcal{U})$, $(Q_\theta, K_\theta) \in \mathcal{L}(\mathcal{V}, \mathbb{R}^p)^2$, $V_\theta \in \mathcal{L}(\mathcal{U}, \mathbb{R}^p)$ and σ a softmax normalization. Forming the following parametric kernel κ_θ ;

$$\kappa_\theta(\mathbf{v}, \mathbf{v}')(\mathbf{u}) \triangleq W_\theta \left(\sigma(Q_\theta(\mathbf{v}').(K_\theta(\mathbf{v}))^T) \cdot V_\theta(\mathbf{u}) \right) \quad (3.5)$$

we see that (3.6) corresponds to the Transformer key-query attention as well as a particular kernel construction following Theorem (4). We will use (3.6) to build our kernel, but note that other constructions are possible.

Proposition 1 (Dot-product attention as \mathcal{U} -valued RKBS) *Let $(p_j)_{j \leq J}$ a finite sequence of strictly positive integers, let $(A_\theta^j)_{j \leq J}$ be applications from $\mathcal{V} \times \mathcal{V}$ to \mathbb{R} , let V_θ^j be linear applications from $L(\mathcal{U}, \mathbb{R}^{p_j})$ and W_θ a linear application from $L(\prod_{j \leq J} \mathbb{R}^{p_j}, \mathcal{U})$, the (multi-head) application $\kappa_\theta : \mathcal{V} \times \mathcal{V} \mapsto L(\mathcal{U})$ defined by $\prod_{j \leq J}$*

$$\kappa_\theta(\mathbf{v}, \mathbf{v}')(\mathbf{u}) \triangleq W_\theta \left([\dots, A_\theta^j(\mathbf{v}, \mathbf{v}') \cdot V_\theta^j(\mathbf{u}), \dots]_{j \leq J} \right) \quad (3.6)$$

is the reproducing kernel of an \mathcal{U} -valued RKBS. In particular, if $\mathcal{U} = \mathcal{V} = \mathbb{R}^p$, for $p \in \mathbb{N}^+$ and $A_\theta^j = \exp(\frac{1}{\tau}(Q_\theta^j \mathbf{v})^T (K_\theta^j \mathbf{v}')) / \sigma(\mathbf{v}, \mathbf{v}')$ with $(Q_\theta^j, K_\theta^j)_{j \leq J}$ applications from $L(\mathcal{V}, \mathbb{R}^d)$, κ_θ corresponds to the dot-product attention mechanism of Vaswani u. a. (2017).

Finally, in order to apply the kernel representation to available data $\mathcal{E}_\mathcal{O}$ for approximating a specific operator \mathcal{O} , we present an extension of the representer theorem to vector-valued RKBS:

Theorem 5 (RKBS Representer Theorem) *Let \mathcal{B} be a operator-valued RKBS from \mathcal{V} to \mathcal{U} . Then the minimal norm interpolation problem:*

$$\inf_{\mathcal{O} \in \mathcal{B}} \{\|\mathcal{O}\|_{\mathcal{B}}, \text{ s.t. } \mathcal{O}(\mathbf{v}_i) = \mathbf{u}_i, i \leq I\} \quad (3.7)$$

admits a unique minimizer \mathcal{O}_0 . Moreover, \mathcal{O}_0 is the solution of (7.5) if and only if $\mathcal{O}_0(\mathbf{v}_i) = \mathbf{u}_i$ for all $i \leq I$ and $\mathcal{O}_0^ \in \overline{\text{span}}\{\mathcal{K}(\mathbf{v}_i, \cdot)(\mathbf{u})^*, i \leq I, \mathbf{u} \in \mathcal{U}\}$.*

The representer theorem tells us that there exists $(\mathbf{y}_i)_{i \leq I}$ functions in \mathcal{U} such that $\mathcal{O}_0^* = \sum_{i \leq I} \mathcal{K}(\mathbf{v}_i, \cdot)(\mathbf{y}_i)^*$. Here, contrary to reproducing kernel Hilbert space theory, we cannot directly solve an explicit linear problem in $(\mathbf{y}_i)_{i \leq I}$ to find the minimizer of (7.5). Instead, we show in the next section that we can jointly learn the kernel \mathcal{K}_θ together with a deep network mapping $(\mathbf{u}_i = \mathcal{O}(\mathbf{v}_i))_{i \leq I}$ to $(\tilde{\mathbf{y}}_i)_{i \leq I}$ such that $\mathcal{O}_0(\mathbf{v}) = \sum \mathcal{K}_\theta(\mathbf{v}_i, \mathbf{v})(\tilde{\mathbf{y}}_i)$. Let \mathcal{D} be the set of all datasets $\mathcal{D}_\mathcal{O}$ previously defined. The following original theorem gives the existence of a solution to our meta-learning problem and relates it to the reproducing kernel.

[RKBS representer map] Let \mathcal{B} be a \mathcal{U} -valued RKBS from \mathcal{V} to \mathcal{U} , if for any dataset $\mathcal{D}_\mathcal{O} \in \mathcal{D}$, $\mathcal{L}(\cdot, \mathcal{D}_\mathcal{O})$ is lower semi-continuous, coercive and bounded below, then there exists a function $\mathcal{T} : \mathcal{D} \mapsto \mathcal{B}$ such that $\mathcal{T}(\mathcal{D}_\mathcal{O})$ is a minimizer of equation (3.1). If \mathcal{L} is of the form $\mathcal{L}(\cdot, \mathcal{D}_\mathcal{O}) = \tilde{\mathcal{L}} \circ \{\delta_{\mathbf{v}_i}\}_{i \leq n}$ with $\tilde{\mathcal{L}} : \mathcal{U}^n \mapsto \mathbb{R}$, then the dual $\mathcal{T}(\mathcal{D}_\mathcal{O})^*$ is in $\overline{\text{span}}\{\mathcal{K}(\mathbf{v}_i, \cdot)(\mathbf{u})^*, i \leq n, \mathbf{u} \in \mathcal{U}\}$. Furthermore, if for any $\mathcal{D}_\mathcal{O}$, $\mathcal{L}(\cdot, \mathcal{D}_\mathcal{O})$ is strictly-convex, then \mathcal{T} is unique.

3.5 Transducer

3.5.1 Discretization

We now focus for the rest of the paper on the case of infinite-dimensional functional input and output spaces \mathcal{V} and \mathcal{U} . In this case, for numerical computation purpose, we can accommodate different types of function representations previously proposed for neural operator regression and allowing for evaluation at an arbitrary point of the domain. For instance, output function representations can be defined as a linear combination of learned or hardcoded finite set of functions, as in [Lu u. a. \(2019\)](#); [Bhattacharya u. a. \(2020\)](#). We focus instead on a different approach inspired by Fourier Neural Operators ([Li u. a., 2020a](#)), by applying our model on the M first modes of the Fourier transform of functions $(\mathbf{v}_i, \mathbf{u}_i)_{i \leq I}$, and transform back its output, allowing us to work with discrete and finite

function representations.

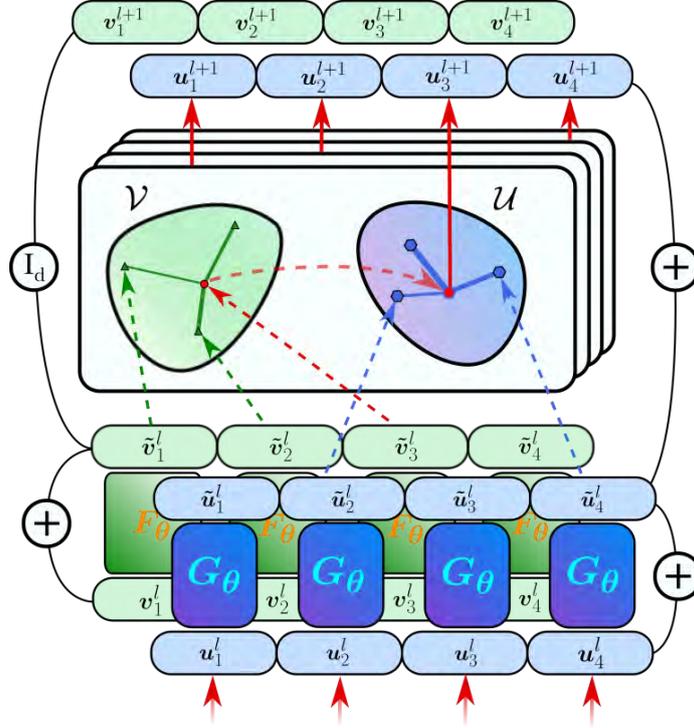


Figure 3.1 – Schema of a *Transducer* layer. A collection of input and output elements $(\mathbf{v}_i^l, \mathbf{u}_i^l)$ are transformed in parallel by one-layer fully connected residual networks into intermediate representations $(\tilde{\mathbf{v}}_i^l, \tilde{\mathbf{u}}_i^l)$. Every element $(\tilde{\mathbf{u}}_j^l)$ is further transformed by the kernel operation \mathcal{K}_{θ_i} which extract relations between $(\tilde{\mathbf{v}}_i^l)$ to form the residual contribution $\sum_i \kappa_{\theta}^l(\tilde{\mathbf{v}}_i^l, \tilde{\mathbf{v}}_j^l)(\tilde{\mathbf{u}}_i^l)$.

Our kernel model consists in a deep neural network of depth L building a sequence of representations $(\mathbf{v}_i^\ell, \mathbf{u}_i^\ell)_{i \leq I, \ell \leq L}$ starting with $(\mathbf{v}_i^1, \mathbf{u}_i^1)_{i \leq I} = \mathcal{E}_O$ and defined by the following iteration:

$$\begin{cases} \mathbf{v}_i^{\ell+1} = \mathbf{v}_i^\ell + F_\theta^\ell(\mathbf{v}_i^\ell) \\ \mathbf{u}_i^{\ell+1} = \mathbf{u}_i^\ell + G_\theta^\ell(\mathbf{u}_i^\ell) + \mathcal{T}_\theta^\ell(\mathbf{v}_i^\ell + F_\theta^\ell(\mathbf{v}_i^\ell)) \end{cases} \quad (3.8)$$

where F_θ^ℓ (resp. G_θ^ℓ) corresponds to pointwise feedforward networks transformations applied in parallel to representations $(\mathbf{v}_i^\ell)_{i \leq I}$ (resp. $(\mathbf{u}_i^\ell)_{i \leq I}$) and $\mathcal{T}_\theta^\ell : \mathcal{V} \mapsto \mathcal{U}$ are intermediate kernel transformations :

$$\forall \mathbf{v} \in \mathcal{V}, \quad \mathcal{T}_\theta^\ell(\mathbf{v}) = \sum_{i \leq I} \kappa_{\theta}^l(\tilde{\mathbf{v}}_i^\ell, \mathbf{v})(\tilde{\mathbf{u}}_i^\ell) \quad (3.9)$$

where $(\tilde{\mathbf{v}}_i^\ell, \tilde{\mathbf{u}}_i^\ell)_{i \leq I}$ refers to the residual transformations $(F_\theta^\ell(\mathbf{v}_i^\ell) + \mathbf{v}_i^\ell, G_\theta^\ell(\mathbf{u}_i^\ell) + \mathbf{u}_i^\ell)_{i \leq I}$. Note that equations (3.8) and (3.9) allows to handle various set sizes. We refer to our system as a *Transducer*, both as a tribute to the Transformer (Vaswani u. a., 2017) computation mechanism from which it is derived and by analogy with engineering devices that convert a signal from one form to another. Our intuition is that network depth allows for a better kernel approximation by iteratively refining the operator estimate and approximate on-the-fly the solution functions $(\tilde{\mathbf{y}}_i)_{i \leq I}$. We investigate further the influence of network depth and its parameterization in the experimental section, noting for instance that sharing the non-linear transformations (F^ℓ, G^ℓ) for the input and output domains can be beneficial to regression performance. Finally, the operator image estimate of any $\mathbf{v} \in \mathcal{V}$ corresponds to the sum of residual transformations:

$$\hat{\mathcal{O}}(\mathbf{v}) = \mathcal{T}_\theta(\mathbf{v} | \mathcal{E}_\mathcal{O}) = \sum_{\ell \leq L} \sum_{i \leq I} \kappa_\theta^\ell(\tilde{\mathbf{v}}_i^\ell, \mathbf{v}^\ell)(\tilde{\mathbf{u}}_i^\ell) \quad (3.10)$$

where the sequence (\mathbf{v}^ℓ) is constructed in the same way as $(\mathbf{v}_i^\ell)_{i \leq I}$ starting with $\mathbf{v}^1 = \mathbf{v}$. Note that the model allows for jointly building sequence (\mathbf{v}^ℓ) with $(\mathbf{v}_i^\ell)_{i \leq I}$, as well as for batch inference by simply masking the corresponding cross-relational features during the kernel operations. Further, all those operations correspond to parallelizable tensorial GPU-accelerated operations such that regression estimates are produced orders of magnitude faster than gradient-based regression methods.

3.5.2 Training

Our goal is to learn the sequence of kernel operations $(\kappa_\theta^\ell)_{\ell \leq L}$ as well as $(F_\theta^\ell, G_\theta^\ell)_{\ell \leq L}$ in order to approximate solutions of (7.5). Let us assume that elements \mathcal{O} are sampled according to a probability measure β , and that for each \mathcal{O} , $\mathcal{E}_\mathcal{O}$ is sampled according to a probability measure $\gamma_\mathcal{O}$ from the set of possible example sets with finite cardinality $E_\mathcal{O}$. Our meta-learning objective can be defined as:

$$\mathcal{J}(\theta) = \mathbb{E}_{\mathcal{O} \sim \beta} \left[\mathbb{E}_{\mathcal{E}_\mathcal{O} \sim \gamma_\mathcal{O}} \left[\|\mathcal{T}_\theta^L(\cdot | \mathcal{E}_\mathcal{O}) - \mathcal{O}\|_{\mathcal{B}} \right] \right] \quad (3.11)$$

that can be tackled with gradient-based optimization given that \mathcal{T}_θ^L is differentiable w.r.t parameters θ . In order to estimate gradient of (3.11), we form the empirical Monte-Carlo estimator over a batch of N operators \mathcal{O} and J query function \mathbf{v}_j :

$$\nabla_\theta \mathcal{J}(\theta) \approx \frac{1}{NJ} \sum_{n \leq N} \sum_{j \leq J} \nabla_\theta \|\mathcal{T}_\theta(\mathbf{v}_j | \mathcal{E}_{\mathcal{O}_n}) - \mathbf{u}_j^n\|_{\mathcal{U}} \quad (3.12)$$

with $\mathbf{u}_j^n = \mathcal{O}_n(\mathbf{v}_j)$. In practice, using a single operator ($N=1$) to estimate (3.12) works well and training is very stable, mitigating the linearly increasing cost of storing the I examples pairs in memory. We show empirically that such meta-optimized models are able to efficiently approximate diverse sets of operators \mathcal{O} given small amounts of data in a single feed-forward pass, bypassing the need for gradient-based tuning. This ability can greatly improve neural-based operators applicability since real physical systems data is often scarce and canonical training approaches require large amounts of it. Furthermore, to the best of our knowledge, this work is the first to show that neural networks can learn transductive regression programs on space of operators.

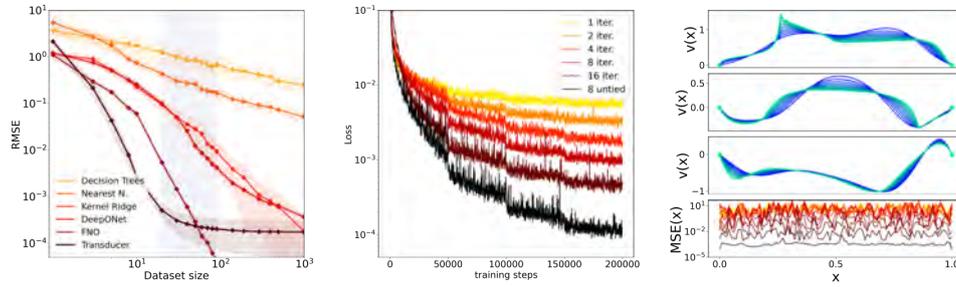


Figure 3.2 – **Left:** Mean relative MSE with a confidence interval at 90% on unseen operators as a function of the dataset size for each regression method. The grey area corresponds to dataset cardinalities seen during training for the *Transducer*. All other methods are trained from scratch with a corresponding number of original examples. **Middle:** Evolution of the training loss of *Transducers* with different number of kernel transformations. Allowing more iterations of kernel transformation improves performance, with untied weights yielding the best performance. **Right:** Up - 3 examples of the evolution of the state $s(x, t)$ as a function of spatial dimension for different ADR equations $E(\boldsymbol{\delta}, \boldsymbol{\nu}, \mathbf{k})$. Bottom - Evolution of the MSE of intermediate representations (\mathbf{u}^ℓ) colored by iteration number. The decreasing profile along with the performance trend observed with a growing number of iterations suggest network depth allows the model to progressively refine its function estimate.

Discretization – As mentioned in the main text, in order to manipulate functional data, we apply our model on the Fourier transforms of the considered input and output functions and transform its output back to form estimate at arbitrary resolution. We specifically apply our model on the d -dimensional finite vector formed by the first modes of the Fourier

transform, and discarding the rest of the function spectrum. For experiments with 2D fields, we describe more precisely in section 3.8.2 how we combine the 2D FFT with our model.

Kernel definition – In order to reduce the number of trainable parameters per layer we split the linear projections defined in the kernel formulation κ_θ into different concatenated "heads" similarly to the Transformer (Vaswani u. a., 2017). We also tested a different kernel definition with a radial basis function $f(\mathbf{v}, \mathbf{v}') = \exp(\alpha^{-1} \|\mathbf{v} - \mathbf{v}'\|)$ replacing the dot product-cosine similarity of the key-query attention, yielding similar result in the ADR experiment at an equal compute cost. For coherence, we present all our results with the original kernel definition.

Feedforward network definition For F_θ^ℓ and G_θ^ℓ , we also directly use the Transformer feedforward network architecture defined as one layer perceptron with GeLU activation and Layer normalization (Ba u. a., 2016) and did not performed architectural search on this part of the network.

In the two following section, we show that our model can perform online regression of various functions, in particular operators defined on scalar and vector-valued function spaces arising in physical science.

Training details – We construct our network by merely stacking Transducer layers defined as above with one-layer perceptrons with layer normalization (Ba u. a., 2016) and GeLU activation functions (Hendrycks und Gimpel, 2016) for (F^ℓ, G^ℓ) and followed by the kernel operation. We carry out meta-learning of operator regression in the following way: We gather a meta-dataset of N_{train} operators example sets $\mathcal{E}_{\mathcal{O}_n}$ with i example function pairs. At each training step, we randomly sample a set $\mathcal{E}_{\mathcal{O}_n}$ as well as query subset \mathcal{Q} of output functions $(\mathcal{O}(\mathbf{v}_i))_{i \in \mathcal{Q}}$ to be estimated. We form the input $\tilde{\mathcal{E}}_{\mathcal{O}_n}$ by concatenating pairs of a non-overlapping set \mathcal{I} of example elements $(\mathbf{v}_i, \mathbf{u}_i)_{i \in \mathcal{I}}$ with pairs of the query set $(\mathbf{v}_i, \mathbf{u}_i)_{i \in \mathcal{Q}}$ with output elements $(\mathbf{u}_i)_{i \in \mathcal{Q}}$ set to zero. We train our model to minimize the sum of L_2 error between each output function of the set \mathcal{Q} and its corresponding ground truth $\mathbf{u}(x) = \mathcal{O}_n(\mathbf{v})(x) = \mathbf{s}(x, 1)$. We use the Adam optimizer (Kingma und Ba, 2014) to train for a fixed number of steps with an initial learning rate of $2e - 4$ that is gradually halved along the training. All the computation is carried on a single Nvidia Titan Xp GPU with 12GB memory. Further details on data generation and baseline settings are given in SI.

3.6 Applications to finite-dimensional meta-learning

3.6.1 MNIST-like datasets classification

We first confirm the generality of our approach in the case of finite-dimensional spaces \mathcal{U} and \mathcal{V} by studying the meta-learning problem presented in ? which consists in regressing classification functions from the 784-dimensional space of MNIST-like images to a 10-dimensional space of one-hot class encoding. (i.e functions considered are $\mathcal{O} : [0, 1]^{784} \mapsto [0, 1]^{10}$). We meta-train a 2-layer Transducer to classify consistently pixel-permuted and class-permuted versions of MNIST. We then meta-test the Transducer to classify the unpermuted MNIST dataset and how the regression map transfer to Fashion MNIST and KMNIST. We show that without particular fine-tuning, the Transducer outperforms previous meta-learning approaches on both the original MNIST classification task as well as Fashion MNIST and K-MNIST.

METHOD	MNIST	FASHIONMNIST	KMNIST
MAML	53.71%	48.44%	36.33%
VSML	79.04%	68.49%	54.69%
GPICL	73.70 %	62.24%	53.39%
TRANSDUCER	81.83%	69.85%	60.64%

Figure 3.3 – Comparison of meta-test accuracies of MNIST-like datasets classification task presented in ? against the Transducer.

3.7 Applications to ∞ -dimensional problems

3.7.1 Regression of Advection-Diffusion Reaction PDEs

First, we examine the problem of regressing operators \mathcal{O} associating functions \mathbf{v} from $\mathcal{V} \subset C([0, 1], \mathbb{R})$ to their future solution $\mathbf{u} = \mathcal{O}(\mathbf{v}) \subset C([0, 1], \mathbb{R})$ with respect to advection-diffusion-reaction equations defined on the domain $\Omega = [0, 1] \times [0, t]$ with Dirichlet boundary conditions $\mathbf{s}(0, t) = \mathbf{s}(1, t) = 0$. We consider the space \mathcal{B} of operators $\mathcal{O}_{(\boldsymbol{\delta}, \boldsymbol{\nu}, \mathbf{k}, t)}$ specifically defined by the the 4-uplet $(\boldsymbol{\delta}, \boldsymbol{\nu}, \mathbf{k}, t)$, such that $\mathbf{v}(x) = \mathbf{s}(x, 0)$, $\mathbf{u}(x) = \mathbf{s}(x, t)$ and \mathbf{s} follows equation $E(\boldsymbol{\delta}, \boldsymbol{\nu}, \mathbf{k})$ that depends on unknown random continuous spatially-varying diffusion $\boldsymbol{\delta}(x)$, advection $\boldsymbol{\nu}(x)$,

and a scalar reaction term $\mathbf{k} \sim \mathcal{U}[0, 0.1]$.

$$\begin{aligned} \partial_t \mathbf{s}(x, t) = & \underbrace{\nabla \cdot (\boldsymbol{\delta}(x) \nabla_x \mathbf{s}(x, t))}_{\text{diffusion}} + \underbrace{\boldsymbol{\nu}(x) \nabla_x \mathbf{s}(x, t)}_{\text{advection}} \\ & + \underbrace{\mathbf{k} \cdot (\mathbf{s}(x, t))^2}_{\text{reaction}} \end{aligned} \quad (3.13)$$

Eq. (3.13) is generic and arises in many physical systems of interest, leading to various forms of solutions $\mathbf{s}(x, t)$. (We show examples for different sampled operators in figure 7.2.) Several methods exist for modeling such PDEs but they require knowledge of the underlying parameters $(\boldsymbol{\delta}, \boldsymbol{\nu}, \mathbf{k})$ and often impose constraints on the evaluation point as well as expansive time-marching schemes to recover solutions. Here instead, we assume no *a priori* knowledge of the solution and directly regress the operator behavior from the example set $\mathcal{E}_{\mathcal{O}}$.

Baselines and evaluation We trained our neural regression model on $N_{train} = 500$ different operators $\mathcal{O}_{(\boldsymbol{\delta}, \boldsymbol{\nu}, \mathbf{k}, 1)}$ with $t = 1$ fixed and varying number of examples $n \in [20, 100]$ each evaluated at 100 equally spaced points $(x_k)_{k \in [0, 100]}$ on the domain $[0, 1]$. We tested on a set of $N_{test} = 500$ new operators with new diffusion, advection, and reaction parameters as well as initial states \mathbf{v} . We compared our method with standard machine learning algorithms as well as inductive neural operator approximators. First, we applied standard finite-dimensional regression methods to the discretized problems: $(\mathcal{O}_i(\mathbf{v}(x_k)) = \mathbf{u}(x_k))_{i, k}$. We apply K -nearest-neighbors (Fix und Hodges, 1989), decision trees (Quinlan, 1986) and kernelized Ridge regression with radial basis kernel (Hastie u. a., 2009). Second, we tried to fit two neural-based operators with gradient-descent to the small-sized datasets: DeepONet (Lu u. a., 2021) and Fourier Neural operator (FNO) (Li u. a., 2020a). Note that for most of these approaches, an explicit optimization scheme is required before inference to fit the target operator. On the other hand, after meta-training, the *Transducer* program amounts to a single feedforward pass of the network, which is orders of magnitude faster (Table 3.1).

Results First, we verified that our model approximates well unseen operators from the test set (table 3.1). Furthermore, since our model can perform inference for varying input set sizes, we examined the *Transducer* accuracy when varying the number of examples and found that it learns a converging regression program (Figure 7.2). We noted that our model learns a non-trivial kernel since the linear estimation produced with Nearest Neighbors remains poor even after $1e^3$ examples. Further, our model

METHOD	RMSE	TIME (s)	GFLOPs
FNO	$2.96e^{-4}$	$1.72e^2$	$1.68e^2$
DEEPONET	$2.02e^{-2}$	$7.85e^1$	$1.54e^2$
TRANSDUCER	$2.39e^{-4}$	$3.10e^{-3}$	$1.06e^{-1}$

Table 3.1 – Mean performance and compute cost of neural operator regression over 50 unseen operators with $n = 50$ examples. Note that DeepONet and FNO are optimized from scratch while the *Transducer* has been pre-trained. GFLOPs represent the total number of floating point (giga-)operations for performing regression.

consistently outperforms other regression approaches to the exception of FNO when enough data is available (> 60). We also found that deeper *Transducer* models with more iterations increase kernel approximation accuracy, with untied weights yielding best performance (figure 7.2.)

Extrapolation We further tested the *Transducer* ability to regress different operators than those defined in the training set. Specifically, we varied the correlation length (C.L) of the gaussian random fields used to generate functions $\delta(x)$ and $\nu(x)$ in the domain $[0.1, 0.5]^2$ and specified a different target time $t' \neq 1$. We showed that the kernel meta-optimized for a solution at $t = 1$ transfers well to these new regression problems and that regression performance degrades gracefully as the target operators behave further away from the training set. (figure 3.4)

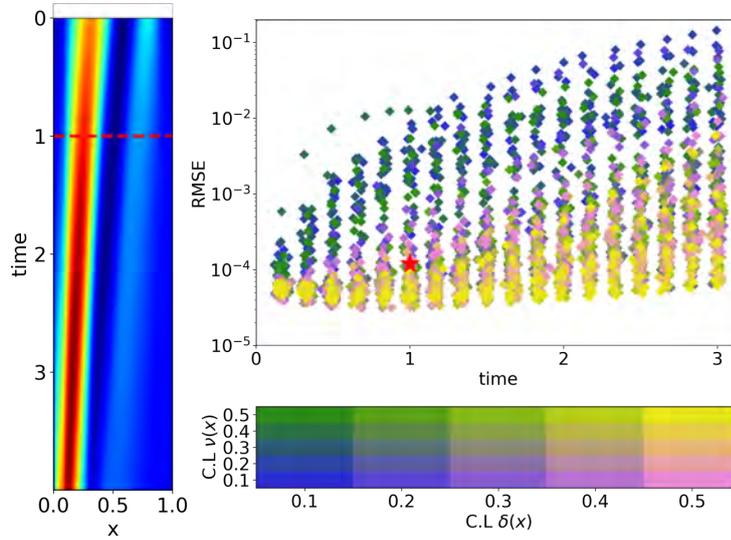


Figure 3.4 – Example of the extrapolated trajectory and Relative Mean Squared Error of the *Transducer* on extrapolation tasks with $n = 100$ examples. Color code corresponds to different correlation lengths used to generate the random functions $\delta(x)$ and $\nu(x)$. Much of the result remains below 1% error despite never being trained on such operators.

3.7.2 Burger’s equation in 2D

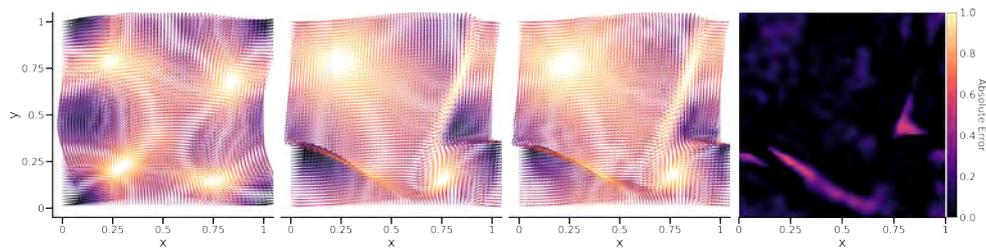


Figure 3.5 – Illustrative example of initial ($t = 0$), target ($t = 10$) and *Transducer* estimation of the vector field $s(\vec{x}, t)$ discretized at resolution 64×64 over the domain $[0, 1]^2$ for the Burger’s equation experiment. The last panel represents absolute error to ground truth.

We further show that our regression method can fit operators of vector-valued functions. We examine the problem of predicting 2D vector-fields defined as a solution of a two-dimensional Burger’s equation with periodic

spatial boundary condition on the domain $\Omega = [0, 1]^2 \times [0, 10]$:

$$\partial_t \mathbf{s}(\vec{\mathbf{v}}, t) = \underbrace{\boldsymbol{\nu} \Delta_{\mathbf{v}} \cdot \mathbf{s}(\vec{\mathbf{v}}, t)}_{\text{diffusion}} - \underbrace{\mathbf{s}(\vec{\mathbf{v}}, t) \nabla_{\mathbf{x}} \mathbf{s}(\vec{\mathbf{v}}, t)}_{\text{advection}} \quad (3.14)$$

Here, we condition our model with operators of the form, $\mathbf{v}(\vec{\mathbf{x}}) = \mathbf{s}(\vec{\mathbf{x}}, t)$, $\mathbf{u}(\vec{\mathbf{x}}) = \mathbf{s}(\vec{\mathbf{x}}, t')$, $\mathbf{s} \sim E(\boldsymbol{\nu})$ such that our model can regress the evolution of the vector field $\vec{\mathbf{v}}$ starting at any time, with arbitrary temporal increment $t' - t \leq 10$ seconds and varying diffusion coefficient $\boldsymbol{\nu} \in [0.1, 0.5]$. We show in figure 3.5 and table ?? that our model is able to fit new instances of this problem with unseen parameters $\boldsymbol{\nu}$.

Fast and differentiable regression \diamond Since fitting with respect to training data is orders of magnitude faster than other operator regression approaches and fully differentiable, we can quickly execute expensive schemes requiring multiple regressions. This can have several applications, from bootstrapping or producing confidence intervals by varying the example set $\mathcal{D}_{\mathcal{O}}^{\text{train}}$, or performing inverse problems using Monte-Carlo Markov Chain in the dataset space. We showcase an example of this potential with an outlier detection experiment: We use the *Transducer* to identify outliers of a dataset of Burgers' equation with coefficient $\boldsymbol{\nu}_1$ artificially contaminated with elements from another dataset $\boldsymbol{\nu}_2 > \boldsymbol{\nu}_1$ at 5% level. We identify outliers by estimating RMSEs over 5000 different regressions using random 50 % splits with outliers potentially present in both training and testing sets. This technique takes only a few seconds to estimate while outliers are clearly identified as data points with significantly higher RMSE than the dataset average (figure 5). As a comparison, performing Spectral Clustering (?) on the FFT of elements (\mathbf{u}_i) yields very poor precision (table 2)

	$t = 5\text{s}$	$t = 10\text{s}$
RMSE (test sets)	$2.2e^{-3}$	$5.9e^{-3}$
Outliers (Pre./Rec.)	100%/100%	100%/100%
S.C. (Pre./Rec.)	6%/85%	7%/85%

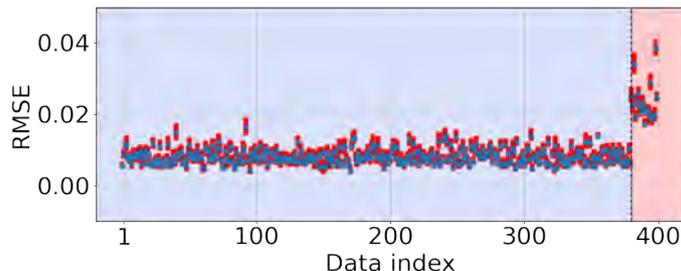


Figure 3.6 – **Up**: Meta-test regression and outlier detection results at two target times. RMSEs on Burgers’ equations averaged over 200 different parameter conditions $\nu \in [0.1, 0.5]$ each with 100 train examples. Precision/Recall in outlier detection of the Transducer versus Spectral clustering. **Bottom**: RMSE distributions of each element in the contaminated dataset over the 5000 regressions. Outliers are clearly identified.

3.7.3 Climate modeling

Robust and precise modeling of climate variables is difficult because models need to account for seasonal variables and adapt their prediction to drifting parameters. Even with a globally large amount of available data, the precise operator mapping variables of interest might change over time or be affected by the unobserved phenomenon. Hence, in order to fully exploit the potential of data-driven methods, being able to capture variations in how variables interact might greatly help prediction performance even under operator fluctuation and on data drifting away from the training distribution. One advantage of our approach is the ability to select the data that is most relevant with respect to a certain prediction task and subsequently adapt the model response. In order to illustrate the applicability and scalability of deep transductive learning, we considered the problem of predicting the Earth’s surface air pressure solely from the Earth’s surface air temperature at a high resolution. Data is taken from the ERA5 reanalysis (Hersbach u. a., 2020) publicly made available by the ECMWF, which consists of hourly high-resolution estimates of multiple atmospheric variables from 1979 to the current day. We model pressure estimate on a 720×720 grid, resulting in a spatial resolution of $0.25^\circ \times 0.5^\circ$, allowing us to capture small features such as local dynamics

METHOD	LWMSE (hPa)	TIME (s)
NEAREST-NEIGHBORS	67.326	5.915
ViT	32.826	0.053
TRANSDUCER - (P.Y)	25.293	0.192
TRANSDUCER - (P.W)	22.718	0.192

Table 3.2 – Latitude-weighted mean-square error (in hectopascals) and inference time for the surface pressure prediction task.

and geographic relief. Similar to (Pathak u. a., 2022), we modify a ViT backbone to incorporate a kernel transduction layer before every patch attention and compare our model to an unmodified ViT baseline with a matching number of parameters. We additionally compare with a fully transductive Nearest Neighbors approach. In Figure 3.7 and Table 3.2, we present results obtained on training a Transducer with data from 2010 to 2014 and testing it on data from 2016 to 2019. We trained our model by predicting 5 random days sampled from random 20-day windows and present two test configurations: We either condition the *Transducer* with a window centered at the previous year’s same date (P.Y), or with a 15 days window lagging by a week (P.W). (see SI for details). Both cases outperform transductive and inductive baselines with fast inference time, confirming that our model can scale to large problems.

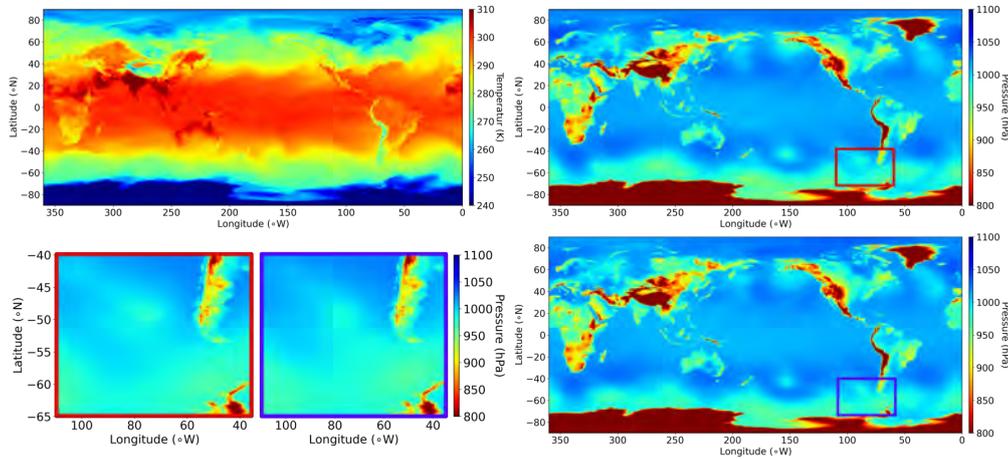


Figure 3.7 – *Up* - Illustrative examples of 720×720 temperature (*left*) and pressure (*right*) fields of the ERA5 dataset. *Bottom* - Estimated pressure field from conditioning the *Transducer* with 15 days data dating 1 week before the target date. Insets show recovered details of the *Transducer* estimation (*blue*) compared with ground truth (*red*).

3.8 Experimental details

EXPERIMENT	DEPTH	MLP DIM	DIM d	#HEADS	HEADS
ADR	1-8	100	50	32	16
BURGER	10	800	800	64	16
CLIMATE	6	512	512	40	16

Table 3.3 – Summary of the architectural hyperparameters used to build the *Transducer* in our three experiments. 'DEPTH' corresponds to network number of layers, 'MLP dim' to the dimensionality of the hidden layer representation in F_θ^ℓ and G_θ^ℓ , d to the dimension of the discrete function representations.

3.8.1 Advection-Diffusion-Reaction operators

Data generation – For our experiment, we collect a meta-dataset of $N = 500$ datasets of the advection-diffusion-reaction trajectories on the domain $\Omega = [0, 1] \times [0, 1]$ by integrating the following equations:

$$\partial_t \mathbf{s}(x, t) = \underbrace{\nabla \cdot (\boldsymbol{\delta}_n(x) \nabla_x \mathbf{s}(x, t))}_{\text{diffusion}} + \underbrace{\boldsymbol{\nu}_n(x) \nabla_x \mathbf{s}(x, t)}_{\text{advection}} + \underbrace{\mathbf{k}_n \cdot (\mathbf{s}(x, t))^2}_{\text{reaction}} \quad (3.15)$$

using an explicit forward Euler method with step-size $1e^{-2}$ and storing all intermediate solutions on a spatial mesh of 100 equally spaced points. Hence, our discretized reference trajectories are of dimensions 100×100 . For each operator \mathcal{O}_n we generate spatially varying diffusion and advection coefficients as random function $\boldsymbol{\delta}_n(x) : [0, 1] \mapsto \mathbb{R}$ and $\boldsymbol{\nu}_n(x) : [0, 1] \mapsto \mathbb{R}$ as well as a random scalar reaction coefficient \mathbf{k}_n . Defining $\mathcal{G}(0, k_l(x1, x2))$ the one-dimensional zero-mean Gaussian random field with the covariance kernel:

$$k_l(x1, x2) = e^{-\frac{\|x1-x2\|^2}{2l^2}} \quad (3.16)$$

and length-scale parameter $l = 0.2$, as well as a boundary mask function $m : [0, 1] \mapsto [0, 1], m(x) = 1 - (2x - 1)^{10}$ (to comply with Dirichlet boundary condition and preserve numerical computation stability), we sample $\boldsymbol{\delta}_n(x)$ and $\boldsymbol{\nu}_n(x)$ according to the following equations:

- **diffusion** $\boldsymbol{\delta}_n(x) = 0.01 \times u_n(x)^2 \times m(x)$ where $u_n \sim \mathcal{G}(0, k_{0.2}(x1, x2))$
- **advection** $\boldsymbol{\nu}_n(x) = 0.05 \times y_n(x) \times m(x)$ where $y_n \sim \mathcal{G}(0, k_{0.2}(x1, x2))$

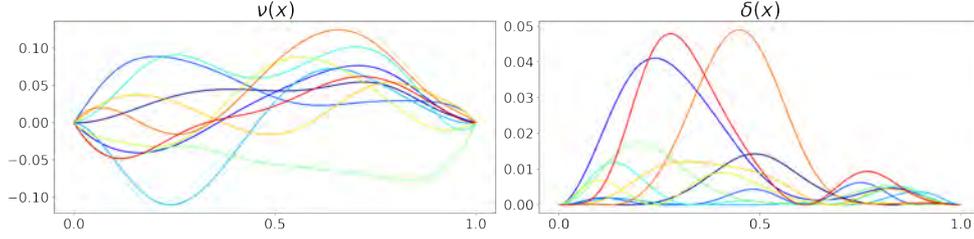


Figure 3.8 – Examples of sampled functions $\delta(x)$ and $\nu(x)$ used to build operators \mathcal{O}_n .

- **reaction** $k_n \sim \mathcal{U}([0, 0.3])$.

Furthermore, we collect for each dataset $i = 100$ trajectories with each different initial state $s(x, 0) = \mathbf{v}_i(x)$, where functions $\mathbf{v}_i(x)$ are sampled according to the following:

- **initial state** $\mathbf{v}_i(x) = m(x) \times u_i(x)$ where $u_i \sim \mathcal{G}(0, k_{0.2}(x1, x2))$.

For meta-testing, we sample $N = 500$ new datasets of the same generic advection-diffusion-reaction equation with new parameters $\delta_n(x), \nu_n(x), k_n(x)$, for up to 1000 different initial states $\mathbf{v}_i(x)$. We present below example of function profiles present in the meta-datasets.

Training – We train Tranducers for 200K gradient steps. At each training step, we randomly draw a single operator \mathcal{O}_n from the meta-training set and isolate the pairs $(\mathbf{v}_i, \mathbf{u}_i)_{i \leq I} = (s_i(x, 0), s_i(x, 1))_{i \leq I}$ to form the set $\mathcal{E}_{\mathcal{O}_n}$. We sample a "query" subset \mathcal{Q} of $J = 10$ pairs from $\mathcal{E}_{\mathcal{O}_n}$ to be regressed and form the input to our model by concatenating pairs of the query set \mathcal{Q} (with output elements $(\mathbf{u}_i)_{i \in \mathcal{Q}}$ set to zero), with a non-overlapping set of $I \in \llbracket 20, 100 \rrbracket$ example elements drawn from $(\mathbf{v}_i, \mathbf{u}_i)_{i \notin \mathcal{Q}}$. We train our model to minimize the sum of L_2 error between each output function of the set \mathcal{Q} and its corresponding ground truth $\mathbf{u}(x) = \mathcal{O}_n(\mathbf{v})(x) = \mathbf{s}(x, 1)$ at the 100 discretized positions. We use the Adam optimizer (Kingma und Ba, 2014) with an initial learning rate of $2e - 4$ that is further halved every 50K steps. All the computation operations are coded using the PyTorch library and carried on a single Nvidia Titan Xp GPU with 12GB memory.

Baselines – In order to implement the baseline regression algorithms, we use the scikit-learn library (Pedregosa u. a., 2011) for decisions trees, K -nearest neighbours and Ridge regression. We specifically tuned Ridge regression using cross-validation and selected the best-performing 'RBF'

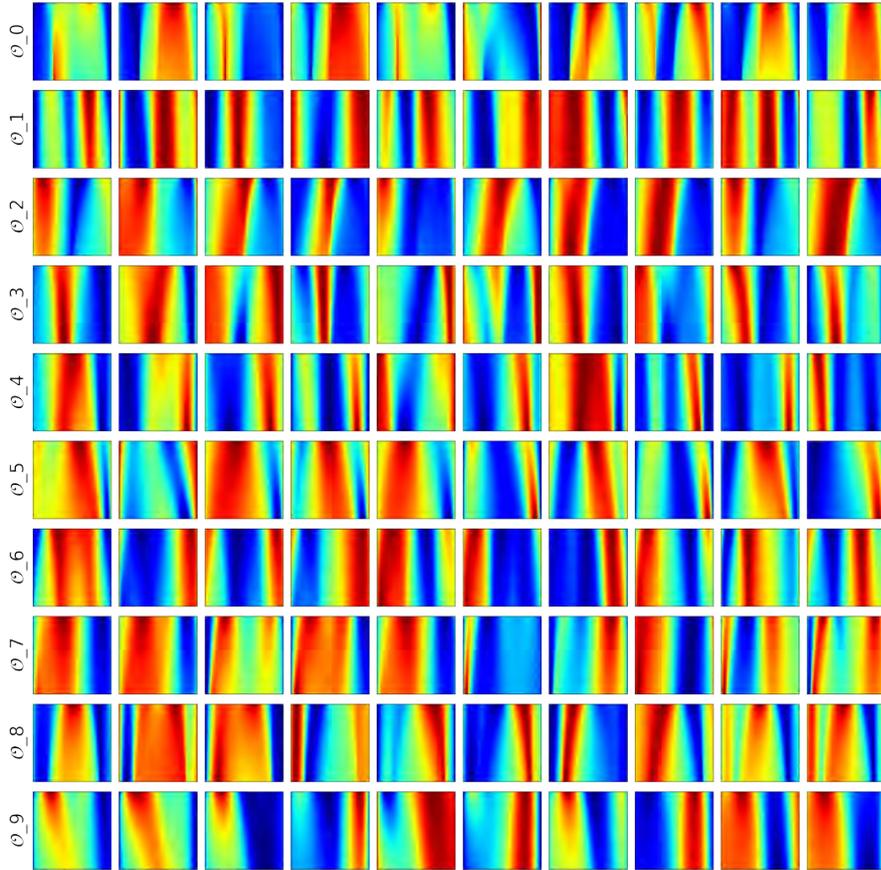


Figure 3.9 – Examples of advection-diffusion-reaction datasets (different operators by row) present in the meta-test set.

kernel with regularisation $\lambda = 1e^{-3}$. For FNO (Li u. a., 2020a), we use the official PyTorch implementation provided at [dsd](#) and defined for each regression, a 4-layer deep 1-dimensional FNO network with 16 modes and 64-dimensional 1×1 convolutions. For DeepOnet (Lu u. a., 2019), we implement our own PyTorch version with 4 hidden layers of 50 hidden units with ReLU activation for the branch and trunk networks.

Extrapolation experiment – In this task, we modify the generative process of the considered operators by changing the length-scale parameter l used to produce functions $\delta(x)$ and $\nu(x)$, as well as the target time t used to define the operator output.



Figure 3.10 – Examples of the spatial function sampled with carrying length scale parameter $l \in [0.1, 0.2, 0.3]$

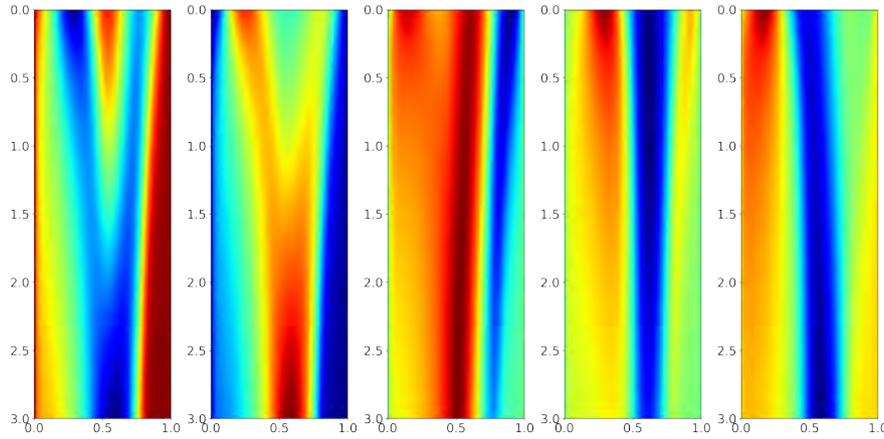


Figure 3.11 – Examples of ADR state evolution forming a set of operators with the same generative parameters but time t allowed to vary in $[0, 3]$

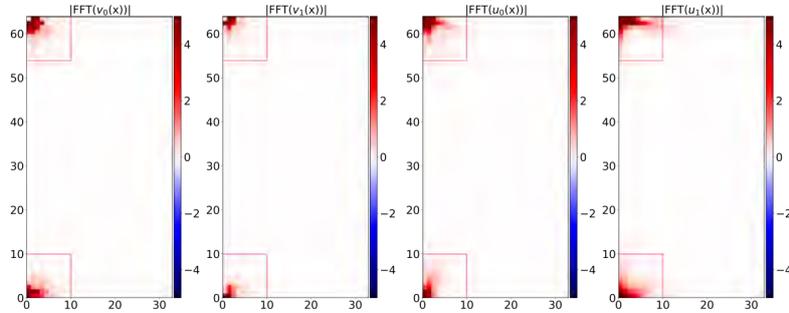


Figure 3.12 – Magnitude of the complex coefficients of the Fourier transform of an exemple pair of input and output functions $(\mathbf{v}(\vec{x}), \mathbf{u}(\vec{x}))$ in the two coordinates dimension. For every pair, the majority of the signal lies in the two the red quadrants.

3.8.2 Burger’s equation

Generation – In order to produce the meta-datasets of our second experiment, we use the Φ Flow library (Holl u. a.) that allows for batched and differentiable simulations of fluid dynamics and available at <https://>

[//github.com/tum-pbs/PhiFlow](https://github.com/tum-pbs/PhiFlow). Following the same methodology as experiment 1, we generate batches of the state evolution of random functions $(\mathbf{v}_i) : \mathbb{R}^2 \mapsto \mathbb{R}^2$ defined on the domain $\Omega = [0, 1]^2$ at a resolution of 64×64 through different parametrization of equation (3.5). We form a meta training set of 200 operator datasets for different parameters $\nu \in [0.1, 0.5]$ each of cardinality $I = 100$, and meta testing set of 200 different operator datasets with the same cardinality. Here, we consider vector fields input functions $\mathbf{v}(\vec{x})$ whose coordinates $(\mathbf{v}_1(\vec{x}), \mathbf{v}_2(\vec{x}))$ are drawn each from a two-dimensional zero-mean Gaussian random fields with uniform exponential covariance function and correlation length $l = 0.125$.

Discrete Fourier representation – Since we are dealing with high-dimensional inputs, we perform kernel regression on the 2D fast Fourier transforms of our model. To reduce further dimensionality, since the FFT of a real signals is Hermitian-symmetric, we pass as input to our model only the flattened 10×10 upper and lower quadrants of the Fourier transform coefficients, since we verified that those are sufficient to reconstruct the signal at relative error level of $1e - 5$. (We present examples of the 2D FFT of our signal.) After regression, we reconstruct our model estimate in the spatial domain at the desired 64×64 resolution and train for the L_2 distance against ground truth.

3.8.3 Climate modeling

Data – We take our data from ERA5 reanalysis (Hersbach u. a., 2020), that is freely available on the Copernicus <https://cds.climate.copernicus.eu/cdsapp#!/dataset/reanalysis-era5-land?tab=overview>. Surface and temperature pressure are re-gridded from a Gaussian grid to a regular Euclidean grid using the standard interpolation scheme provided by the Copernicus Climate Data Store (CDS) to form 2D fields that we further interpolate in the longitude dimension to obtain images of size 720×720 . Although the ERA5 possess hourly estimates, we subsample the dataset by considering only measurement at 12:00am UTC every day.

Training We train our model with learning rate $\lambda = 1e^{-4}$ halved every 50K steps until 200K steps. As mentioned in the main text, we trained our model to predict 5 days randomly sampled from a 20-day window and do not explore larger settings due to GPU memory constraints.

3.9 Discussion

We proposed a novel transductive regression model combining kernel principles and neural networks. We grounded our model on the theory of

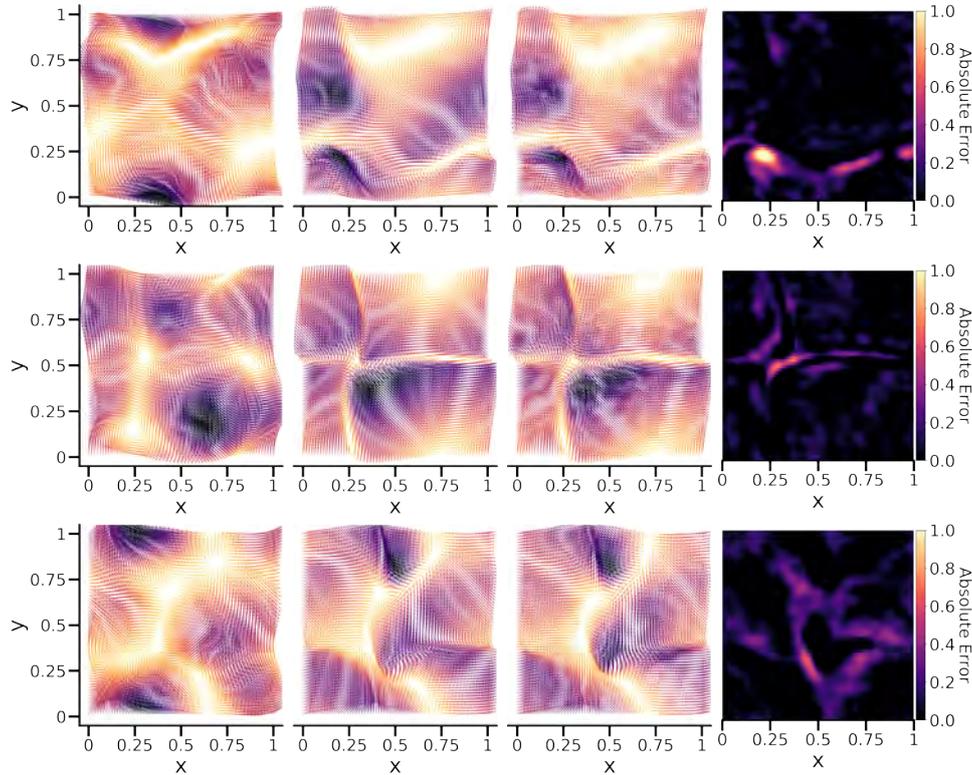


Figure 3.13 – Illustrative examples of initial ($t = 0$), target ($t = 10$) and *Transducer* estimation of the vector field $s(\vec{x}, t)$ discretized at resolution 64×64 over the domain $[0, 1]^2$ for the Burger’s equation experiment. The last panel represents absolute error compared to ground truth.

Reproducing Kernel Banach Spaces and applied it to perform operator regression between function spaces. We showcased several instances where it outperformed previous approaches at a lower computational cost. Our approach holds the potential to yield flexible tools for modeling physical systems. However, one limitation is that it relies on meta-training of the model which requires collecting a sufficiently diverse meta-dataset to explore the kernel space. In future work, we plan to investigate methods such as synthetic augmentation to reduce meta-training costs.

Part II

On Meta-learning

Introduction

The narrow adaptability exhibited by current artificial neural networks comes in contrast to natural intelligence which exhibit a striking ability to self-reflexively accelerate and expand skill acquisition through experience synthesis (Thrun und Pratt, 1998). This ability to *learn how to learn*, has since long been identified by researchers in artificial intelligence aiming at building similar **meta-learning** agents, i.e agents that contains the principles governing their own adaptation (Schmidhuber, 1992; Schmidhuber u. a., 1996; Vilalta und Drissi, 2002; Wang, 2021). In this part, we discuss such an approach and its recent development from the perspective of the **free-energy** principle which has been put forward as a general theory for perceptual inference and learning in natural intelligence (Friston und Stephan, 2007; Friston, 2010). As a generic principle driving multiple scales of biological neural adaptation, we show that the idea of free-energy minimization offers an unifying structure to interpret different approaches in artificial neural meta-learning. We also discuss how such principle can integrate the currently dual notions of variable optimization powering neural network training and Bayesian inference. We also argue that it represent a general inspiration theme for programmatic searches of artificial neural rules of adaptation and learning, that can automate further the discovery of original and robust knowledge promised by the deep learning revolution, as well as potentially providing a better understanding of natural intelligence. We finally present in chapter 5 a new class of neural networks with dynamic weights inspired by this discussion as well as previous results regarding "fast weights" in artificial neural networks and the role of fast synaptic plasticity for working memory. This model is tested in a range reinforcement learning problems, such as robotic control and spatial navigation, where it validates that dynamic weights can provide a standalone mechanism for reinforcement learning when exposed to a feedback signal from its environment, and advocates for a more general exploration of the role fast plasticity in artificial and biological networks.

Chapter 4

Meta-learning and the free-energy principle

“... something fundamentally constructive about the dynamics of things that self-organize or simply exist: they are always moving up the log of the probability gradients that describe the states that they are characteristically in. And when you write that down you basically get the free energy principle.”

Karl Friston

Contents

4.1	The free-energy principle	99
4.2	Current approaches of meta-learning	104
4.3	Potential directions and challenges	106
4.4	Transition	110

4.1 The free-energy principle

A defining characteristic of living things is their homeostatic persistence over time despite being exposed to varying and challenging environmental conditions. Biological agents maintain the integrity of their potentially highly complex structure within physiological boundaries against a natural tendency of the world towards disordered states with high configurational entropy (Prigogine und Nicolis, 1985). At several time-scales, ranging from short individual experience to generational transfer, serving this objective has determined the emergence of complex adaptive behaviors, allowing to capture information and react appropriately to both

internal and external variations of the state of the world, and referred broadly to as *learning*.

In most complex living species endowed with cognitive abilities, learning is manifested during the individual lifetime by a complex hierarchy of inter-related electro-chemical processes happening in the nervous system. We define the sequence of states describing the evolution of these neural processes (that we will describe further below) in a single agent as $\tilde{\mathbf{s}}_t = \{\mathbf{s}_0, \dots, \mathbf{s}_t\}$ which broadly range from connection strength between neurons, neuromodulative quantities, to neuronal activity itself. We additionally define $\tilde{\mathbf{z}}_t = \{\mathbf{z}_0, \dots, \mathbf{z}_t\}$, a sequence of (potentially evolving) generative variables that causes the agent sequence of observations $\tilde{\mathbf{o}}_t = \{\mathbf{o}_0, \dots, \mathbf{o}_t\}$. It has been proposed by Friston (2010) that such processes aim at minimizing a *free-energy* function E fundamentally measuring an "adequation" between $\tilde{\mathbf{o}}_t$ and the internal states $\tilde{\mathbf{s}}_t = \{\mathbf{s}_0, \dots, \mathbf{s}_t\}$ of the neural system :

$$E(\tilde{\mathbf{o}}_t, \tilde{\mathbf{s}}_t) = -\mathbb{E}_{q_{\tilde{\mathbf{s}}_t}(\tilde{\mathbf{z}}_t)} [\ln(p(\tilde{\mathbf{o}}_t, \tilde{\mathbf{z}}_t)) - \ln(q_{\tilde{\mathbf{s}}_t}(\tilde{\mathbf{z}}_t))] \quad (4.1)$$

In (4.1), $q_{\tilde{\mathbf{s}}_t}(\tilde{\mathbf{z}}_t)$ represents the density function over the environmental causes $\tilde{\mathbf{z}}_t$ that is encoded by the system's internal states $\tilde{\mathbf{s}}_t$. Thus, in essence, the *free-energy* E hypothesis assumes that neural variables $\tilde{\mathbf{s}}_t$ encode for a *generative model* of the observed world, i.e a probabilistic model of the dependencies between causes $\tilde{\mathbf{z}}_t$ and consequential observations $\tilde{\mathbf{s}}_t$. Furthermore, the *free-energy* E is an upper bound on the surprise of the collection of observed states $\tilde{\mathbf{s}}_t$ which is defined as the negative log-evidence of observed states $\mathcal{S}(\tilde{\mathbf{o}}_t) = -\ln(\int_{\tilde{\mathbf{z}}} p(\tilde{\mathbf{o}}_t, \tilde{\mathbf{z}}) d\tilde{\mathbf{z}})$ through the following relation:

$$-\int_{\tilde{\mathbf{z}}} q_{\tilde{\mathbf{s}}_t}(\tilde{\mathbf{z}}_t) \ln\left(\frac{p(\tilde{\mathbf{o}}_t, \tilde{\mathbf{z}}_t)}{q_{\tilde{\mathbf{s}}_t}(\tilde{\mathbf{z}}_t)}\right) d\tilde{\mathbf{z}} \underbrace{\geq}_{\text{Jensen}} -\ln\left(\int_{\tilde{\mathbf{z}}} q_{\tilde{\mathbf{s}}_t}(\tilde{\mathbf{z}}_t) \frac{p(\tilde{\mathbf{o}}_t, \tilde{\mathbf{z}}_t)}{q_{\tilde{\mathbf{s}}_t}(\tilde{\mathbf{z}}_t)} d\tilde{\mathbf{z}}\right) = \mathcal{S}(\tilde{\mathbf{o}}_t) \quad (4.2)$$

Thus, minimizing of the *free-energy* in equation (4.1) allows implicitly the agent to decrease the surprise contained in its observations of the world. In order to perform such minimization, agents can decrease the energy by varying its internal neural state $\tilde{\mathbf{s}}_t$ or subsequently vary its sensory information $\tilde{\mathbf{o}}_t$. We note, as stated in Friston (2010), that such variable distinction maps nicely onto the agent action and perception abilities. We will focus on a synthetic view of neural state minimization $\tilde{\mathbf{s}}_t$, as it can indirectly bias the agent policy determining observations $\tilde{\mathbf{o}}_t$.

From a perceptual perspective, the free energy allows to interpret how the minimization framework of a classification or reconstruction objec-

tives of a subset of artificial neurons used to train artificial networks can implicitly drive internal representation learning within the network (Le-cun u. a., 1998). Defining and minimizing heuristic energy functions to force a partial adequation of a subset neurons with external stimuli force implicitly the entire system to adopt explicative or at least correlative structures that capture the statistical features present in external world. Moreover, the free-energy principle is not only useful to hypothesize a generic energy minimization problem performed by biological brains as a surprise minimization, it also nicely account for a nested hierarchy of inter-related neural processes occurring in the brain at different time-scales, each orchestrating the dynamics of downstream systems, and potentially entailing integrative strategies for learning.

4.1.1 Hierarchy of inter-related learning systems

Characterizing the different physiological mechanisms that govern the evolution of the state $\tilde{\mathbf{s}}_t$ of complex neural systems over time as a function of observations $\tilde{\mathbf{o}}_t$ and their own state reveals a nested hierarchy of processes influencing each other along different time scales. Different cognitive and behavioral faculties ranging from direct brain functions such as perception, motor and attentional control or decision making, to more long-lasting effects such as memorization, or preference acquisition have been gradually mapped to such changes, although their precise interaction is still very partially understood. We briefly identify below such processes from fastest to longest time horizon, by partitioning the variables $\tilde{\mathbf{s}}_t$ into neural activity $\tilde{\mathbf{x}}_t$, synaptic connections $\tilde{\boldsymbol{\theta}}_t$ and neural connectivity $\tilde{\mathbf{c}}_t$, and describe how they are hypothesized to support different levels of learning.

At fast time-scales in the range of a few milliseconds, fast signaling processes between neurons (action potentials) can propagate information between different brain regions by changing their electromagnetic states $\tilde{\mathbf{x}}_t$ to encode directly rich contextual representation of the current state of the environment (Hyman u. a., 2012) as well as a probabilistic transition between such states (Alexander und Brown, 2011). For instance, activity levels within the anterior cingulate cortex has been identified to track contextual volatility and uncertainty in a Bayesian manner (Khamassi u. a., 2013). Similarly, activations of specific population in the prefrontal cortex have been shown to elicit a competition between relevant neural functional modes (Lee u. a., 2014; Daw u. a., 2005). Beyond the simple feedforward model usually proposed in deep learning, which goes unidirectionally from lower level sensory areas to higher-level integrative areas,

the precise organization and connectivity in biological brain is forming specific partitions which allow to directly embed rapid feedback control mechanisms over neural dynamics such that the distribution of neural activity can be factored as

$$q_{\tilde{\theta}_t, \tilde{c}_t}(\tilde{\theta}_t, \tilde{\mathbf{x}}_t) = q_{\tilde{\theta}_0, \tilde{c}_0}(\tilde{\theta}_0, \tilde{\mathbf{x}}_0) \prod_{l=1}^t \prod_{i=1}^I q_{\tilde{\theta}_{t-l}, \tilde{c}_{t-l}}(\mathbf{x}_{t-l}^i | \mathbf{x}_{\leq t-l-1}^{\rightarrow i}) \quad (4.3)$$

where $q_{\tilde{\theta}_t, \tilde{c}_t}(\mathbf{x}_{t-l}^i | \mathbf{x}_{\leq t-l-1}^{\rightarrow i})$ represent the conditional distribution of \mathbf{x}_{t-l}^i given history of all afferent inputs. In this context, understanding the functional connectivity of biological network is of paramount importance to isolate clearly how such rapid feedback system can operate and maintain the network free-energy in low states.

Second, the release and absorption of chemical messengers throughout the chain of fast neural transmissions can alter information integration and transfer properties of neurons at a second slower time scale. This variable set encompass phasic and localized modulations such as calcium-dependent effects on synaptic efficacy between neurons or tonic transmissions such as classic neuromodulation from external sources with more diffuse effect (dopaminergic, serotonergic or cholinergic systems) all of which have been associated with various behavioral changes as well as theories of memory update and learning (Doya, 2002). These effects are directly dependant on the neurons activity level dynamics and largely ignored in current deep learning systems although they might play a key role for learning in conjunction with environmental reward signals. (Nakahara und Hikosaka, 2012)

Finally, a third scale of change in neural states represent quantities $\tilde{\theta}_t$ that change more slowly; for example long-term potentiation (LTP) and depression (LTD) in synaptic connections as a function of the discussed neurons activity and chemical messengers, or the making and breaking of synaptic contacts that change on a neurodevelopmental timescale changing connectivity \tilde{c}_t . It is expected that the distribution of such quantities also follows some form of autoregressive generative process at a slower timescale than $\tilde{\mathbf{x}}_t$. Let k a strictly positive natural number, we have the following factorization:

$$q_{\tilde{\mathbf{x}}_{<kt}}(\tilde{\theta}_{kt}, \tilde{c}_{kt}) = q(\tilde{\theta}_0, \tilde{c}_0 | \tilde{\mathbf{x}}_0) \prod_{k=t}^{Kt} \int_{l=kt}^{(k+1)t} q(\tilde{\theta}_{kt} | \tilde{\theta}_{k(t-1)}, \tilde{c}_{k(t-1)}, \tilde{\mathbf{x}}_l) dt \quad (4.4)$$

While one can argue that LTP and LTD mechanisms identify most with gradient descent as they directly and durably tune the selectivity of neu-

rons with respect to their afferent inputs, one important pending question is how error information are propagated in the brain for tuning this selectivity. In this regard, backpropagation of error in artificial network is highly implausible as it posits a set of weights that is symmetric to feedforward propagation for propagating errors across layers of neurons. Other theories have been advanced, such as predictive coding (Rao und Ballard, 1999) which offers a falsifiable system linking neural activity to synaptic updates locally and in a Bayesian way which accommodate well to the free-energy principle (Friston, 2005).

Finally, let us note that evolution pressure through darwinian selection will likely influence all these adaptive systems jointly and constitute itself a higher level of learning favoring neural systems more able to maintain lower energetic states in specific ecosystems.

4.1.2 Inference and control

One essential aspect of the free energy principle is based on the Bayesian idea of the brain as an “inference engine”. In this sense, the free-energy principle unifies under one single physical perspective the notion of system control and Bayesian inference.

First, it is likely that the natural hierarchy of nested adaptation systems that discussed above is no accident but follows directly a hierarchical and compositional organization of the generative variable governing the world arrangement (Lake u. a., 2017). Indeed, the complexity and variety the variety of conditions in which agents experience reality is likely to drive agents towards adopting structured learning strategies in order to makes sense of the world. One illustration is hierarchical Bayesian models, which have been used for a long time to explain the learning-to-learn abilities of humans in various areas of cognition, from word learning to intuitive physics (Tenenbaum u. a., 2011) and have shown encouraging zero-shot compositional abilities in programmatic bayesian learning (Lake u. a., 2015). This observation calls for a research focus on discovering how to articulate learning in a representation-agnostic manner (Gershman und Niv, 2010; Wang u. a., 2021).

However, computing a probability distribution over an entire space of programs is usually intractable, and often even finding a single high-probability program poses an intractable search problem or require highly simplifying hypothesis and heuristics. A promising solution is to auto-

matically discover such self-arranging organizations through optimization. It has been observed that natural agents can learn their own learning bias when exposed to a serie of tasks (Baxter, 1998). More recently, meta-learning the weights of a artificial recurrent neural network, resulted in the spontaneous emergence of activation dynamics effectively tracking the sufficient statistics of the task at hand and performing decision making in a way close to Bayes optimal estimation (Ortega u. a., 2019).

In the next section, we comparatively review the existing work specifically combining artificial neural networks (ANNs) and meta-learning, before exploring how the principle of free-energy could drive research at the intersection of these fields.

4.2 Current approaches of meta-learning

If we accept the principle of free-energy minimization, it seems clear that learning-to-learn should actually be a core objective of machine learning, not only because discovering automatically new forms of learning might be the fastest path towards the emergence of broadly intelligent neural systems, but because this notion of self-update might be the very principle from which neural functions have unfolded in nature, from inference, memorization and learning. However, the general problem in equation (4.1) is intractable from a machine learning perspective because we generally don't have access to the generative model $p(\tilde{\mathbf{o}}_t, \tilde{\mathbf{z}}_t)$ of the world in which the agent evolve, nor do we know the specific form that the neural model $q_{\tilde{\mathbf{s}}_t}(\tilde{\mathbf{z}}_t)$ should take.

On the other hand, the current approach for building neural networks consists in a rudimentary bi-level optimization system: Heuristic choices guide a community wide trial-and-error process of model hypothesis that are further tuned with stochastic gradient descent with respect to a pre-defined energy function, leaving modestly interpretable models with restricted generality (Lake und Baroni, 2018; Cobbe u. a., 2020). A potential solution to the previously discussed challenges currently faced by current artificial neural networks is to *meta-learn* computational mechanisms able to rapidly capture task structures and automatically operate complex feedback control. In this view, artificial neural meta-learning constitutes a promising direction to build rapidly adaptive systems that could bridge current limitations of ANNs (sample complexity, robustness, generalization...) (Clune, 2019) and identify key neuroscience mechanisms that endow humans with their versatile learning abilities (Botvinick u. a.,

2019). Several works have explored different implementations of this idea that we broadly taxonomize below:

- A direct line of work consists in applying automatic strategies to design components of the modeling and optimization arsenal of deep learning to accelerate and improve learning performance of the target model over heuristic settings. Here, meta-learning refers to a principled exploitation of meta-knowledge extracted from previous experience in order to model and train neural networks more efficiently. While this research reach outside of deep learning, and converge in principle with automated machine learning (AutoML) (Guyon u. a., 2015), it has found interesting development for training neural networks beyond model selection from automatic hyperparameter tuning (Houthoof t u. a., 2018; Xu u. a., 2018; Gupta u. a., 2018) to neural architecture searches (Liu u. a., 2018, 2019).
- Other explicit optimization-based techniques have tried to cast fast adaptation at the episodic level as an explicit optimization procedure either by treating the optimizer as a black-box system (Santoro u. a., 2016a; Andrychowicz u. a., 2016; Ravi und Larochelle, 2017) or by learning a weight parametrization such that one or a few gradient steps are sufficient to adapt the input/output mapping to a specific task (Finn u. a., 2017a). This latter approach have been shown to approximate any learning algorithm in the limit of large models (Finn und Levine, 2017) and to be theoretically equivalent to maximum a posteriori estimation in bayesian inference (Grant u. a., 2018).
- Metric-based approaches have casted the problem of meta-learning as that of learning feature extractors whose output space that can be reused for new downstream tasks. Conceptually simple, this idea consist in learning a kernel function allowing to build a notion of similarity between new data points. This line of research has yield notably efficient models of few-shot classification (Vinyals u. a., 2016; Snell u. a., 2017b; Hariharan und Girshick, 2017) but suffer from lack of applicability and generality as data and tasks become more distant from the meta-training set.
- Orthogonally, work leveraging the Turing-completeness of recurrent neural networks has shown that such networks can be trained to store past information in their persistent activity state to inform current decision, in such a way that the network implements a form of reinforcement learning over each episode (Hochreiter u. a., 2001; Duan u. a., 2016; Wang u. a., 2018b). It is believed that

vanilla recurrent networks alone are not sufficient to meta-learn the efficient forms of episodic control found in biological agents (Lengyel und Dayan, 2008; Botvinick u. a., 2019). Hence additional work has tried to enhance the system with a better episodic memory model (Santoro u. a., 2016b; Pritzel u. a., 2017a; Ritter u. a., 2018b).

- Finally, a more recent line of research have exhibited impressive meta-learning abilities through direct neural conditioning. This form of learning consist essentially in marrying a transductive approach with neural netowrks by conditioning inputs neurons with exemplar data points. Various instances of such approaches are modern Hopfield networks (Ramsauer u. a., 2020) or neural conditional processes (Garnelo u. a., 2018). Recently, the key-query-value attentional mechanism of the Transformer (Vaswani u. a., 2017) applied over sets of object representations have been shown to exhibit emerging *in-context* learning abilities. This has been particularly identified in language modeling (Laskin u. a.) but also generalized to other arbitrary function classes (Garg u. a., 2022) or in meta-reinforcement learning (Mishra u. a., 2018; Laskin u. a.).

While these techniques have demonstrated that adaptive properties can emerge from such problem formulation in well-delimited contexts, they are far from the versatility of natural neural agents when faced to new contexts and tasks. In the next section, we introduce some potential work directions that will be further explored in the next chapters of this thesis.

4.3 Potential directions and challenges

In this section, we set forth several proposition to endow artificial neural networks with computational principles suceptible to drives them towards good and updactable representation of the world.

4.3.1 Explorable forms of neural control

Controllable dynamics

As stated previously, the main form of feedback control for building neural networks consists consist in gradient-based optimization of the parametric connections between neurons. These parameters are adjusted during a predefined curriculum with a fixed computational budget, yielding static solution with restricted generality (Lake und Baroni, 2018; Cobbe

u. a., 2020) and arguably preventing the emergence of online adaptivity. However, more online dynamics such as those performed by hidden state activations in recurrent neural networks can support adaptation as shown in (Duan u. a., 2016; Wang u. a., 2018b). As stated in section 4.3.1, there could exist also complementary dynamics to neurons activity that might potentially support adaptation and learning in artificial neural networks. For instance, the strenght of synaptic connections $\tilde{\theta}$ controlling gain of information transfer between neurons constitute a promising set of variable able to rapidly transform representations encoded by neurons activations that we explore in chapter 5. (Munkhdalai und Yu, 2017; Ha u. a., 2017; Miconi u. a., 2018b,a; Schlag und Schmidhuber, 2018a; Schlag u. a., 2020). While such models can be more memory intensive to train due to the need to track such dynamics for training, it might be possible to circumvent this challenge with new optimization tools (see chapter 2 and the adjoint method). Additionally, the dynamic formation of mesoscopic arrangement of neural states such as neural rhythms and synchrony (Deco u. a., 2011) can also be investigated as a possible to serve an intermediary purpose for neural computation and learning. Finally, an other possibility is to simulate the evolution of the network graph structure itself through creation and destruction of new connections, a process called neuroevolution. This also echoes theories of functional maintenance through structural "scaffolding" of neural circuitry (Park und Reuter-Lorenz, 2009). Modeling neuroevolution is however at odd with the differentiability requirement of gradient control. Hence, optimizing such dynamics require alternative methods such as evolutionary algorithm or reformulation as symbolic regression problem (Gong u. a., 2019). As current tensor programming tools and high-performance computing infrastructures become more flexible and offer the possibility to run larger simulations, the possibility to test such wider classes of algorithmic principles is opening up (Stanley u. a., 2019).

Hierarchy and time-scales

Developing co-active forms of control requires to carefully mind the interrelation between their dynamics and how they reciprocally influence each other. One reading template that was already described for biological neurons in section is temporality (Botvinick u. a., 2019): Building models that can detect and adapt at multiple time scale can be a key feature for the emergence of compact and factorized representations of the generative variables describing the environment. This idea has already found a theoretical development in unsupervised learning through slow feature analysis (Wiskott und Sejnowski, 2002). At another abstract level, one

way of managing the difficulty inherent to the conceptualization of such hierarchical systems is to break it down into simpler problems through recursive thinking as in dynamic programming (Bellman, 1966). Another interesting concept is the notion of self-referential dependence of the adaptation rule to the neural states themselves which might actually be beneficial as it allows for plasticity of the control rule itself (Schmidhuber, 2005; Irie u. a., 2022b).

The role of memory

Specifically describing how neural networks can compress, store and manipulate past information is also a critical aspect for optimal inference and planning in complex environments. Again, the different forms of memory abilities found in natural intelligence (working, episodic or long-term) can be interpreted as a by-product of the energetic surprise minimization objective, but the question of their computational implementation remains largely open. On the other hand, meta-reinforcement learning has been shown to benefit from explicit modeling of information storage and retrieval in recurrent neural networks (Pritzel u. a., 2017b; Ritter u. a., 2018a; Lin u. a., 2018). Additionally, the success of Transformers (Vaswani u. a., 2017) in offline reinforcement learning (Chen u. a., 2021) or natural language generation (Brown u. a., 2020b), whose memory system consist in an explicit and external storage of past experience in the forms of a sequence of states passed as input to the neural model, has determined the emergence of an intermediary useful adaptation process called *in-context learning*. The poor scalability of this memory system, quadratic in the length of sequences, suggest however that current artificial neural memory systems are lacking satisfactory compression and representation mechanisms for efficiently perform lifelong learning.

Operator learning

A recent line of work have shown that the universal approximation abilities of neural networks extend to higher-order functions (Neural Operators) (Chen und Chen, 1995; Lu u. a., 2019; Li u. a., 2020a) which calls for a deeper exploration of neural control with neural networks themselves. The universal approximation capabilities of neural network could in fact allow to discover learning rules in the form of higher-level neural parametrization. This idea has already found development from the perspective of weight compression (Ha u. a., 2017), or equivariance in vision with dynamic convolution filters (Jaderberg u. a., 2015; Jia u. a., 2016). Though not explicitly formulated as neural operator control, these approaches effectively implement a form of modulation of neural network

activity as a function of input data and activation state, resulting in more compact and expressive models. These findings point towards the possibility of building neural networks with automatically searchable adaptive mechanisms that determines the spontaneous emergence of learning principles from underspecified formulation. For instance, in the next chapters, we will show that systems endowed with trainable neural feedback loops can automatically operate closed-loop control to achieve higher-level goals or to guide a dynamical system to specified optimal states.

Approximating the free-energy objective

As noted in the previous chapter, training neural networks can be seen as finding and solving objective functions that implicitly approximate some aspects of the general free-energy minimization problems. However, the search for objective functions that are able to force high representation capacity in a more ecologically plausible way while guaranteeing modularity and transferability of this acquired cognitive property to new contexts, is still lacking a principled analysis. One direction of research is to continue reverse-engineering the computational principle that characterise biological neurons by optimizing artificial models to display specific behaviors with particular objectives. For instance, theories that unify neuronal and synaptic dynamics under a single minimization objective such as in predictive coding (Huang und Rao, 2011) can be tested at large scale. While at the moment deep learning objectives are more oriented toward goal completion rather than skill acquisition, the recent interpretation of supervised discriminative neural networks as an implicit energy model (Grathwohl u. a., 2020) or the success of self-supervised objective functions for crafting highly transferable artificial neural feature extractors (Misra und Maaten, 2020) suggest that vanilla supervised learning objectives can be efficiently regularised to learn better representations.

Curriculum learning

Finally, another dimension of research for neural meta-learning is curriculum learning. Designing tasks and sequence of tasks that indirectly favor artificial neural networks solution that with spontaneously adaptive behavior is a difficult problem, often led by intuition or computation constraints in deep learning. However, varying adaptation tasks and minding the order in which they are presented to learning neural networks, is likely to accelerate the emergence of well-behaved solutions as observed in the case of supervised learning (Soviany u. a., 2022). In this view, automatic generation of curriculum through co-evolution is also likely to find original solutions for curriculum learning (Miconi, 2021; Pontes-

[Filho u. a., 2022](#)). Going further, scaling curriculum to larger pool of tasks can determine the emergence of reusable skills by constraining the meta-training solution space ([Team u. a., 2021](#); [Reed u. a., 2022](#)). Such approaches will require a systematic and compositional formulation of designing new tasks and contexts that only programmatic solutions will be to offer ([Clune, 2020](#)).

4.4 Transition

Throughout the next chapters, we will explore ideas originating from these propositions in different context ranging from supervised classification to reinforcement learning or regression of physical systems. The positive results obtained throughout over a variety of tested computational principles and applications are clear markers of the validity and the potential of meta-learning for more general and efficient learning in artificial neural networks.

Chapter 5

Self-modifying networks

“To improve is to change; to be perfect is to change often.”

Winston Churchill

Contents

5.1	Introduction	112
5.2	Meta-Reinforcement Learning	113
5.3	Synaptic plasticity	116
5.4	Reinforcement with meta-plasticity	117
5.5	Experiments	119
5.6	Optimization	128
5.7	Experimental details	131
5.8	Discussion	135

5.1 Introduction

Deep Reinforcement Learning has demonstrated the potential of neural networks tuned with gradient descent for solving complex tasks in well-delimited environments. However, these neural systems are slow learners producing specialized agents with no mechanism to continue learning beyond their training curriculum. On the contrary, biological synaptic plasticity is persistent and manifold, and has been hypothesized to play a key role in executive functions such as working memory and cognitive flexibility, potentially supporting more efficient and generic learning abilities. Inspired by this, we explore in this chapter, an original class of

neural networks with dynamic weights, able to continually perform self-reflexive modification as a function of their current synaptic state and action-reward feedback though tuned plasticity principles parameterized at the synapse level. The resulting model, MetODS (for Meta-Optimized Dynamical Synapses) is a broadly applicable meta-reinforcement learning system able to learn efficient and powerful control rules in the agent policy space. A single layer with dynamic synapses can perform one-shot learning, generalize navigation principles to unseen environments and demonstrate a strong ability to learn adaptive motor policies, comparing favorably with previous meta-reinforcement learning approaches.

5.2 Meta-Reinforcement Learning

5.2.1 Background

In this chapter, we refer to “tasks” as Markov decision processes (MDP) defined by the following tuple $\tau = (\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \rho_0)$, where \mathcal{S} and \mathcal{A} are respectively the state and action sets, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto [0, 1]$ refers to the state transition distribution measure associating a probability to each tuple (state, action, new state), $r : \mathcal{A} \times \mathcal{S} \mapsto \mathbb{R}$ is a bounded reward function and ρ_0 is the initial state distribution. For simplicity, we consider finite-horizon MDP with T time-steps although our discussion can be extended to the infinite horizon case as well as partially observed MDP. We further specify notation when needed by subscripting symbols with the corresponding task τ or time-step t .

5.2.2 Connection to optimal transport

Meta-Reinforcement learning considers the problem of generating policies π in a policy space Π that perform well on a set \mathbb{T} of tasks with distribution $\mu_{\mathbb{T}}$, using a reinforcement signal r coming from a sequence of interactions with the task environments. Provided the existence of an optimal policy π^* for any task $\tau \in \mathbb{T}$, we can define the distribution measure μ_{π^*} of these policies over Π . Arguably, an ideal system aims at associating to any task τ its optimal policy π^* , i.e, finding the transport plan γ in the space $\Gamma(\mu_{\mathbb{T}}, \mu_{\pi^*})$ of couplings with marginals $\mu_{\mathbb{T}}$ and μ_{π^*} that maximizes the expected cumulative reward \mathcal{R} :

$$\max_{\gamma \in \Gamma(\mu_{\mathbb{T}}, \mu_{\pi^*})} \mathbb{E}_{(\tau, \pi) \sim \gamma} \left[\mathcal{R}(\tau, \pi) \right] \quad \text{where} \quad \mathcal{R}(\tau, \pi) = \mathbb{E}_{\pi, \mathcal{P}_{\tau}} \left[\sum_{t=0}^T r_{\tau}(\mathbf{a}_t, \mathbf{s}_t) \right] \quad (5.1)$$

Here \mathcal{R} corresponds to the expected cumulative reward of action-state trajectories $(\mathbf{s}_0, \mathbf{a}_0, \dots, \mathbf{s}_T, \mathbf{a}_T)$ for task τ and policy π , i.e such that state transitions are governed by $\mathbf{s}_{t+1} \sim \mathcal{P}_\tau(\cdot | \mathbf{s}_t, \mathbf{a}_t)$, actions are sampled according to the policy π : $\mathbf{a}_t \sim \pi$ and initial state \mathbf{s}_0 follows the distribution $\rho_{0,\tau}$. Most generally, problem (5.1) is intractable, since μ_{π^*} is unknown or has no explicit form. Instead, previous approaches aim to optimize a surrogate problem, by defining an iterative ‘‘specialization’’ procedure which builds for any task τ , a sequence (π_t) of improving policies (see Fig. 5.1).

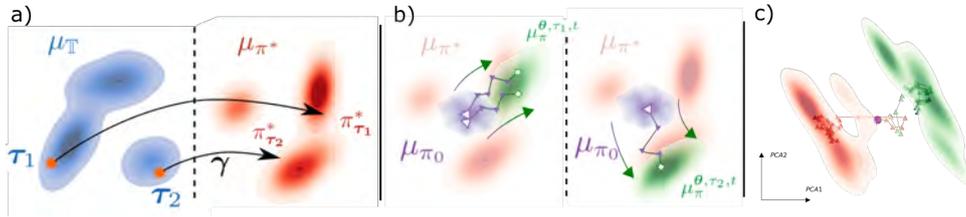


Figure 5.1 – **Meta-Reinforcement Learning as a transport problem and MetODS synaptic adaptation:** **a)** Associating any task τ in \mathbb{T} to its optimal policy π_τ^* in Π can be regarded as finding an optimal transport plan from $\mu_{\mathbb{T}}$ to μ_{π^*} with respect to the cost $-\mathcal{R}$. Finding this transport plan is generally an intractable problem. **b)** Meta-RL approximates a solution by defining a stochastic flow in the policy space Π conditioned by the current task τ and driving a prior distribution μ_{π_0} of policies π_0 towards a distribution $\mu_{\pi}^{\theta, \tau, t}$ of policies with high score \mathcal{R} . **c)** Density and mean trajectories of our model dynamic weights principal components over several episodes of the Harlow task (see section 5.5.1) reveal this policy specialization. Two modes colored with respect to whether the agent initial guess was good or **bad** emerge, corresponding to two different policies to solve the task.

Defining θ the meta-parameters governing the evolution of the sequences (π_t) and $\mu_{\pi}^{\theta, \tau, t}$ the distribution measure of the policy π_t after learning task τ during some period t , the optimization problem amounts to finding the meta-parameters θ that best adapt $\pi_t \sim \mu_{\pi}^{\theta, \tau, t}$ over the task distribution.

$$\max_{\theta} \mathbb{E}_{\tau \sim \mu_{\mathbb{T}}} \left[\mathbb{E}_{\pi \sim \mu_{\pi}^{\theta, \tau, t}} [\mathcal{R}(\tau, \pi)] \right] \quad (5.2)$$

Formulation (7.4) allows to appreciate the different meta-RL approaches

previously proposed under the unifying paradigm of flows in the policy space Π . How such flows are constructed to "guide" policies towards high-reward regions of the policy space determines the quality of a meta-learning system. For instance, imitation learning leverages an oracle teacher that represents a good approximation of the target distribution μ_{π^*} of optimal policies that inform the direction of the policy update (Finn u. a., 2017b). When no information or demonstration is provided, τ can only be known by sampling state-action trajectories $(\mathbf{s}_t, \mathbf{a}_t)_{t \leq T}$ of τ . Model-based meta-RL approaches with posterior approximation of the task such as PEARL (Rakelly u. a., 2019) and VariBAD (Zintgraf u. a., 2019) accelerate policy adaptation by learning efficient task identification, but might fail when tasks are too different from the training task distribution. On the other hand, MAML (Finn u. a., 2017a) is more generic as it corresponds to an explicit gradient flow with respect to \mathcal{R} and benefits theoretically from convergence properties towards a distribution over fixed points in Π even for unseen tasks. However, gradient-based update might be inefficient as it requires a lot of interaction with the environment to yield an accurate estimate of the update direction, and it is unclear at which frequency or which event should trigger an update of the synaptic configuration of the network. Altogether, problem (7.4) emphasizes three desirable properties of a reinforcement learning program that we further discuss below and for which we show that meta-learned synaptic updates have a strong potential.

Efficiency: How "fast" the distribution $\mu_{\pi}^{\theta, \tau, t}$ is transported towards a distribution of high-performing policies when accumulating experience on a task τ . Arguably, an efficient learning mechanism should require few interaction steps t with the environment to identify the task rule and adapt its policy accordingly. We show that learnt synaptic update rules are able to change the agent transfer function drastically in a few updates, thus supporting one-shot learning of a task-contingent association rule in the Harlow task, adapting a motor policy in a few steps or exploring original environments quickly in the Maze experiment.

Capacity: That property defines how sensitive the learner is to specific task features and determines its achievable level of performance for a distribution of tasks μ_{τ} . It is linked to the sensitivity of learning system shaping $\mu_{\pi}^{\theta, \tau, t}$ to task particularities, i.e how the agent captures and retains task sufficient statistics, as well as their conversion into a precise state in the policy space. Because our mechanism is continual, it allows for constant tracking of the environment information and policy update. We test this property in the maze experiment in Section 5.5, showing that tuned online synaptic updates obtain the best capacity under sys-

tematic variation of the environment.

Generality: We refer here to the overall ability of the meta-learnt policy flow to drive $\mu_{\pi}^{\theta, \tau, t}$ towards high-performing policy regions for a diverse set of tasks (*generic trainability*) but also to how general is the resulting reinforcement learning program and how well it transfers to tasks unseen during training (*transferability*). In the former case, since the proposed synaptic mechanism is model-free, it allows for tackling diverse types of policy learning, from navigation to motor control. Arguably, to build reinforcing agents that learn in open situations, we should strive for generic and efficient computational mechanisms rather than learnt heuristics. Transferability corresponds to the ability of the meta-learned policy flow to generally yield improving updates even in unseen policy regions of the space $\mathbf{\Pi}$ or conditioned by unseen task properties: new states, actions and transitions, new reward profile. etc. We show in a motor-control experiment using the Meta-World benchmark that meta-tuned synaptic updates are a potential candidate to produce a more systematic learner agnostic to environment setting and reward profile. The generality property remains the hardest for current meta-RL approaches, demonstrating the importance of building more stable and invariant control learning rules.

5.3 Synaptic plasticity

5.3.1 Artificial fast plasticity:

Networks with dynamic weights that can adapt as a function of neural activation have shown promising results over regular recurrent neural networks to handle sequential data (Munkhdalai und Yu, 2017; Ha u. a., 2017; Miconi u. a., 2018b,a; Schlag und Schmidhuber, 2018a; Schlag u. a., 2020). However, contrary to our work, these models postulate a persistent neural activity orchestrating weights evolution. On the contrary, we show that synaptic states are the sole persistent components needed to perform fast adaptation. Additionally, the possibility of optimizing synaptic dynamics with evolutionary strategies in randomly initialized networks (Najarro und Risi, 2020) or through gradient descent (Miconi, 2016) has been demonstrated, as well as in a time-continuous setting (Choromanski u. a., 2020a). Recent results have shown that plasticity rules differentially tuned at the synapse level allow to dynamically edit and query networks memory (Miconi u. a., 2018b; Chalvidal u. a., 2021). However another specificity of this work is that our model synaptic rule is a function of reward and synaptic state, allowing to drive weight dy-

namics conditionally on both an external feedback signal and the current model belief.

5.3.2 Potential mechanism for working memory

As discussed above, efficient memory storage and manipulation is a crucial feature for building rapidly learning agents. To improve over vanilla recurrent neural network policies (Duan u. a., 2016), some models have augmented recurrent agents with content-addressable dictionaries able to reinstate previously encoded patterns given the current state (Weston u. a., 2014; Zaremba und Sutskever, 2016; Pritzel u. a., 2017c; Botvinick u. a., 2019). However these slot-based memory systems are subject to interference with incoming inputs and their memory cost grows linearly with experience. Contrastingly, attractor networks can be learnt to produce fast compression of sensory information into a fixed size tensorial representation (Bartunov u. a., 2019; Zhang und Zhou, 2017). One class of such network are Hopfield networks (Hopfield, 1982a; Koiran, 1994; Demircigil u. a., 2017; Krotov und Hopfield, 2016) which benefit from a large storage capacity (Krotov und Hopfield, 2016), can possibly perform hetero-associative concept binding (Schlag und Schmidhuber, 2018b; Schlag u. a., 2020) and produce fast and flexible information retrieval (Ramsauer u. a., 2020).

5.4 Reinforcement with meta-plasticity

5.4.1 Learning weights update rules

Our model learns to train itself by updating its weights through interaction with the environment and its own current weight state. This mechanism enables MetODS to rapidly compress experience of a task τ into a particular synaptic configuration, building the following policy sequence:

$$\forall t \leq T, \quad \pi(\mathbf{a}|\mathbf{s}, \mathbf{W}_t) \sim \mu_{\pi}^{\theta, \tau, t} \quad (5.3)$$

where \mathbf{W}_t are instance-particular dynamic weights governed by locally parameterized update rules driving their evolution over time and with respect to the state trajectory $(\mathbf{s}_{i \leq t}, \mathbf{a}_{i < t}, \mathbf{r}_{i < t})$. Specifically, at every time-step t , network computation and learning rules consists in recursive application of read-write operations such that the model learns to update its weights given both external stimuli \mathbf{v}_t and relevant information stored in the network synaptic configuration \mathbf{W}_t .

Update operations: The core mechanism consists in two simple operations that respectively linearly project neurons activation \mathbf{v} through the dynamic weights \mathbf{W} followed by an element-wise non-linearity, and build an update through a hebbian update rule with element-wise weighting α :

$$\begin{cases} \phi(\mathbf{W}, \mathbf{v}) = \sigma(\mathbf{W} \cdot \mathbf{v}) & \text{read} \\ \psi(\mathbf{v}) = \alpha \odot \mathbf{v} \otimes \mathbf{v} & \text{write} \end{cases} \quad (5.4)$$

where, α is a matrix of $\mathbb{R}^{N \times N}$, \otimes denotes the outer-product, \odot is the element-wise multiplication, σ is a non-linear activation function. The element-wise weighting α allows for different plasticity amplitudes at every connection consistent with biology and locally tuned synaptic plasticity (Abbott und Nelson, 2000; LEE201431, 2014), and generating a matrix update with potentially more than rank one as in the classic hebbian rule.

Multi-step scheme We further introduce a multi-step scheme that recursively applies the previous rules S times. This scheme allows to learn relations between stored patterns and incoming information by mixing information between current neural activation and previous iterates. Starting from an initial activation pattern $\mathbf{v}^{(0)}$ and previous weight state $\mathbf{W}^{(0)} = \mathbf{W}_{t-1}$, the model recursively applies equations in (5.4) on $\mathbf{v}^{(s)}$ and $\mathbf{W}^{(s)}$ such that:

$$\text{for } s \in [1, S] \quad : \quad \begin{cases} \mathbf{v}^{(s)} = \sum_{l=0}^{s-1} \kappa_s^{(l)} \mathbf{v}^{(l)} + \kappa_s^{(s)} \phi(\mathbf{W}^{(s-1)}, \mathbf{v}^{(s-1)}) \\ \mathbf{W}^{(s)} = \sum_{l=0}^s \beta_s^{(l)} \mathbf{W}^{(l)} + \beta_s^{(s)} \psi(\mathbf{v}^{(s-1)}) \end{cases} \quad (5.5)$$

Parameters $\kappa_s^{(l)}$ and $\beta_s^{(l)}$ are learnt along with plasticity parameters α , and correspond to delayed contributions of previous patterns and synaptic states to the current operation. This is motivated by biological evidence of different time-scales in synaptic neuromodulators concentration change and of their mutual retroactive influence over synaptic efficacy. Finally, $(\mathbf{v}^{(S)}, \mathbf{W}^{(S)})$ are respectively used as activation for the next layer, and as the new synaptic state \mathbf{W}_t . In this work, we test a single dynamic layer and leave the extension of the synaptic plasticity to the full network for future work. In order for the model to learn a credit assignment strategy, state transition and previous reward information $[\mathbf{s}_t, \mathbf{a}_{t-1}, \mathbf{r}_{t-1}]$ are embedded into a vector \mathbf{v}_t by a feedforward map \mathbf{f} as in previous meta-RL approaches (Duan u. a., 2016; Mishra u. a., 2018). Action and

advantage estimate are read-out by a feedforward policy map \mathbf{g} . Altogether, the synaptic control as well as the state-value and action policy estimate of MetODS consists in the update presented in Algorithm 2. We meta-learn the plasticity and update coefficients, as well as the embedding and read-out function altogether, hence the meta-parameters are $\boldsymbol{\theta} = [\mathbf{f}, \mathbf{g}, \boldsymbol{\alpha}, \boldsymbol{\kappa}, \boldsymbol{\beta}]$. Additionally, the initial synaptic configuration \mathbf{W}_0 can be learnt, fixed a priori or sampled from an specified distribution.

Computational interpretation: We note that if $S=1$ in equation (5.5), the operation boils down to a simple hebbian update with a synapse-specific weighting $\boldsymbol{\alpha}^{i,j}$. This perspective makes MetODS an original form of modern Hopfield network (Ramsauer u. a., 2020) with hetero-associative memory that can dynamically access and edit stored representations driven by observations, rewards and actions. While pattern retrieval from Hopfield networks has a dense litterature, our model recursive scheme is an original proposal to learn automatic synaptic updates able to bind representations across timesteps. The promising results shown in our experimental section suggest that such learnt updates can generate useful self-modifications to sequentially adapt to incoming information at runtime.

Algorithm 1 MetODS algorithm

- 1: **Require:** $\boldsymbol{\theta} = [\mathbf{f}, \mathbf{g}, \boldsymbol{\alpha}, \boldsymbol{\kappa}, \boldsymbol{\beta}]$ and \mathbf{W}_0
 - 2: **for** $1 \leq t \leq T$ **do**
 - 3: $\mathbf{v}^{(0)} \leftarrow f(\mathbf{s}_t, \mathbf{a}_{t-1}, \mathbf{r}_{t-1})$
 - 4: $\mathbf{W}^{(0)} \leftarrow \mathbf{W}_{t-1}$
 - 5: **for** $1 \leq s \leq S$ **do**
 - 6: $\mathbf{v}^{(s)} \leftarrow \sum_{l=0}^{s-1} \boldsymbol{\kappa}_s^{(l)} \mathbf{v}^{(l)} + \boldsymbol{\kappa}_s^{(s)} \sigma(\mathbf{W}^{(s-1)} \cdot \mathbf{v}^{(s-1)})$
 - 7: $\mathbf{W}^{(s)} \leftarrow \sum_{l=0}^{s-1} \boldsymbol{\beta}_s^{(l)} \mathbf{W}^{(l)} + \boldsymbol{\beta}_s^{(s)} (\boldsymbol{\alpha} \odot \mathbf{v}^{(s-1)} \otimes \mathbf{v}^{(s-1)})$
 - 8: **end for**
 - 9: $\mathbf{a}_t, \mathbf{v}_t \leftarrow \mathbf{g}(\mathbf{v}^{(s)})$
 - 10: $\mathbf{W}_t \leftarrow \mathbf{W}^{(s)}$
 - 11: **end for**
-

5.5 Experiments

In this section, we explore the potential of our meta-learnt synaptic update rule with respect to the three properties of the meta-RL problem exposed in section ???. Namely, 1) *efficiency*, 2) *capacity* and 3) *generality*

of the produced learning algorithm. We compare it with three state-of-the-art meta-RL models based on different adaptation mechanisms: RL² (Duan u. a., 2016) a memory-based algorithm based on training a GRU-cell to perform Reinforcement Learning within the hidden space of its recurrent unit, MAML (Finn u. a., 2017a) which performs online gradient descent on the weights of three-layer MLP and PEARL (Rakelly u. a., 2019), which performs probabilistic task inference for conditioning policy. Details for each experimental setting are further discussed in annex and the code can be found at <https://github.com/mathieuchal/methods2022>.

5.5.1 Efficiency: One-shot reinforcement learning

To first illustrate that learnt synaptic dynamics can support fast behavioral adaptation, we use a classic experiment from the neuroscience literature originally presented by Harlow (Harlow, 1949) and recently reintroduced in artificial meta-RL in (Wang u. a., 2018b) as well as a heavily-benchmarked MuJoCo directional locomotion task (see Fig. 5.2). To behave optimally in both settings, the agent must quickly identify the task rule and implement a relevant policy : The Harlow task consists of five sequential presentations of two random variables placed on a one-dimensional line with random permutation of their positions that an agent must select by reaching the corresponding position. One value is associated with a positive reward and the other with a negative reward. The five trials are presented in alternance with periods of fixation where the agent should return to a neutral position between items. In the MuJoCo robotic *Ant-dir* experiment, a 4-legged agent must produce a locomotive policy given a random rewarded direction.

5.5.2 Harlow task

This experiment consists in a 1-dimensional simplification of the task presented in (Wang u. a., 2018b). The action space is the discrete set $\{-1, 1\}$ moving the agent respectively to the left/right on a discretized line. The state space consist in 17 positions while the agent receptive field is eight dimensional. Values are placed at 3 positions from the fixation target position. The fixation position yields a reward of 0.2 while the values are drawn uniformly from $\llbracket 0, 10 \rrbracket$ and randomly associated with a reward of -1 and 1 at the beginning of each episode. The maximal duration for an episode is 250 steps, although the model solves the 5 trials in ~ 35 steps on average. Since values location are randomly permuted across

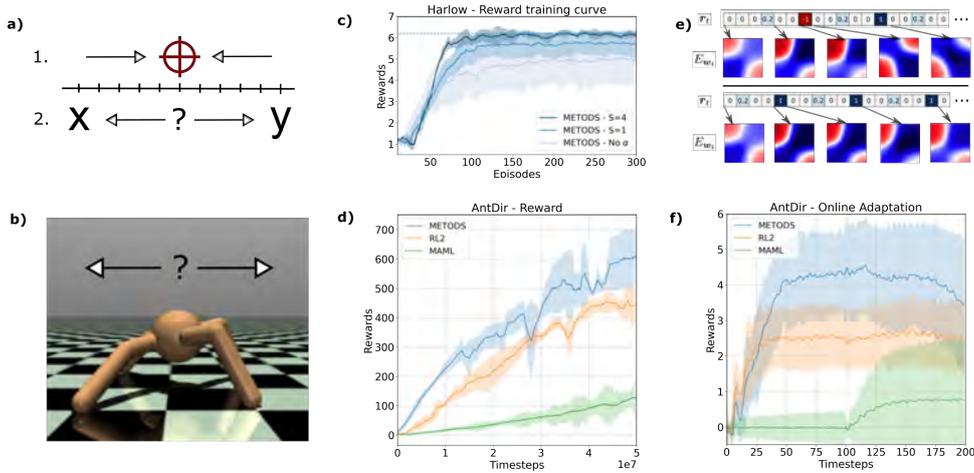


Figure 5.2 – **a-b)** Schemas of the Harlow and Mujoco *Ant-directional* locomotion task. An episode of the Harlow task consists of five sequential presentations of two random variables placed on a one-dimensional line with random permutation of their positions that an agent must select by reaching the corresponding position. One value is associated with a positive reward and the other with a negative reward. The five trials are presented in alternance with periods of fixation where the agent should return to a neutral position between items. In *Ant-dir*, the agent must learn to select a rewarded direction of locomotion over a single episode of 200 steps. **c-d)** Evolution of accumulated reward over training. In the Harlow task, we conduct an ablation study by either reducing the number of recursive iterations ($S=1$) or removing the trainable plasticity weights α resulting in sub-optimal policy. In *Ant-dir* we compare our agent training profile against MAML and RL². **e)** We can interpret the learned policy in terms of a Hopfield energy adapting with experience. We show horizontally two reward profiles of different episodes and the energy $E_{\mathbf{W}_t}(v_1, v_2) = -v_1^T \mathbf{W}_t v_2$ along two principal components of the vector trajectory \mathbf{v}_t . In the first episode, the error in the first presentation (red square) transforms the energy landscape which changes the agent policy, while on the other episode, the model belief does not change over time. Note the two modes for every energy map, which allows the model to handle the potential position permutation of the presented values. **f)** Average rewards per timestep during a single episode of the *Ant-dir* task.

presentations within one episode, the learner cannot develop a mechanistic strategy to reach high rewards based on initial position. Instead, to reach the maximal expected reward over the episode, the agent needs to perform one-shot learning of the task-contingent association rule during the first presentation. We found that even a very small network of

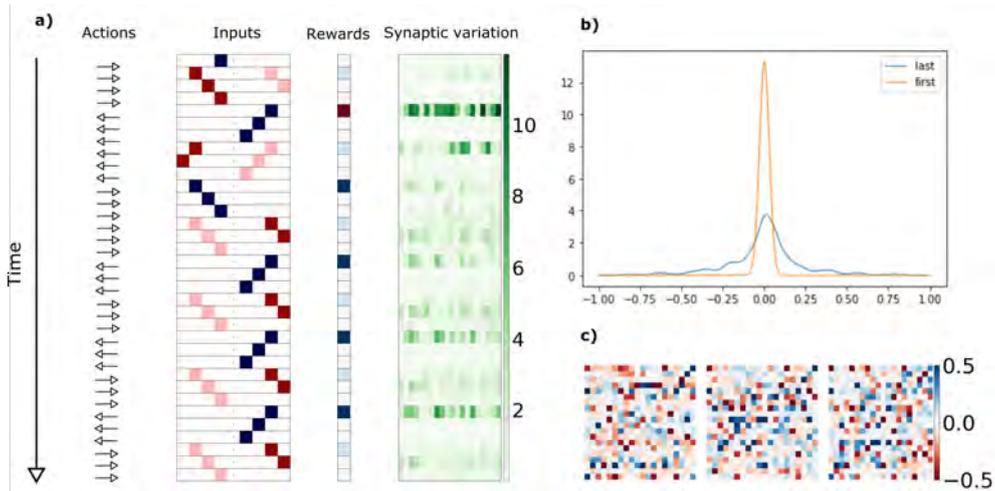


Figure 5.3 – **Harlow task:** **a)** An episode of the Harlow task. From left to right: Actions, visual field of the agent (blue squares: fixation target, red squares: values), Rewards (red; negative, light blue: fixation reward, dark blue: value reward) and sum of absolute synaptic variation $\|\sum \Delta \mathbf{W}\|$ per neuron. **b)** Distribution of weights before and after adaptation over 5 presentations. **c)** Three instances of the dynamic weights \mathbf{W}_t after adaptation that solved the Harlow task. Every synaptic configuration presents differences but perform optimally.

$N=20$ neurons proved to be sufficient to solve the task perfectly. We investigated the synaptic mechanism encoding the agent policy. A principal component analysis reveals a policy differentiation with respect to the initial value choice outcome supported by synaptic specialization in only a few time-steps (see Figure 5.1-c). We can further interpret this adaptation in terms of sharp modifications of the Hopfield energy of the dynamic weights (Figure 5.2-e). Finally, the largest synaptic variations measured by the sum of absolute synaptic variations occur for states that carry a non-null reward signal (see S.I). These results suggest that the recursive hebbian update combined with reward feedback is sufficient to support one-shot reinforcement learning of the task association rule.

5.5.3 Gym Mujoco directional robot control

◇ **Setting:** We consider the directional rewards task proposed in (Finn *u. a.*, 2017a) with the *Ant* and *Cheetah* robots, as a more complex test of rapid adaptation. We apply standard RL practice for continuous con-

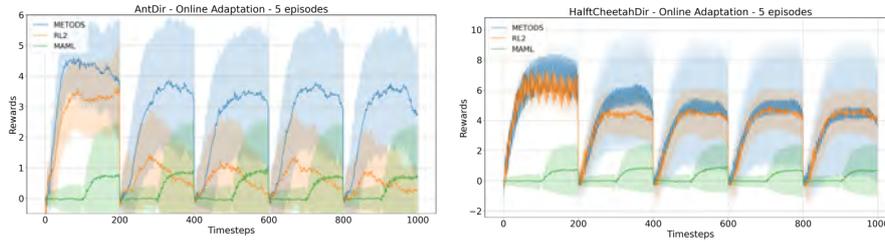


Figure 5.4 – Reward profiles of MAML, RL^2 and MetODS over 5 consecutive episodes with randomly varying rewarded direction. In *Cheetah*, MetODS slightly overperforms RL^2 and in *Ant*, MetODS is the only approach able to reach performance comparable to first episode in subsequent episodes.

trol by parameterizing the stochastic policy as a product of independent normal distributions with fixed standard deviation $\sigma = 0.1$ and mean inferred by the agent network. A training meta-episode consists in a single rollout of 200 steps with random sampling of the reward direction for each episode. We trained models for performing the task on an episode of 200 timesteps and found that MetODS can adapt in a few time-steps, similar to memory-based models such as RL^2 , (Figure 5.2-f) thanks to its continual adaptation mechanism. By design, MAML and PEARL do not present such a property, and they need multiple episodes before being able to perform adaptation correctly. We still report MAML performance after running its gradient adaptation at timestep $t=100$. We further note that our agent overall performance in a single episode is still superior to MAML performance reported in (Finn u. a., 2017a; Mishra u. a., 2018) when more episodes are accessible for training.

◇ **Robot impairment:** We partially impaired the agent motor capabilities by "freezing" one of the robot actuator. Namely we consistently passed a value of zero to the the *right_back_leg* in Ant (coordinate 8 in OpenAI Gym XML asset files) and *ffoot* in Cheetah (coordinate 6 in OpenAI Gym XML asset files).

◇ **Continual adaptation:** Finally, we further investigated whether the adaptation mechanism was also adjustable after performing initial adaptation by testing trained agents over 5 consecutive episodes with randomly changing rewarded direction while retaining weight state across episodes. We show in Fig 10 the average reward profile of the three meta-RL agents over 1000 test episodes, where MetODS adapt faster and better with respect to reward variations.

5.5.4 Capacity : Maze exploration task

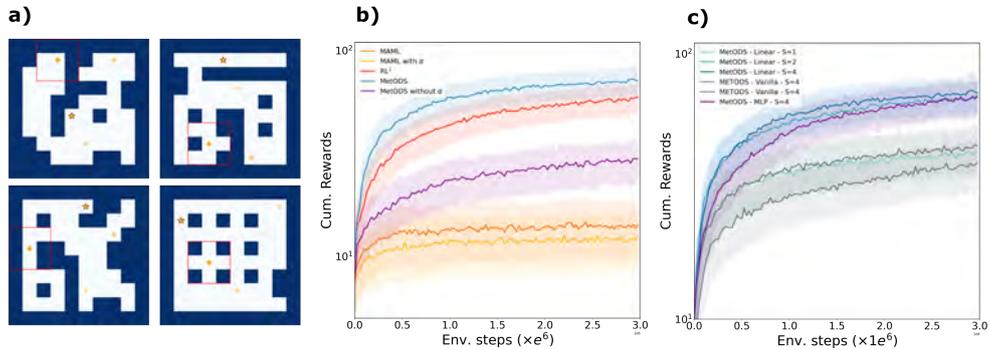


Figure 5.5 – **a)** Examples of generated maze configurations. The mazes consist in 8×8 pix. areas, with walls and obstacles randomly placed according to a variation of Prim’s algorithm (Prim, 1957) with target location randomly selected for the entire duration of the episode (star). The agent receptive field is highlighted in red. **b)** Comparisons with MAML and RL^2 and effect of element-wise plasticity α . **c)** Variations of MetODS writing mechanism as well as depth S of recursivity yield different performances (see. S.I for further details)

We further tested the systematicity of MetODS learnt reinforcement program on a partially observable Markov decision process (POMDP): An agent must locate a target in a randomly generated maze while starting from random locations and observing a small portion of its environment (depicted in Figure 5.5). While visual navigation has been previously explored in meta-RL (Duan u. a., 2016; Mishra u. a., 2018; Miconi u. a., 2018b), we here focus on the mnemonic component of navigation by complexifying the task in two ways, we reduce the agent’s visual field to a small size of 3×3 pix. and randomize the agent’s position after every reward encounter. The agent can take discrete actions in the set $\{\text{up,down,left,right}\}$ which moves it accordingly by one coordinate. The agent’s reward signal is solely received by hitting the target location, thus receiving a reward of 10. Each time the agent hits the target, its position is randomly reassigned on the map (orange) and the exploration resumes until 100 steps are accumulated during which the agent must collect as much reward as possible. Note that the reward is invisible to the agent, and thus the agent only knows it has hit the reward location because of the activation of the reward input. The reduced observability of the environment and the sparsity of the reward signal (most of the state transitions yield no reward) requires the agent to perform logical binding between distant temporal events to navigate the maze. Again, this setting rules out PEARL since its latent context encoding mecha-

nism erases crucial temporal dependencies between state transitions for efficient exploration. Despite having no particular inductive bias for efficient spatial exploration or path memorization, a strong policy emerges spontaneously from training.

Agent	1st rew.* (\downarrow)	Success (\uparrow)	Cum. Rew. (\uparrow)	Cum. Rew (L) (\uparrow)
Random	96.8 ± 0.5	5 %	3.8 ± 8.9	3.7 ± 6.4
MAML	64.3 ± 39.3	45.2%	14.95 ± 4.5	5.8 ± 10.3
RL ²	16.2 ± 1.1	96.2%	77.7 ± 46.5	28.1 ± 29.7
MetODS	14.7 ± 1.4	96.6%	86.5 ± 46.8	34.9 ± 34.9

Figure 5.6 – Performance of Meta-RL models tested at convergence ($1e^7$ env. steps). MetODS better explores the maze as measured by the average number of steps before 1st reward and the success rate in finding the reward at least once. It then better exploits the maze as per the accumulated reward. (* We assign 100 to episodes with no reward encounter.)

◇ **Models capacity:** In this experiment, since model memory capacity seems to directly impact final performance, we specifically control for the number of learnable parameters in order to compare models and fix a training budget of 10M env. steps. Since MAML does not have a continual adaptation mechanism, we perform gradient adaptation every 20 time-steps during an episode, in order to balance noisiness of the gradient and rapidity of exploration. We show in Table 5.1 the mean over 1000 episodes of the accumulated reward of agents trained on 8×8 mazes and tested on different maze sizes $N \in (4, 6, 8, 10, 12)$. Agents performances are highly varying within a size setting due to differences in maze configurations and across maze sizes due to increasing complexity. However, MetODS and RL² are able to generalize at least partially to these new settings (see table 5.1). MetODS outperforms RL² in every setting, and generalizes better, retaining an advantage of up to 25% in accumulated reward in the biggest maze.

Maze size	MAML	RL ²	MetODS	Rel imp. to RL ²
6	21.0 ± 18.4	149.3 ± 66.7	169.1 ± 66.1	13%
8	14.9 ± 4.5	72.1 ± 45.6	87.3 ± 48.3	20%
10	5.7 ± 7.9	28.1 ± 29.7	34.9 ± 34.9	21%
12	3.9 ± 6.9	11.1 ± 15.8	13.9 ± 19.8	25%

Table 5.1 – Average accumulated reward over 1000 episodes for different maze sizes for agents trained for N=8.

Additionally, we investigated the selectivity of neurons of MetODS plastic weight layer with respect to spatial location, to investigate whether

there could be emergence of activation patterns resembling those of grid cells found in the mammalian entorhinal cortex. We measured the selectivity of each neuron for specific agent locations by measuring the average activation rate normalized by the number of agent passages in different maze configurations. Interestingly, we found that without any explicit regularization, some neurons displayed sparse activation and consistent remapping between maps.

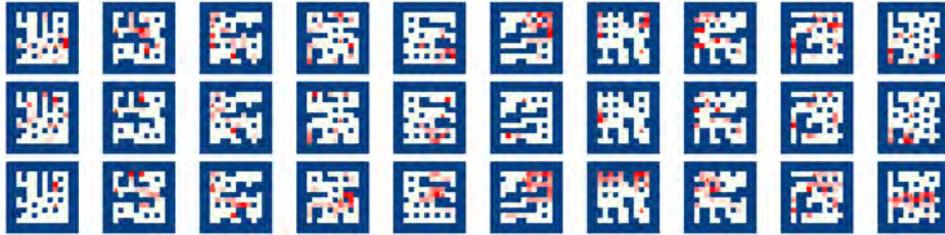


Figure 5.7 – Heatmap of activation rate of three neurons of a trained model across 10 different maze maps normalized by agent passage. Activation patterns show selectivity to particular positions within an episode and strong remapping across maze configurations.

◇ **Ablation study and variations:** We explored the contribution of the different features combined in MetODS update rule, showing that they all contribute to the final performance of our meta-learning model. First, we tested the importance of the element-wise tuning of plasticity in weight-based learning models and note that while it adversely affects MAML gradient update, it greatly improves MetODS performance, suggesting different forms of weights update. Second, we verified that augmenting recursivity depth S was beneficial to performance, consistently with Harlow results. Third, we noted that transforming the rightmost vector of the writing equation in 5.4 with a linear projection (*Linear* in figure 5.5, see S.I for full experimental details) yields major improvement while non-linear (*MLP*) does not improve performance. Finally, we additionally test the capability of the learnt navigation skills to generalize to a larger maze size of 10×10 pix. unseen during training. We show that MetODS is able to retain its advantage (see figure 5.5 and table 5.6 for results).

5.5.5 Generality : Motor control

Finally, we test the generality of the reinforcement learning program learnt by our model for different continuous control tasks:

◇ **MetaWorld:** First, we use the dexterous manipulation benchmark proposed in (Yu u. a., 2019) using the benchmark suite (?), in which a Sawyer robot is tasked with diverse operations. A full adaptation episode consists in $N=10$ rollouts of 500 timesteps of the same task across which dynamic weights are carried over. Observation consists in the robot’s joint angles and velocities, and the actions are its joint torques. We compare MetODS with baseline methods in terms of meta-training and meta-testing success rate for 3 settings, *push*, *reach* and *ML-10*. We show in Fig. 7.1 the meta-training results for all the methods in the MetaWorld environments. Due to computational resource constraints, we restrict our experiment to a budget of 10M steps per run. While we note that our benchmark does not reflect the final performance of previous methods reported in (Yu u. a., 2019) at 300M steps, we note that MetODS test performance outperforms these methods early in training and keeps improving at 10M steps, potentially leaving room for improvement with additional training. (see S.I for additional discussion). Finally, we note that all tested approaches performed modestly on ML10 for test tasks, which highlights the limitation of current methods. We conjecture that this might be due to the absence of inductive bias for sharing knowledge between tasks or fostering systematic exploration of the environment of the tested meta-learning algorithms.

◇ **Experiments:** In order to run experiments and baseline of MetaWorld (Yu u. a., 2019), we used the training routines and functions offered in the python library GARAGE (?). We used PPO (Schulman u. a., 2017) for training our model, with clip range $r = 0.2$ and 10 inner optimization steps maximum, for its good empirical performance and in order to compare with other methods with recurrent computation scheme. We kept the experimental settings from the original paper with $N=10$ episodes of 500 time-steps and sampled proximal gradient update in a meta-batch size of 25 trajectories. To estimate value function, we both tested a feed-forward neural network trained at each iteration of the meta-training or an additional MetODS network, which resulted in similar training results. We ran two different types of experiments:

1. **ML1:** In this setting, we restrict the task distribution to a single type of robotic manipulation while varying the goal location. The meta-training “tasks” corresponds to 50 random initial agent and goal positions, and meta-testing to 50 heldout positions. We tested our model on the *reach-v2* and *push-v2* tasks.
2. **ML10:** This set-up tests generalization to new manipulation tasks, the benchmark provides 10 training tasks and holds out 5 meta-

testing tasks.

◊ **Robot impairment:** We also tested the robustness of MetODS learnt reinforcement programs by evaluating the agent ability to perform in a setting not seen during training: specifically, when partially impairing the agent motor capabilities. We adopt the same experimental setting as section 5.5.1 for the *Ant* and *Cheetah* robots and evaluate the performance when freezing one of the robots torque. We show that our model policy retains a better proportion of its performance compared to other approaches. These results suggest that fast synaptic dynamics are not only better suited to support fast adaptation of a motor policy in the continuous domain, but also implement a more robust reinforcement learning program when impairing the agent motor capabilities.

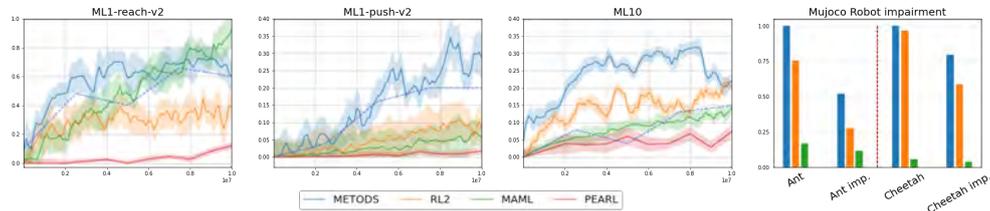


Figure 5.8 – **Left** Meta-training results for MetaWorld benchmarks. Subplots show tasks success rate over training timesteps. Average meta-test results for MetODS is shown in dotted line. **Right** Cumulative reward of the *Ant* and *Cheetah* directional locomotion task. For each condition, results are normalized against the best performing policy

5.6 Optimization

Defining the weight parameters \mathbf{W} of MetODS as dynamic variables lifts the optimization problem (7.4) into a functional space of control functions parameterized by θ . Hence, meta-optimizing the control necessitates the estimation of gradients with respect to θ over the space \mathbb{T} and for any possible trajectory π_t in Π . Interestingly, previous meta RL approaches have performed policy gradient optimization by sampling a single policy trajectory $\pi_t \sim \mathcal{M}_\theta(\tau)$ over M multiple tasks, showing that it is sufficient to obtain correct gradient estimates on θ . We proceed in the same way, by estimating the gradient policy update integrated over the space of tasks as mini-batches over tasks.

$$\frac{\partial}{\partial \theta} \mathbb{E}_{\tau \sim \mu_{\mathbb{T}}} \left[\mathbb{E}_{\pi \sim \mu_{\pi}^{\theta, \tau, t}} [\mathcal{R}(\tau, \pi)] \right] \approx \sum_{\tau_1, \dots, \tau_n} \sum_{t=0}^T \frac{\partial \log \pi_t(\mathbf{a}_t | \mathbf{W}_t, \theta)}{\partial \theta} r_{\tau_i}(\mathbf{a}_t, \mathbf{s}_t) \quad (5.6)$$

Additionally, the memory cost of storing synaptic weights trajectories instead of hidden activity in a network of N neurons is $\mathcal{O}(N^2)$ instead of $\mathcal{O}(N)$. This might lead to prohibitively large memory requirements for training with BPTT (Rumelhart u. a., 1985) over long episodes. We present in S.I an alternative solution to train the model through the discrete adjoint sensitivity method, leveraging the work of (Betancourt u. a., 2020) yielding a memory cost of $\mathcal{O}(1)$. The agent’s log-policy total derivative with respect to $\boldsymbol{\theta}$ can be computed as the solution of an augmented adjoint problem (Chalvidal u. a., 2021).

5.6.1 Gradient policy update

We define the evolution of the agent policy in task τ up to step T as the stochastic policy process $(\pi_t)_{t \leq T}$ in the space of policy Π with measure $\mu_{\pi}^{\boldsymbol{\theta}, \tau, 1 \dots T}$ and write $(\mathbf{W}_t)_{t \leq T}$ the trajectory of weights, such that:

$$(\pi_t)_{t \leq T} = (\pi_t(\cdot | \mathbf{W}_t, \mathbf{s}_t, \boldsymbol{\theta}))_{t \leq T} \sim \mu_{\pi}^{\boldsymbol{\theta}, \tau, 1 \dots T} \quad (5.7)$$

We also recall the definition of $\mathcal{R}(\tau, \pi)$ as the average accumulated reward under the realisation of $(\pi_t)_{t \leq T}$ for task τ . Then, The policy gradient update used to train MetODS can be written as the average gradient of $\boldsymbol{\theta}$ with respect to $\mu_{\mathbb{T}}$ and $\mu_{\pi}^{\boldsymbol{\theta}, \tau, 1 \dots T}$:

$$\frac{\partial}{\partial \boldsymbol{\theta}} \mathbb{E}_{\tau \sim \mu_{\mathbb{T}}} \left[\mathbb{E}_{\pi \sim \mu_{\pi}^{\boldsymbol{\theta}, \tau, 1 \dots T}} [\mathcal{R}(\tau, \pi)] \right] = \mathbb{E}_{\tau \sim \mu_{\mathbb{T}}} \left[\sum_{t=0}^T \frac{\partial}{\partial \boldsymbol{\theta}} \mathbb{E}_{\pi_t, \mathcal{P}_{\tau}} [r_{\tau}(\mathbf{a}_t, \mathbf{s}_t)] \right] \quad (5.8)$$

Then for any $t \leq T$, we can rewrite the gradient of the average, using the log-policy trick:

$$\frac{\partial}{\partial \boldsymbol{\theta}} \mathbb{E}_{\pi_t, \mathcal{P}_{\tau}} [r_{\tau}(\mathbf{a}_t, \mathbf{s}_t)] = \mathbb{E}_{\pi_t, \mathcal{P}_{\tau}} \left[r_{\tau}(\mathbf{a}_t, \mathbf{s}_t) \frac{\partial \log \pi_t(\mathbf{a}_t^{\tau} | \mathbf{W}_t, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right] \quad (5.9)$$

As specified in section 3.3, we do not sample over the policy distribution π_t and probability transition \mathcal{P}_{τ} to estimate the inner expectation in (5.22). Instead, we rely on a single evaluation, which yield, combining (5.20) with (5.21):

$$\frac{\partial}{\partial \boldsymbol{\theta}} \mathbb{E}_{\tau \sim \mu_{\mathbb{T}}} \left[\mathbb{E}_{\pi \sim \mu_{\pi}^{\boldsymbol{\theta}, \tau, 1 \dots T}} [\mathcal{R}(\tau, \pi)] \right] \approx \sum_{t=0}^T \mathbb{E}_{\tau \sim \mu_{\mathbb{T}}} \left[r_{\tau}(\mathbf{a}_t, \mathbf{s}_t) \frac{\partial \log \pi_t(\mathbf{a}_t^{\tau} | \mathbf{W}_t, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right] \quad (5.10)$$

By sampling over tasks, this last equation allows us to write the following gradient estimator

$$\frac{\partial}{\partial \boldsymbol{\theta}} \mathbb{E}_{\tau \sim \mu_T} \left[\mathbb{E}_{\pi \sim \mu_{\pi}^{\boldsymbol{\theta}, \tau, 1 \dots T}} [\mathcal{R}(\tau, \pi)] \right] \approx \frac{1}{M} \sum_{i=0}^M \sum_{t=0}^T \frac{\partial \log \pi_t(\mathbf{a}_t^{\tau_i} | \mathbf{W}_t, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} r(\mathbf{a}_t^{\tau_i}, \mathbf{s}_t^{\tau_i}) \quad (5.11)$$

5.6.2 Discrete adjoint system

With the previous notation, let us define the total gradient function $\frac{\partial \mathcal{J}}{\partial \boldsymbol{\theta}}$ as the gradient of our objective function:

$$\frac{\partial \mathcal{J}}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}) = \frac{1}{M} \sum_{t=0}^T \sum_{i=0}^M \frac{\partial \log \pi_t(\mathbf{a}_t^{\tau_i} | \mathbf{W}_t, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} r(\mathbf{a}_t^{\tau_i}, \mathbf{s}_t^{\tau_i}) \quad (5.12)$$

To clarify, we shall introduce the intermediary cost notation:

$$c_t(\mathbf{W}, \boldsymbol{\theta}, t) = \sum_{i=0}^M \log \pi_t(\mathbf{a}_t^{\tau_i} | \mathbf{W}_t, \boldsymbol{\theta}) r(\mathbf{a}_t^{\tau_i}, \mathbf{s}_t^{\tau_i}) \quad (5.13)$$

and we note the following update equation

$$\mathbf{W}_{t+1} = \delta(\mathbf{W}_t, \boldsymbol{\theta}) \quad (5.14)$$

This identify a discrete dynamical system with finite sum and differentiable cost, whose gradient can be computed mediating the introduction of an *adjoint* dynamical system presented in section 2 of (Betancourt u. a., 2020). Defining $(\boldsymbol{\nu}_t)_{t \leq T}$ the adjoint sequence, the general gradient equation can be computed as:

$$\frac{\partial \mathcal{J}}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}) = \left[\frac{\partial c_0}{\partial \mathbf{W}_0} - \boldsymbol{\nu}_0 \cdot \frac{\partial \delta}{\partial \mathbf{W}_0} - \boldsymbol{\nu}_0 \right]^\dagger \frac{\partial \mathbf{W}_0}{\partial \boldsymbol{\theta}} + \sum_{i=1}^T \left[\frac{\partial c_t}{\partial \boldsymbol{\theta}} - \boldsymbol{\nu}_t^\dagger \cdot \frac{\partial \delta}{\partial \boldsymbol{\theta}} \right] \quad (5.15)$$

Applying formula (5.27) to MetODS yields the following gradient formula:

$$\frac{\partial \mathcal{J}}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}) = \left[\frac{\partial c_0}{\partial \mathbf{W}_0} - \boldsymbol{\nu}_0 \right] + \sum_{i=1}^T \left[\frac{\partial c_t}{\partial \boldsymbol{\theta}} - \boldsymbol{\nu}_t^\dagger \cdot \frac{\partial \delta}{\partial \boldsymbol{\theta}} \right] \quad (5.16)$$

where $(\boldsymbol{\nu}_t)_{t \leq T}$ follows the following update rule backwards:

$$\begin{cases} \boldsymbol{\nu}_T = 0 \\ \boldsymbol{\nu}_{t-1} = \boldsymbol{\nu}_t - \frac{\partial c_t}{\partial \mathbf{W}_t} \end{cases} \quad (5.17)$$

5.7 Experimental details

General information: As specified in section 3 and 5, we test a single model definition for all experiments in this work, with one layer of dynamic weights \mathbf{W}_t . This layer consists in a dense matrix of size $\mathbf{n} \in \mathbb{N}$ with learnt or random initialization. Our model lightweight parametrization of the synaptic update rule makes it a very parameter efficient technique, which can perform batch-computation and be ported to GPU hardware to accelerate training. We refer readers to table 5.2 for specific details of each experiment presented in section 5. In addition to this *pdf* file, we provide the code for training METODS agents in experiments presented in section 5 at <https://>

Feature - Experiment	Harlow	Gym Mujoco	Maze	Meta-World
Dynamic layer size \mathbf{n}	20	100	200	100
Input size \mathbf{i}	12	134 (Ant) / 27 (Cheetah)	15	i
Output size \mathbf{o}	2	8 (Ant)/ 6 (Cheetah)	4	8
Embedding \mathbf{f}	$[\mathbf{i} \times 32, \sigma, 32 \times \mathbf{n}]$	$[\mathbf{i} \times 64, \sigma, 64 \times \mathbf{n}]$	$[\mathbf{i} \times 32, \sigma, 32 \times \mathbf{n}]$	$[\mathbf{i} \times 32, \sigma, 32 \times 64, \sigma, 64 \times \mathbf{n}]$
Read-out \mathbf{g}	$[\mathbf{n} \times 32, \sigma, 32 \times \mathbf{o}]$	$[\mathbf{n} \times 64, \sigma, 64 \times \mathbf{o}]$	$[\mathbf{n} \times 32, \sigma, 32 \times \mathbf{o}]$	$[\mathbf{n} \times 64, \sigma, 64 \times \mathbf{o}]$
Non-linearities	tanh	tanh	tanh	tanh
Init. \mathbf{W}_0	$\mathcal{N}(0, 1e-3)$	Learned	Learned	Learned
# Num. of episodes	5 (max)	1	1	10
Length of 1 episode	250 (max)	200	100	500
Meta-training Alg.	A2C	A2C	A2C	PPO
learning rate	5e-4	1e-4	5e-4	5e-4
meta-batch-size	50	50	20	25
discount factor λ	9e-1	9e-1	9.9e-1	9.9e-1
gae	1.	9.5E-1	9.5e-1	9.5e-1
value function coeff.	4e-1	4e-1	4e-1	-
entropy reg. factor	3e-2	3e-2	1e-2	1e-2

Table 5.2 – Summary of training hyper-parameters for the four experiments presented in this work.

Meta-training algorithm: We show in our experiments, that the all meta-parameters $\boldsymbol{\theta} = [\boldsymbol{\alpha}, \boldsymbol{\kappa}, \boldsymbol{\beta}, \mathbf{f}, \mathbf{g},]$ can be jointly optimized with two policy gradient algorithms. We use policy gradient methods to meta-train the synaptic parameters: Advantage Actor-critic algorithm (A2C) (Mnih u. a., 2016) where we consider that temporal dependancies are crucial to solve the task, and we use PPO (Schulman u. a., 2017) as a sequential policy optimization over fixed rollouts for the motor control experiment. Additionally, to reduce noise in policy gradient updates, we further show that it is possible to learn a dynamic advantage estimate of the Generalized Advantage Estimation (GAE) (Schulman u. a., 2018) and that we can meta-learn it as a second head of the MetODS layer output $\mathbf{v}^{(s)}$. This fact confirms that tuned hebbian-updates are also a sufficient mechanism to keep track of a policy value estimate.

Plasticity parameters: In all our experiments, α consists in a real-valued matrix of $\mathbb{R}^{n \times n}$ initialized with independent normal distribution $\mathcal{N}(\mu = 0, \sigma = 1e-3)$. Similarly, multi-step weighting parameters $(\kappa_s^{(l)})_{l \leq s}$ and $(\beta_s^{(l)})_{l \leq s}$ can be stored as entries of triangular inferior matrices of $\mathbb{R}^{S \times S}$ and are initialized with $\mathcal{N}(\mu = 0, \sigma = 1e-2)$.

Embedding and read-out: At each time-steps, inputs to the feed-forward embedding function \mathbf{f} consist in a concatenation of new observable/state \mathbf{s}_t as well as previous action and reward \mathbf{a}_{t-1} and \mathbf{r}_{t-1} . After adaptation procedure, policy is read-out from the last activation vector $\mathbf{v}^{(S)}$ by a feed-forward function \mathbf{g} which output statistics of a parameterized distribution in the action space (categorical with Softmax normalization in the discrete case or Gaussian in the continuous case). Both input and output mappings \mathbf{f} and \mathbf{g} consist in 2-layer Perceptrons with dense connections initialized with orthogonal initialization and hyperbolic tangent non-linearities.

5.7.1 METODS Optimization

Defining the weight parameters \mathbf{W} of MetODS as dynamic variables lifts the optimization problem (7.4) into a functional space of control functions parameterized by θ . Hence, meta-optimizing the control necessitates the estimation of gradients with respect to θ over the space \mathbb{T} and for any possible trajectory π_t in Π . Interestingly, previous meta RL approaches have performed policy gradient optimization by sampling a single policy trajectory $\pi_t \sim \mathcal{M}_\theta(\tau)$ over M multiple tasks, showing that it is sufficient to obtain correct gradient estimates on θ . We proceed in the same way, by estimating the gradient policy update integrated over the space of tasks as mini-batches over tasks.

$$\frac{\partial}{\partial \theta} \mathbb{E}_{\tau \sim \mu_{\mathbb{T}}} \left[\mathbb{E}_{\pi \sim \mu_{\pi}^{\theta, \tau, t}} [\mathcal{R}(\tau, \pi)] \right] \approx \sum_{\tau_1, \dots, \tau_n} \sum_{t=0}^T \frac{\partial \log \pi_t(\mathbf{a}_t | \mathbf{W}_t, \theta)}{\partial \theta} r_{\tau_t}(\mathbf{a}_t, \mathbf{s}_t) \quad (5.18)$$

Additionally, the memory cost of storing synaptic weights trajectories instead of hidden activity in a network of N neurons is $\mathcal{O}(N^2)$ instead of $\mathcal{O}(N)$. This might lead to prohibitively large memory requirements for training with BPTT (Rumelhart u. a., 1985) over long episodes. We present in S.I an alternative solution to train the model through the discrete adjoint sensitivity method, leveraging the work of (Betancourt u. a., 2020) yielding a memory cost of $\mathcal{O}(1)$. The agent’s log-policy total derivative with respect to θ can be computed as the solution of an augmented adjoint problem (Chalvidal u. a., 2021).

5.7.2 Gradient policy update

We define the evolution of the agent policy in task τ up to step T as the stochastic policy process $(\pi_t)_{t \leq T}$ in the space of policy Π with measure $\mu_\pi^{\theta, \tau, 1 \dots T}$ and write $(\mathbf{W}_t)_{t \leq T}$ the trajectory of weights, such that:

$$(\pi_t)_{t \leq T} = (\pi_t(\cdot | \mathbf{W}_t, \mathbf{s}_t, \boldsymbol{\theta}))_{t \leq T} \sim \mu_\pi^{\theta, \tau, 1 \dots T} \quad (5.19)$$

We also recall the definition of $\mathcal{R}(\tau, \pi)$ as the average accumulated reward under the realisation of $(\pi_t)_{t \leq T}$ for task τ . Then, The policy gradient update used to train MetODS can be written as the average gradient of $\boldsymbol{\theta}$ with respect to μ_π and $\mu_\pi^{\theta, \tau, 1 \dots T}$:

$$\frac{\partial}{\partial \boldsymbol{\theta}} \mathbb{E}_{\tau \sim \mu_\tau} \left[\mathbb{E}_{\pi \sim \mu_\pi^{\theta, \tau, 1 \dots T}} [\mathcal{R}(\tau, \pi)] \right] = \mathbb{E}_{\tau \sim \mu_\tau} \left[\sum_{t=0}^T \frac{\partial}{\partial \boldsymbol{\theta}} \mathbb{E}_{\pi_t, \mathcal{P}_\tau} [r_\tau(\mathbf{a}_t, \mathbf{s}_t)] \right] \quad (5.20)$$

Then for any $t \leq T$, we can rewrite the gradient of the average, using the log-policy trick:

$$\frac{\partial}{\partial \boldsymbol{\theta}} \mathbb{E}_{\pi_t, \mathcal{P}_\tau} [r_\tau(\mathbf{a}_t, \mathbf{s}_t)] = \mathbb{E}_{\pi_t, \mathcal{P}_\tau} \left[r_\tau(\mathbf{a}_t, \mathbf{s}_t) \frac{\partial \log \pi_t(\mathbf{a}_t^\tau | \mathbf{W}_t, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right] \quad (5.21)$$

As specified in section 3.3, we do not sample over the policy distribution π_t and probability transition \mathcal{P}_τ to estimate the inner expectation in (5.22). Instead, we rely on a single evaluation, which yield, combining (5.20) with (5.21):

$$\frac{\partial}{\partial \boldsymbol{\theta}} \mathbb{E}_{\tau \sim \mu_\tau} \left[\mathbb{E}_{\pi \sim \mu_\pi^{\theta, \tau, 1 \dots T}} [\mathcal{R}(\tau, \pi)] \right] \approx \sum_{t=0}^T \left[r_\tau(\mathbf{a}_t, \mathbf{s}_t) \frac{\partial \log \pi_t(\mathbf{a}_t^\tau | \mathbf{W}_t, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right] \quad (5.22)$$

By sampling over tasks, this last equation allows us to write the following gradient estimator

$$\frac{\partial}{\partial \boldsymbol{\theta}} \mathbb{E}_{\tau \sim \mu_\tau} \left[\mathbb{E}_{\pi \sim \mu_\pi^{\theta, \tau, 1 \dots T}} [\mathcal{R}(\tau, \pi)] \right] \approx \frac{1}{M} \sum_{i=0}^M \sum_{t=0}^T \frac{\partial \log \pi_t(\mathbf{a}_t^{\tau_i} | \mathbf{W}_t, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} r(\mathbf{a}_t^{\tau_i}, \mathbf{s}_t^{\tau_i}) \quad (5.23)$$

5.7.3 Discrete adjoint system

With the previous notation, let us define the total gradient function $\frac{\partial \mathcal{J}}{\partial \boldsymbol{\theta}}$ as the gradient of our objective function:

$$\frac{\partial \mathcal{J}}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}) = \frac{1}{M} \sum_{t=0}^T \sum_{i=0}^M \frac{\partial \log \pi_t(\mathbf{a}_t^{\tau_i} | \mathbf{W}_t, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} r(\mathbf{a}_t^{\tau_i}, \mathbf{s}_t^{\tau_i}) \quad (5.24)$$

To clarify, we shall introduce the intermediary cost notation:

$$c_t(\mathbf{W}, \boldsymbol{\theta}, t) = \sum_{i=0}^M \log \pi_t(\mathbf{a}_t^{\tau_i} | \mathbf{W}_t, \boldsymbol{\theta}) r(\mathbf{a}_t^{\tau_i}, \mathbf{s}_t^{\tau_i}) \quad (5.25)$$

and we note the following update equation

$$\mathbf{W}_{t+1} = \delta(\mathbf{W}_t, \boldsymbol{\theta}) \quad (5.26)$$

This identify a discrete dynamical system with finite sum and differentiable cost, whose gradient can be computed mediating the introduction of an *adjoint* dynamical system presented in section 2 of (Betancourt u. a., 2020) . Defining $(\boldsymbol{\nu}_t)_{t \leq T}$ the adjoint sequence, the general gradient equation can be computed as:

$$\frac{\partial \mathcal{J}}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}) = \left[\frac{\partial c_0}{\partial \mathbf{W}_0} - \boldsymbol{\nu}_0 \cdot \frac{\partial \delta}{\partial \mathbf{W}_0} - \boldsymbol{\nu}_0 \right]^\dagger \frac{\partial \mathbf{W}_0}{\partial \boldsymbol{\theta}} + \sum_{i=1}^T \left[\frac{\partial c_t}{\partial \boldsymbol{\theta}} - \boldsymbol{\nu}_t^\dagger \cdot \frac{\partial \delta}{\partial \boldsymbol{\theta}} \right] \quad (5.27)$$

Applying formula (5.27) to MetODS yields the following gradient formula:

$$\frac{\partial \mathcal{J}}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}) = \left[\frac{\partial c_0}{\partial \mathbf{W}_0} - \boldsymbol{\nu}_0 \right] + \sum_{i=1}^T \left[\frac{\partial c_t}{\partial \boldsymbol{\theta}} - \boldsymbol{\nu}_t^\dagger \cdot \frac{\partial \delta}{\partial \boldsymbol{\theta}} \right] \quad (5.28)$$

where $(\boldsymbol{\nu}_t)_{t \leq T}$ follows the following update rule backwards:

$$\begin{cases} \boldsymbol{\nu}_T = 0 \\ \boldsymbol{\nu}_{t-1} = \boldsymbol{\nu}_t - \frac{\partial c_t}{\partial \mathbf{W}_t} \end{cases} \quad (5.29)$$

General information: As specified in section 3 and 5, we test a single model definition for all experiments in this work, with one layer of dynamic weights \mathbf{W}_t . This layer consists in a dense matrix of size $\mathbf{n} \in \mathbb{N}$ with learnt or random initialization. Our model lightweight parametrization of the synaptic update rule makes it a very parameter efficient technique, which can perform batch-computation and be ported to GPU hardware to accelerate training. We refer readers to table 5.2 for specific details of each experiment presented in section 5. In addition to this *pdf* file, we provide the code for training METODS agents in experiments presented in section 5 at <https://>

Meta-training algorithm: We show in our experiments, that the all meta-parameters $\boldsymbol{\theta} = [\boldsymbol{\alpha}, \boldsymbol{\kappa}, \boldsymbol{\beta}, \mathbf{f}, \mathbf{g},]$ can be jointly optimized with two policy gradient algorithms. We use policy gradient methods to meta-train the synaptic parameters: Advantage Actor-critic algorithm (A2C)

(Mnih u. a., 2016) where we consider that temporal dependencies are crucial to solve the task, and we use PPO (Schulman u. a., 2017) as a sequential policy optimization over fixed rollouts for the motor control experiment. Additionally, to reduce noise in policy gradient updates, we further show that it is possible to learn a dynamic advantage estimate of the Generalized Advantage Estimation (GAE) (Schulman u. a., 2018) and that we can meta-learn it as a second head of the MetODS layer output $\mathbf{v}^{(s)}$. This fact confirms that tuned hebbian-updates are also a sufficient mechanism to keep track of a policy value estimate.

Plasticity parameters: In all our experiments, $\boldsymbol{\alpha}$ consists in a real-valued matrix of $\mathbb{R}^{n \times n}$ initialized with independent normal distribution $\mathcal{N}(\mu = 0, \sigma = 1e-3)$. Similarly, multi-step weighting parameters $(\boldsymbol{\kappa}_s^{(l)})_{l \leq s}$ and $(\boldsymbol{\beta}_s^{(l)})_{l \leq s}$ can be stored as entries of triangular inferior matrices of $\mathbb{R}^{S \times S}$ and are initialized with $\mathcal{N}(\mu = 0, \sigma = 1e - 2)$.

Embedding and read-out: At each time-steps, inputs to the feed-forward embedding function \mathbf{f} consist in a concatenation of new observable/state \mathbf{s}_t as well as previous action and reward \mathbf{a}_{t-1} and \mathbf{r}_{t-1} . After adaptation procedure, policy is read-out from the last activation vector $\mathbf{v}^{(S)}$ by a feed-forward function \mathbf{g} which output statistics of a parameterized distribution in the action space (categorical with Softmax normalization in the discrete case or Gaussian in the continuous case). Both input and output mappings \mathbf{f} and \mathbf{g} consist in 2-layer Perceptrons with dense connections initialized with orthogonal initialization and hyperbolic tangent non-linearities.

5.8 Discussion

5.8.1 Synthesis

In this chapter, we introduce a novel meta-RL system, MetODS, which leverages a self-referential weight update mechanism for rapid specialization at the episodic level. Our approach is generic and supports discrete and continuous domains, giving rise to a promising repertoire of skills such as one-shot adaptation, spatial navigation or motor coordination. MetODS demonstrates that locally tuned synaptic updates whose form depends directly on the network configuration can be meta-learned for solving complex reinforcement learning tasks. We conjecture that further tuning the hyperparameters as well as combining MetODS with more sophisticated reinforcement learning techniques can boost its performance. Generally, the success of the approach provides evidence for the benefits of fast plasticity in artificial neural networks, and the exploration of self-referential networks.

5.8.2 Outlook

Our work explores the emergence of reinforcement-learning programs through fast synaptic plasticity. The proposed pieces of evidence that information memorization and manipulation as well as behavioral specialization can be supported by such computational principles 1) help question the functional role of such mechanisms observed in biology and 2) reaffirms that alternative paradigms to gradient descent might exist for efficient artificial neural network control. Additionally, the proposed method is of interest for interactive machine learning systems that operate in quickly changing environments and under uncertainty (bayesian optimization, active learning and control theory). For instance, the meta RL approach proposed in this work could be applied to brain-computer interfaces for tuning controllers to rapidly drifting neural signals. Improving medical applications and robot control promises positive impact, however deeper theoretical understanding and careful deployment monitoring are required to avoid misuse.

Chapter 6

Discussion

Contents

6.1	Synthesis	138
6.2	Looking forward	140

6.1 Synthesis

Borrowing Marr’s levels of analysis (Marr, 2010), we conclude that despite the huge advancement that the field of artificial intelligence has recently known at the *algorithmic* (the what) and *implementational* (the how) levels, we haven’t fully discovered what *computational* principles (the why) can emulate the versatility and efficiency of natural learning. In this situation, the principle questions addressed in this thesis are:

- What are the limitations of the current principles used for deep neural networks control and can we qualify how they differ from their biological counterpart?
- Can we automatically discover such principles from the optimization of pretextual problems?

In this regard, this thesis has been organized in two parts, summarized as follows:

- We introduced the concept of artificial neural networks as particular dynamical systems useful in machine learning in order to bridge the large differences between their different previous algorithmic definitions, and presented the associated energy minimization problem

leveraged to operate functional control in their state-space potentially enabling higher-order functions regression with such systems.

- We present in the first part, two different models for learning higher-order functions with artificial neural networks. These two models are building on recent advances in machine learning (Neural ODEs and Transformer’s attention layer) while demonstrating the possibility of flexible functional learning with diverse applications from system identification to generative modeling.
 - We propose first a generalization of the Neural ODEs framework that augment the class of functions expressible with such framework while experimentally demonstrating that it can generally support rapid and efficient memorization, as well as accrued performance in supervised and unsupervised learning applications. We further show that it can bear interesting properties for controlling the generative process of complex data distributions with normalizing flows.
 - In the next chapter, we leverage the theory of reproducing kernel Banach space to learn a transductive regression operator that can associate to any dataset describing a function in the kernel space, a corresponding function estimate that can be queried on its input domain. We show that a deep Transformer-like network build on-the-fly such functional estimator by iteratively refining its estimate over network depth. Finally, we demonstrate that such estimator is more robust and present better generalization properties than inductive deep learning models.
- In the second part, we focus our discussion on the problem of meta-learning. In particular, we propose various mechanistic hypothesis to support efficient control in relation to this theory:
 - We present the problem of artificial meta-learning and draw a parallel to the principle of free-energy minimization set forth in [Friston \(2005\)](#); [Friston und Stephan \(2007\)](#); [Friston \(2010\)](#) by replacing the learning-to-learn abilities of biological agents as a byproduct of a more general process aiming at surprise minimization. Specifically, we emphasized the hierarchical and inferential nature of this theory and how it could help inform the design for synthetic and automatic learners.
 - We demonstrate in a second chapter that neuroscientific theories of fast synaptic plasticity can also constitute a possible

hypothesis space of meta-learners. Our model defines biologically plausible hebbian rules that are locally specialized to perform fast reinforcement learning in conjunction with an external reward signal. We show through extensive experimentation that such optimized rule can perform fast and efficient learning in different domains and generalize better to out-of-distribution test tasks.

- We finally conclude our discussion in the final chapter. We present a summary of our findings and outlines future potential research directions.

6.2 Looking forward

The interpretation of information manipulation and preference acquisition operated by artificial neural networks as a global thermodynamical process that this thesis emphasize constitutes a high-level and primitive basis for novel ideas and reinterpretations, which can potentially inspire new testable hypothesis in combination with the recent numerical advances in deep learning, but it also leaves many questions unanswered. We conclude by raising some of them below.

Expliciting the energy formulation One shared point of the the presented explorative projects is that despite not knowing the precise form of the energy function E that natural agents optimize when learning a specific task, it is possible to leave implicit the energy formulation describing the evolution of artificial networks, the same way classic supervised learning allows for fitting functions with no known analytical form. One fundamental question for artificial meta-learning will be to gain insight regarding such implicit forms in converged agents, in order to reverse-engineer the spontaneous mechanisms responsible for network adaptation that emerged during meta-optimization in order to build more precise and falsifiable theories of learning.

Designing learning curriculum While we particularly focused this thesis on providing a mechanistic description of original numerical principles able to support adaptation and learning in fixed setups, one putative key aspect of the development of natural learning abilities is the progressive evolution of complexity and abstraction of experience synthesis over an agent lifetime. Such dynamic aspect is likely to heavily influence the form of the resulting learning solutions as opposed to fixed curricula and remains an overlooked aspect of meta-learning, which usu-

ally assume a uniform distribution of tasks durring meta-training. In this context, understanding how the learning "sensitivity" of artificial agents unfold from progressively from experience refinement might again help inform our understanding of biological learning.

Learning measures to quantify learning Finally, one important missing aspect of meta-learning will be to propose systematic processes to measure the learning capacity of a system. At the moment, such capacity is measured in the litterature through experimental test of held-out tasks defined ad-hoc. Adopting standardized and generic measure of learning entails a more precise behaviorial and information theoretic definition that will be theoretically fruitful, while generating a heuristic pressure that will help design more generally able agents when deployed in open-ended environments.

Chapter 7

Résumé de la thèse en Français

Contents

7.1	Chapitre 1: Introduction	142
7.2	Chapitre 2 : Méta-apprentissage et le principe de l'énergie libre	144
7.3	Chapitre 3: Equations différentielles ordinaires neurales contrôlées	146
7.4	Chapitre 4: Méta-apprentissage par renforcement de réseaux self-référentiels	147
7.5	Chapitre 5: Méta-apprentissage par transduction fonctionnelle	148

7.1 Chapitre 1: Introduction

Cette thèse synthétise de manière cohérente différents projets visant à découvrir des principes computationnels dotant les réseaux neuronaux artificiels de capacités d'apprentissage automatique. Cette approche se situe à l'interface de l'apprentissage profond, des systèmes dynamiques et du méta-apprentissage, tout en s'inspirant de la neuroscience et de la théorie du contrôle. Je commence par présenter l'état de la recherche passée et actuelle sur les réseaux neuronaux artificiels sous la forme d'une note historique ainsi que son interaction avec ces deux domaines au **chapitre 1**, avant d'introduire plus spécifiquement le périmètre et les défis du méta-apprentissage neuronal du point de vue du principe de minimisation de l'énergie libre.

Essentiellement, l'informatique s'efforce de construire et de comprendre des machines automatiques dotées d'un comportement de plus en plus autonome. Ces dernières années, cette aspiration s'est concrétisée par un changement conceptuel clair, passant de la conception d'algorithmes spécifiés par l'homme à des solutions computationnelles se déployant automatiquement à partir de principes d'ordre supérieur. L'apprentissage automatique des réseaux de neurones a été à l'avant-garde de cette transformation en développant des approches centrées sur les données peu spécifiées aboutissant aux capacités de modélisation inégalées qui caractérisent le *deep learning*. Cependant, bien que le deep learning ait montré, des résultats inégalés pour l'intelligence artificielle, les générations actuelles de réseaux neuronaux fonctionnent à des échelles de temps fixes et à un faible niveau d'abstraction, sont très inefficaces en matière d'échantillonnage, suivent souvent des règles de décision peu intuitives et restent faiblement adaptatifs aux nouveaux contextes ayant des propriétés statiques tout au long de leur déploiement, nécessitant une intervention humaine spécifique pour leur réglage à des applications spécifiques.

Une solution potentielle aux défis auxquels sont confrontés les réseaux de neurones artificiels actuels consiste à *méta-apprendre* des mécanismes computationnels capables de capturer rapidement la structures des tâches et des distributions de données et de fonctionner automatiquement par un mécanisme de control rétroactif. Dans cette perspective, le méta-apprentissage artificiel neural constitue un champ prometteur pour construire de manière originale des systèmes adaptatifs et capables de combler les limitations précédemment discutées (complexité de l'échantillon, robustesse, généralisation...) (Clune, 2019) tout en permettant d'identifier certains mécanismes neuroscientifique conférant au cerveau humain ses remarquables capacités d'apprentissage (Botvinick u. a., 2019). Dans ce contexte, cette thèse vise à produire des modèles de meta-learning nouveaux, dont j'énumère ci-dessous les contributions notables:

- Au **chapitre 2**, nous proposons une revue des approches récentes de méta-apprentissage organisées en relation avec le notion de minimisation de l'énergie libre comme principe général pour l'apprentissage de l'intelligence naturelle et artificielle. Je présente en outre certaines directions potentielles de recherche ainsi que les défis scientifiques liés à ces systèmes de méta-apprentissage qui seront abordés dans les chapitres suivants.
- Le **chapitre 3** traite d'une généralisation des Neural ODEs (réseaux neuronaux en temps continu) avec paramétrisation dynamique. Cette

généralisation est associée aux notions classiques de contrôle en boucle ouverte et en boucle fermée et testée dans des applications d'apprentissage supervisé et non-supervisé.

- Le **chapitre 4** définit une classe original de modèle présentant une plasticité synaptique Hebbienne rapide inspirée de la neuroscience décrivant une forme de réseaux de Hopfield modernes dynamiques. Nous montrons que ce modèle peut implémenter un programme d'apprentissage par renforcement autonome grâce à travers la dynamique des poids, avec des applications à l'apprentissage one-shot, la navigation spatiale et le contrôle moteur robotique.
- Le **chapitre 5** propose une interprétation des capacités d'apprentissage "en-contexte" des architectures récentes de type Transformer du point de vue de la théorie des espaces de Banach à noyau reproduisant, ce qui conduit à la construction d'un cadre de méta-apprentissage basé sur des principes de transduction originaux permettant d'effectuer des régressions fonctionnelles ultra-rapides en dimension finie ou infinie.
- Enfin, le **chapitre 6** conclut notre discussion et souligne les futures prolongement potentiels de ce travail.

7.2 Chapitre 2 : Méta-apprentissage et le principe de l'énergie libre

Une caractéristique essentielle des êtres vivants est leur persistance homéostatique au fil du temps malgré l'exposition à des conditions environnementales variables et difficiles. Les agents biologiques maintiennent l'intégrité de leur structure potentiellement hautement complexe dans des limites physiologiques contre une tendance naturelle du monde vers des états désordonnés à haute entropie de configuration (Prigogine und Nicolis, 1985). À plusieurs échelles de temps, allant de l'expérience individuelle à la transmission générationnelle, la réalisation de cet objectif a déterminé l'émergence de comportements adaptatifs complexes, permettant de capturer l'information et de réagir de manière appropriée aux variations internes et externes de l'état du monde, et appelés de manière générale *apprentissage*.

Dans la plupart des espèces vivantes complexes dotées de capacités cognitives, l'apprentissage se manifeste au cours de la vie individuelle par une hiérarchie complexe de processus électro-chimiques inter-reliés se produisant dans le système nerveux. Nous définissons la séquence d'états

décrivant l'évolution de ces processus neuronaux (que nous décrirons plus loin) chez un seul agent comme $\tilde{\mathbf{s}}_t = \mathbf{s}_0, \dots, \mathbf{s}_t$, qui englobe largement la force de connexion entre les neurones, les quantités neuromodulatrices et l'activité neuronale elle-même. Nous définissons également $\tilde{\mathbf{z}}_t = \mathbf{z}_0, \dots, \mathbf{z}_t$, une séquence de variables génératives (potentiellement évolutives) qui cause la séquence d'observations de l'agent $\tilde{\mathbf{o}}_t = \mathbf{o}_0, \dots, \mathbf{o}_t$. Il a été proposé par Friston (2010) que de tels processus visent à minimiser une fonction d'*énergie libre* E qui mesure fondamentalement une "adéquation" entre $\tilde{\mathbf{o}}_t$ et les états internes $\tilde{\mathbf{s}}_t = \mathbf{s}_0, \dots, \mathbf{s}_t$ du système neuronal :

$$E(\tilde{\mathbf{o}}_t, \tilde{\mathbf{s}}_t) = -\mathbb{E} q_{\tilde{\mathbf{s}}_t}(\tilde{\mathbf{z}}_t) [\ln(p(\tilde{\mathbf{o}}_t, \tilde{\mathbf{z}}_t)) - \ln(q_{\tilde{\mathbf{s}}_t}(\tilde{\mathbf{z}}_t))] \quad (7.1)$$

Dans l'équation (7.1), $q_{\tilde{\mathbf{s}}_t}(\tilde{\mathbf{z}}_t)$ représente la fonction de densité sur les causes environnementales $\tilde{\mathbf{z}}_t$ qui est encodée par les états internes du système $\tilde{\mathbf{s}}_t$. Ainsi, en essence, l'hypothèse d'*énergie libre* E suppose que les variables neuronales $\tilde{\mathbf{s}}_t$ codent un modèle génératif du monde observé, c'est-à-dire un modèle probabiliste des dépendances entre les causes $\tilde{\mathbf{z}}_t$ et les observations conséquentes $\tilde{\mathbf{s}}_t$. De plus, l'*énergie libre* E est une borne supérieure sur la surprise de la collection d'états observés $\tilde{\mathbf{s}}_t$, qui est définie comme le log-négatif de l'évidence des états observés $\mathcal{S}(\tilde{\mathbf{o}}_t) = -\ln(\int \mathcal{Z} p(\tilde{\mathbf{o}}_t, \tilde{\mathbf{z}}_t) d\tilde{\mathbf{z}})$ par la relation suivante :

$$-\int_{\mathcal{Z}} q_{\tilde{\mathbf{s}}_t}(\tilde{\mathbf{z}}_t) \ln\left(\frac{p(\tilde{\mathbf{o}}_t, \tilde{\mathbf{z}}_t)}{q_{\tilde{\mathbf{s}}_t}(\tilde{\mathbf{z}}_t)}\right) d\tilde{\mathbf{z}} \underbrace{\geq}_{\text{Jensen}} -\ln\left(\int \mathcal{Z} q_{\tilde{\mathbf{s}}_t}(\tilde{\mathbf{z}}_t) \frac{p(\tilde{\mathbf{o}}_t, \tilde{\mathbf{z}}_t)}{q_{\tilde{\mathbf{s}}_t}(\tilde{\mathbf{z}}_t)} d\tilde{\mathbf{z}}\right) = \mathcal{S}(\tilde{\mathbf{o}}_t) \quad (7.2)$$

Ainsi, la minimisation de l'*énergie libre* dans l'équation (7.1) permet implicitement à l'agent de réduire la surprise contenue dans ses observations du monde. Pour effectuer une telle minimisation, les agents peuvent réduire l'énergie en faisant varier leur état neuronal interne $\tilde{\mathbf{s}}_t$ ou en faisant varier subséquemment leur information sensorielle $\tilde{\mathbf{o}}_t$. Nous notons, comme indiqué dans Friston (2010), que cette distinction de variables est également utile pour cartographier les capacités d'action et de perception de l'agent. Nous nous concentrerons sur une vue synthétique de la minimisation de l'état neuronal $\tilde{\mathbf{s}}_t$, car elle peut indirectement biaiser la politique de l'agent déterminant les observations $\tilde{\mathbf{o}}_t$.

Cette interprétation de l'apprentissage comme minimisation de la surprise nous permet de faire plusieurs remarques sur l'organisation du système neuronal afin de soutenir ces facultés d'apprentissage:

- Un aspect essentiel du principe de l'énergie libre repose sur l'idée bayésienne du cerveau en tant que "moteur d'inférence". En ce

sens, le principe de l'énergie libre **unifie sous une seule perspective thermodynamique la notion de contrôle et d'inférence bayésienne**.

- La caractérisation des différents mécanismes physiologiques qui régissent l'évolution de l'état $\tilde{\mathbf{s}}_t$ de systèmes neuronaux complexes au fil du temps en fonction des observations $\tilde{\mathbf{o}}_t$ et de leur propre état révèle une **hiérarchie de processus possédant différentes échelles temporelles** et s'influençant mutuellement.

Après une discussion sur les types d'approche précédemment proposée en méta-learning, nous identifions plusieurs axes de recherche principaux: comme le notion de contrôlabilité des réseaux, l'importance de modèles hiérarchiques, le rôle transverse de la mémoire, la notion d'apprentissage fonctionnel, l'importance d'une définition exacte de l'énergie libre et enfin la notion de curriculum pour l'apprentissage.

7.3 Chapitre 3: Equations différentielles ordinaires neurales contrôlées

Dans ce chapitre, nous présentons la formulation générique des équations différentielles ordinaires neurales (NODEs) elle-même contrôlées par un réseau de neurones (NCODEs). L'idée principale de NCODE est de considérer les paramètres $\boldsymbol{\theta}$ dans l'équation 2.1 comme des variables de contrôle dynamiques $\boldsymbol{\theta}(\mathbf{x}, t)$ pour l'état dynamique $\mathbf{x}(t)$. Cette généralisation est inspirée par plusieurs observations. Tout d'abord, les problèmes génériques de théorie du contrôle optimal (Liberzon, 2011), similaires à l'objectif d'apprentissage que nous introduirons dans la section 2.3.4, sont formulés sur des variables dynamiques $\boldsymbol{\theta}(\mathbf{x}, t)$ au lieu de paramètres statiques $\boldsymbol{\theta}$, ce qui permet de résoudre des problèmes de contrôle plus généraux et notamment de construire la notion de rétroaction, qui joue un rôle central pour l'apprentissage. Deuxièmement, ce développement fait écho à des recherches récentes concernant la méta-paramétrisation des réseaux profonds par l'intermédiaire de *hyper-réseaux* (Ha u. a., 2017) ainsi que de réseaux autoréférentiels (Irie u. a., 2022b). Ces idées explorent un angle mort de l'apprentissage profond basé sur les gradients, en considérant les paramètres de connexion des réseaux comme des processus dynamiques régis par des mécanismes hiérarchiques entraînaibles. Enfin, cela fait également écho à un principe connu en neuroscience computationnelle, la *plasticité synaptique à court terme* qui décrit les processus électrochimiques déterminant les variations rapides de l'efficacité synaptique entre les neurones biologiques, créant ainsi des

fluctuations dans la fonction de transfert exprimée par les réseaux biologiques à des échelles de temps rapides (Wu u. a., 2013). Nous verrons que NCODE fournit un cadre générique pour intégrer et explorer de telles idées dans la dynamique en temps continu. Nous notons également que les développements récents proposés pour les NODEs peuvent être combinés avec notre discussion, tels que les équations à retard (Zhu u. a., 2021), la paramétrisation polymorphe (Yoon u. a., 2022), les flux de gradients (Massaroli u. a., 2020a) ou la paramétrisation sur des variétés riemanniennes (Choromanski u. a., 2020b).

7.3.1 Formulation générale

Dans NCODE, la dynamique du modèle est gouvernée par un système couplé d'équations sur l'état augmenté $\mathbf{z}(t) = (\mathbf{x}(t), \boldsymbol{\theta}(t))$. Tout au long, nous supposons que le point de temps initial est $t = 0$. La valeur initiale des poids de contrôle, $\boldsymbol{\theta}(0)$, est donnée par une fonction $\gamma : \mathcal{X} \rightarrow \Theta$. La trajectoire complète des poids de contrôle, $\boldsymbol{\theta}(t)$, est ensuite produite par un contrôleur g , donné par une autre équation différentielle de mouvement différentiable $g : \Theta \times \mathcal{X} \times \mathbb{R} \mapsto \Theta$ avec une condition initiale $\gamma(\mathbf{x}_0)$. Étant donné un point initial, $\mathbf{x}(0)$, nous pouvons résoudre le problème de valeur initiale (IVP)

$$\begin{cases} \frac{d\mathbf{z}}{dt} = h(\mathbf{z}, t) \\ \mathbf{z}(0) = \mathbf{z}_0 \end{cases} = \begin{cases} \left(\frac{d\mathbf{x}}{dt}, \frac{d\boldsymbol{\theta}}{dt} \right) = (f(\mathbf{x}, \boldsymbol{\theta}, t), g(\boldsymbol{\theta}, \mathbf{x}, t)) \\ (\mathbf{x}(0), \boldsymbol{\theta}(0)) = (\mathbf{x}_0, \gamma(\mathbf{x}_0)) \end{cases} \quad (7.3)$$

Cette définition nous permet de démarquer les propriétés d'expressivité de NCODE par rapport au modèle classique (NODE):

Proposition 2 *There exists a function $\phi : \mathbb{R}^d \rightarrow \mathbb{R}^d$ which can be expressed by NCODE but not by NODEs. In particular, ϕ is not a homeomorphism.*

7.4 Chapitre 4: Meta-apprentissage par renforcement de réseaux self-référentiels

L'apprentissage des réseaux de neurones par renforcement produisent des agents spécialisés privés de mécanisme leur permettant de contin-

uer à apprendre au-delà de leur programme de formation. Au contraire, la plasticité synaptique biologique est persistante au cours de la vie de l'individu, et a été hypothétisée comme jouant un rôle clé dans les fonctions exécutives telles que la mémoire de travail et la flexibilité cognitive, potentiellement en soutenant des capacités d'apprentissage plus efficaces et génériques. Inspiré par cela, nous explorons dans ce chapitre une classe originale de réseaux de neurones à poids dynamiques, capables d'effectuer continuellement une modification autoréflexive en fonction de leur état synaptique actuel et de la rétroaction de boucles d'action-récompense, grâce à des principes de plasticité réglables au niveau des synapses. Le modèle résultant, MetODS (pour Meta-Optimized Dynamical Synapses), est un système de méta-apprentissage par renforcement largement applicable et capable d'apprendre des règles de contrôle efficaces dans l'espace des polices décisionnelles. Une seule couche avec des synapses dynamiques peut effectuer un apprentissage en une seule étape, généraliser les principes de navigation à des environnements inconnus et démontrer une forte capacité à apprendre des politiques motrices adaptatives, se comparant favorablement à des approches précédentes d'apprentissage par renforcement méta.

En définissant θ comme les méta-paramètres régissant l'évolution des séquences (π_t) et $\mu^{\theta, \tau, t} \pi$ comme la mesure de distribution de la politique π_t après l'apprentissage de la tâche τ pendant une certaine période t , le problème d'optimisation consiste à trouver les méta-paramètres θ qui s'adaptent le mieux à $\pi_t \sim \mu^{\theta, \tau, t} \pi$ sur la distribution de tâches.

$$\max_{\theta}; \mathbb{E}_{\tau} \sim \mu \mathbb{T} \left[\mathbb{E}_{\pi} \sim \mu^{\theta, \tau, t} \pi [\mathcal{R}(\tau, \pi)] \right] \quad (7.4)$$

Nous présentons un mechanism original d'adaptation de la policy par dynamique synaptique dont l'algorithm peut être formalisé de la manière suivante:

Nous montrons de manière empirique que cet algorithm peut être entraîné a apprendre des taches de navigation dans l'espace ou de controle robotique.

7.5 Chapitre 5: Meta-apprentissage par transduction fonctionnelle

La recherche en apprentissage automatique s'est polarisée sur deux approches générales de régression : les méthodes transductives dérivent des estimations directement à partir des données disponibles mais sont généralement peu spécifiques au problème. Les méthodes inductives peu-

Algorithm 2 MetODS algorithm

- 1: **Require:** $\theta = [f, g, \alpha, \kappa, \beta]$ and W_0
- 2: **for** $1 \leq t \leq T$ **do**
- 3: $v^{(0)} \leftarrow f(s_t, a_{t-1}, r_{t-1})$
- 4: $W^{(0)} \leftarrow W_{t-1}$
- 5: **for** $1 \leq s \leq S$ **do**
- 6: $v^{(s)} \leftarrow \sum_{l=0}^{s-1} \kappa_s^{(l)} v^{(l)} + \kappa_s^{(s)} \sigma(W^{(s-1)} \cdot v^{(s-1)})$
- 7: $W^{(s)} \leftarrow \sum_{l=0}^{s-1} \beta_s^{(l)} W^{(l)} + \beta_s^{(s)} (\alpha \odot v^{(s-1)} \otimes v^{(s-1)})$
- 8: **end for**
- 9: $a_t, v_t \leftarrow g(v^{(s)})$
- 10: $W_t \leftarrow W^{(s)}$
- 11: **end for**

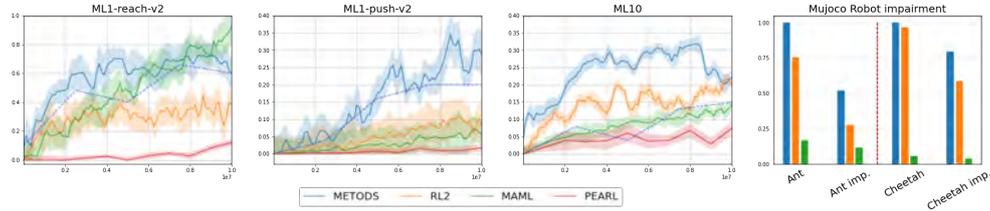


Figure 7.1 – **Left** Meta-training results for MetaWorld benchmarks. Subplots show tasks success rate over training timesteps. Average meta-test results for MetODS is shown in dotted line. **Right** Cumulative reward of the *Ant* and *Cheetah* directional locomotion task. For each condition, results are normalized against the best performing policy

vent être beaucoup plus spécifiques, mais nécessitent généralement un ajustement et des recherches intensives en calcul pour trouver des solutions. Dans ce chapitre, nous adoptons une approche hybride : nous exploitons la théorie des espaces de Banach à noyau reproduisant (RKBS) et montrons que les principes transductifs peuvent être induits par descente de gradient pour former des approximations neuronales efficaces *in-contexte*. Nous appliquons cette approche aux RKBS des opérateurs à valeurs de fonction et montrons que, une fois entraîné, notre modèle *Transducer* peut capturer les relations entre les fonctions d’entrée et de sortie de dimension infinie en temps réel, en utilisant quelques paires d’exemples, et renvoyer de nouvelles estimations de fonctions. Nous démontrons les avantages de notre approche transductive pour modéliser des systèmes physiques complexes influencés par des facteurs externes variables avec peu de données, à une fraction du coût habituel de calcul d’entraînement en apprentissage profond pour des applications de

modélisation d'équations aux dérivées partielles et de modélisation climatique.

Ce développement s'appuie sur la théorie des espaces de Banach à noyau reproduisant de [Zhang u. a. \(2009\)](#) et en particulier ce résultat:

Theorem 6 (Théorème du représentant RKBS) *Soit \mathcal{B} une RKBS à valeurs opérateurs de \mathcal{V} à \mathcal{U} . Alors le problème d'interpolation à norme minimale suivant :*

$$\inf_{\mathcal{O} \in \mathcal{B}} |\mathcal{O}|_{\mathcal{B}}, \text{ s.t. } \mathcal{O}(\mathbf{v}_i) = \mathbf{u}_i, i \leq I \quad (7.5)$$

admet un minimiseur unique \mathcal{O}_0 . De plus, \mathcal{O}_0 est la solution de (7.5) si et seulement si $\mathcal{O}_0(\mathbf{v}_i) = \mathbf{u}_i$ pour tout $i \leq I$ et $\mathcal{O}_0 \in \overline{\text{span}}\mathcal{K}(\mathbf{v}_i, \cdot)(\mathbf{u}), i \leq I, \mathbf{u} \in \mathcal{U}$.

Ce résultat permet d'interpréter les opérations d'auto-attention au sein du Transformer ([Vaswani u. a., 2017](#)) comme une approximation du kernel reproduisant, permettant ainsi d'entraîner un modèle à régresser de manière transductive, un jeu de données relatif à une certaine relation fonctionnelle. Nous expérimentons particulièrement dans le cas d'opérateurs définis comme allant d'un état initial à sa solution pour des équations aux dérivées partielles connues et démontrons la capacité de régression générales induites au cours du méta-apprentissage.

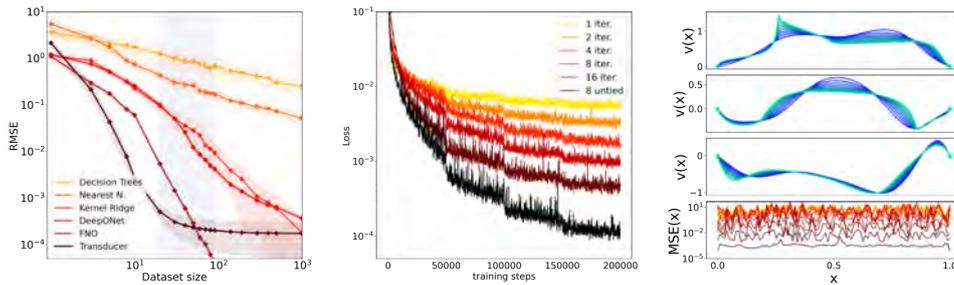


Figure 7.2 – **Gauche** : Erreur quadratique moyenne relative avec un intervalle de confiance à 90% sur les opérateurs inconnus en fonction de la taille de l’ensemble de données pour chaque méthode de régression. La zone grise correspond aux cardinalités d’ensemble de données vues lors de l’entraînement pour le *Transducer*. Toutes les autres méthodes sont entraînées à partir de zéro avec un nombre correspondant d’exemples originaux. **Milieu** : Évolution de la perte d’entraînement des *Transducers* avec différents nombres de transformations de noyau. Le fait de permettre plus d’itérations de transformation de noyau améliore les performances, les poids détachés offrant les meilleures performances. **Droite** : Haut - 3 exemples de l’évolution de l’état $s(x, t)$ en fonction de la dimension spatiale pour différentes équations ADR $E(\delta, \nu, \mathbf{k})$. Bas - Évolution de l’erreur quadratique moyenne des représentations intermédiaires (\mathbf{u}^ℓ) colorées par le numéro d’itération. Le profil décroissant ainsi que la tendance de performance observée avec un nombre croissant d’itérations suggèrent que la profondeur du réseau permet au modèle d’affiner progressivement son estimation de fonction.

Bibliography

- [LEE201431 2014] Tuning into diversity of homeostatic synaptic plasticity. In: *Neuropharmacology* 78 (2014), S. 31–37. – Homeostatic Synaptic Plasticity. – ISSN 0028-3908
- [Abbott und Nelson 2000] ABBOTT, Larry F. ; NELSON, Sacha B.: Synaptic plasticity: taming the beast. In: *Nature neuroscience* 3 (2000), Nr. 11, S. 1178–1183
- [Aggarwal u. a. 2001] AGGARWAL, Charu C. ; HINNEBURG, Alexander ; KEIM, Daniel A.: On the surprising behavior of distance metrics in high dimensional space. In: *International conference on database theory* Springer (Veranst.), 2001, S. 420–434
- [Alexander und Brown 2011] ALEXANDER, William H. ; BROWN, Joshua W.: Medial prefrontal cortex as an action-outcome predictor. In: *Nature neuroscience* 14 (2011), Nr. 10, S. 1338–1344
- [Amari 1967] AMARI, Shunichi: A theory of adaptive pattern classifiers. In: *IEEE Transactions on Electronic Computers* (1967), Nr. 3, S. 299–307
- [Andrychowicz u. a. 2016] ANDRYCHOWICZ, Marcin ; DENIL, Misha ; GÓMEZ, Sergio ; HOFFMAN, Matthew W. ; PFAU, David ; SCHAUL, Tom ; SHILLINGFORD, Brendan ; FREITAS, Nando de: Learning to learn by gradient descent by gradient descent. In: LEE, D. (Hrsg.) ; SUGIYAMA, M. (Hrsg.) ; LUXBURG, U. (Hrsg.) ; GUYON, I. (Hrsg.) ; GARNETT, R. (Hrsg.): *Advances in Neural Information Processing Systems* Bd. 29, Curran Associates, Inc., 2016
- [Ba u. a. 2016] BA, Jimmy L. ; KIROS, Jamie R. ; HINTON, Geoffrey E.: Layer normalization. In: *arXiv preprint arXiv:1607.06450* (2016)
- [Bahdanau u. a. 2014] BAHDANAU, Dzmitry ; CHO, Kyunghyun ; BENGIO, Yoshua: Neural machine translation by jointly learning to align and translate. In: *arXiv preprint arXiv:1409.0473* (2014)

- [Banino u. a. 2018] BANINO, Andrea ; BARRY, Caswell ; URIA, Benigno ; BLUNDELL, Charles ; LILICRAP, Timothy ; MIROWSKI, Piotr ; PRITZEL, Alexander ; CHADWICK, Martin J. ; DEGRIS, Thomas ; MODAYIL, Joseph u. a.: Vector-based navigation using grid-like representations in artificial agents. In: *Nature* 557 (2018), Nr. 7705, S. 429–433
- [Bartlett und Maass 2003] BARTLETT, Peter L. ; MAASS, Wolfgang: Vapnik-Chervonenkis dimension of neural nets. In: *The handbook of brain theory and neural networks* (2003), S. 1188–1192
- [Bartunov u. a. 2019] BARTUNOV, Sergey ; RAE, Jack ; OSINDERO, Simon ; LILICRAP, Timothy: Meta-Learning Deep Energy-Based Memory Models. In: *International Conference on Learning Representations*, 2019
- [Batchelor 2000] BATCHELOR, G. K.: *Kinematics of the Flow Field*. S. 71–130. In: *An Introduction to Fluid Dynamics*, Cambridge University Press, 2000 (Cambridge Mathematical Library)
- [Bauschke u. a.] BAUSCHKE, Heinz H. ; COMBETTES, Patrick L. u. a.: *Convex analysis and monotone operator theory in Hilbert spaces*. Bd. 408. Springer
- [Baxter 1998] BAXTER, Jonathan: *Theoretical Models of Learning to Learn*. S. 71–94. In: THRUN, Sebastian (Hrsg.) ; PRATT, Lorien (Hrsg.): *Learning to Learn*. Boston, MA : Springer US, 1998. – ISBN 978-1-4615-5529-2
- [Bellman 1966] BELLMAN, Richard: Dynamic programming. In: *Science* 153 (1966), Nr. 3731, S. 34–37
- [Betancourt u. a. 2020] BETANCOURT, Michael ; MARGOSSIAN, Charles C. ; LEOS-BARAJAS, Vianey: The Discrete Adjoint Method: Efficient Derivatives for Functions of Discrete Sequences. In: *arXiv preprint arXiv:2002.00326* (2020)
- [Bhattacharya u. a. 2020] BHATTACHARYA, Kaushik ; HOSSEINI, Bammad ; KOVACHKI, Nikola B. ; STUART, Andrew M.: *Model Reduction and Neural Networks for Parametric PDEs*. 2020
- [Bianchini und Scarselli 2014] BIANCHINI, Monica ; SCARSELLI, Franco: On the Complexity of Neural Network Classifiers: A Comparison Between Shallow and Deep Architectures. In: *IEEE Transactions on Neural Networks and Learning Systems* 25 (2014), Nr. 8, S. 1553–1565

- [Boser u. a. 1992] BOSER, Bernhard E. ; GUYON, Isabelle M. ; VAPNIK, Vladimir N.: A training algorithm for optimal margin classifiers. In: *Proceedings of the fifth annual workshop on Computational learning theory*, 1992, S. 144–152
- [Böttcher u. a. 2022] BÖTTCHER, Lucas ; ANTULOV-FANTULIN, Nino ; ASIKIS, Thomas: AI Pontryagin or how artificial neural networks learn to control dynamical systems. In: *Nature communications* 13 (2022), Nr. 1, S. 333
- [Botvinick u. a. 2019] BOTVINICK, Matthew ; RITTER, Sam ; WANG, Jane X. ; KURTH-NELSON, Zeb ; BLUNDELL, Charles ; HASSABIS, Demis: Reinforcement Learning, Fast and Slow. In: *Trends in Cognitive Sciences* 23 (2019), Nr. 5, S. 408–422. – ISSN 1364-6613
- [Botvinick u. a. 2020] BOTVINICK, Matthew ; WANG, Jane X. ; DABNEY, Will ; MILLER, Kevin J. ; KURTH-NELSON, Zeb: Deep Reinforcement Learning and Its Neuroscientific Implications. In: *Neuron* 107 (2020), Nr. 4, S. 603–616. – ISSN 0896-6273
- [Brown u. a. 2020a] BROWN, Tom ; MANN, Benjamin ; RYDER, Nick ; SUBBIAH, Melanie ; KAPLAN, Jared D. ; DHARIWAL, Prafulla ; NEELAKANTAN, Arvind ; SHYAM, Pranav ; SASTRY, Girish ; ASKELL, Amanda u. a.: Language models are few-shot learners. In: *Advances in neural information processing systems* 33 (2020), S. 1877–1901
- [Brown u. a. 2020b] BROWN, Tom B. ; MANN, Benjamin ; RYDER, Nick ; SUBBIAH, Melanie ; KAPLAN, Jared ; DHARIWAL, Prafulla ; NEELAKANTAN, Arvind ; SHYAM, Pranav ; SASTRY, Girish ; ASKELL, Amanda ; AGARWAL, Sandhini ; HERBERT-VOSS, Ariel ; KRUEGER, Gretchen ; HENIGHAN, Tom ; CHILD, Rewon ; RAMESH, Aditya ; ZIEGLER, Daniel M. ; WU, Jeffrey ; WINTER, Clemens ; HESSE, Christopher ; CHEN, Mark ; SIGLER, Eric ; LITWIN, Mateusz ; GRAY, Scott ; CHESS, Benjamin ; CLARK, Jack ; BERNER, Christopher ; MCCANDLISH, Sam ; RADFORD, Alec ; SUTSKEVER, Ilya ; AMODEI, Dario: *Language Models are Few-Shot Learners*. 2020
- [Cao 2021] CAO, Shuhao: Choose a transformer: Fourier or galerkin. In: *Advances in Neural Information Processing Systems* 34 (2021), S. 24924–24940
- [Caponnetto u. a. 2008] CAPONNETTO, Andrea ; MICCHELLI, Charles A. ; PONTIL, Massimiliano ; YING, Yiming: Universal multi-task kernels. In: *The Journal of Machine Learning Research* 9 (2008), S. 1615–1646

-
- [Chalvidal u. a. 2021] CHALVIDAL, Mathieu ; RICCI, Matthew ; VAN-RULLEN, Rufin ; SERRE, Thomas: Go with the flow: Adaptive control for Neural ODEs. In: *International Conference on Learning Representations*, 2021
- [Chang u. a. 2017] CHANG, Bo ; MENG, Lili ; HABER, Eldad ; RUTHOTTO, Lars ; BEGERT, David ; HOLTHAM, Elliot: *Reversible Architectures for Arbitrarily Deep Residual Neural Networks*. 2017
- [Chang u. a. 2018] CHANG, Bo ; MENG, Lili ; HABER, Eldad ; RUTHOTTO, Lars ; BEGERT, David ; HOLTHAM, Elliot: Reversible architectures for arbitrarily deep residual neural networks. In: *Proceedings of the AAAI conference on artificial intelligence* Bd. 32, 2018
- [Chen u. a. 2021] CHEN, Lili ; LU, Kevin ; RAJESWARAN, Aravind ; LEE, Kimin ; GROVER, Aditya ; LASKIN, Misha ; ABBEEL, Pieter ; SRINIVAS, Aravind ; MORDATCH, Igor: Decision transformer: Reinforcement learning via sequence modeling. In: *Advances in neural information processing systems* 34 (2021), S. 15084–15097
- [Chen u. a. 2018] CHEN, Ricky T. Q. ; RUBANOVA, Yulia ; BETTENCOURT, Jesse ; DUVENAUD, David K.: Neural Ordinary Differential Equations. In: *Advances in Neural Information Processing Systems*, 2018
- [Chen und Chen 1995] CHEN, Tianping ; CHEN, Hong: Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. In: *IEEE Transactions on Neural Networks* 6 (1995), Nr. 4, S. 911–917
- [Choromanski u. a. 2020a] CHOROMANSKI, Krzysztof ; DAVIS, Jared ; LIKHOSHERSTOV, Valerii ; SONG, Xingyou ; SLOTINE, Jean-Jacques E. ; VARLEY, Jacob ; LEE, Honglak ; WELLER, Adrian ; SINDHWANI, Vikas: An Ode to an ODE. In: *ArXiv abs/2006.11421* (2020)
- [Choromanski u. a. 2020b] CHOROMANSKI, Krzysztof ; DAVIS, Jared Q. ; LIKHOSHERSTOV, Valerii ; SONG, Xingyou ; SLOTINE, Jean-Jacques ; VARLEY, Jacob ; LEE, Honglak ; WELLER, Adrian ; SINDHWANI, Vikas: *An Ode to an ODE*. 2020
- [Clune 2019] CLUNE, Jeff: AI-GAs: AI-generating algorithms, an alternate paradigm for producing general artificial intelligence. In: *CoRR abs/1905.10985* (2019). – URL <http://arxiv.org/abs/1905.10985>

-
- [Clune 2020] CLUNE, Jeff: *AI-GAs: AI-generating algorithms, an alternate paradigm for producing general artificial intelligence*. 2020
- [Cobbe u. a. 2020] COBBE, Karl ; HESSE, Christopher ; HILTON, Jacob ; SCHULMAN, John: *Leveraging Procedural Generation to Benchmark Reinforcement Learning*. 2020
- [Costandi 2016] COSTANDI, Moheb: *Neuroplasticity*. MIT Press, 2016
- [Cover und Hart 1967] COVER, Thomas ; HART, Peter: Nearest neighbor pattern classification. In: *IEEE transactions on information theory* 13 (1967), Nr. 1, S. 21–27
- [Cunningham u. a. 2008] CUNNINGHAM, Pádraig ; CORD, Matthieu ; DELANY, Sarah J.: *Supervised Learning*. S. 21–49. In: CORD, Matthieu (Hrsg.) ; CUNNINGHAM, Pádraig (Hrsg.): *Machine Learning Techniques for Multimedia: Case Studies on Organization and Retrieval*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2008
- [Cybenko 1989] CYBENKO, George: Approximation by superpositions of a sigmoidal function. In: *Mathematics of control, signals and systems* 2 (1989), Nr. 4, S. 303–314
- [Daw u. a. 2005] DAW, Nathaniel D. ; NIV, Yael ; DAYAN, Peter: Uncertainty-based competition between prefrontal and dorsolateral striatal systems for behavioral control. In: *Nature neuroscience* 8 (2005), Nr. 12, S. 1704–1711
- [Deco u. a. 2011] DECO, Gustavo ; BUEHLMANN, Andres ; MASQUELIER, Timothée ; HUGUES, Etienne: The Role of Rhythmic Neural Synchronization in Rest and Task Conditions. In: *Frontiers in Human Neuroscience* 5 (2011). – URL <https://www.frontiersin.org/articles/10.3389/fnhum.2011.00004>. – ISSN 1662-5161
- [Degraeve u. a. 2022] DEGRAEVE, Jonas ; FELICI, Federico ; BUCHLI, Jonas ; NEUNERT, Michael ; TRACEY, Brendan ; CARPANESE, Francesco ; EWALDS, Timo ; HAFNER, Roland ; ABDOLMALEKI, Abbas ; LAS CASAS, Diego de u. a.: Magnetic control of tokamak plasmas through deep reinforcement learning. In: *Nature* 602 (2022), Nr. 7897, S. 414–419
- [Demircigil u. a. 2017] DEMIRCIGIL, Mete ; HEUSEL, Judith ; LÖWE, Matthias ; UPGANG, Sven ; VERMET, Franck: On a Model of Associative Memory with Huge Storage Capacity. In: *Journal of Statistical Physics* (2017), 02

- [Dinh u. a. 2016] DINH, Laurent ; SOHL-DICKSTEIN, Jascha ; BENGIO, Samy: Density estimation using real nvp. In: *arXiv preprint arXiv:1605.08803* (2016)
- [Doya 2002] DOYA, Kenji: Metalearning and neuromodulation. In: *Neural Networks* 15 (2002), Nr. 4, S. 495–506. – URL <https://www.sciencedirect.com/science/article/pii/S0893608002000448>. – ISSN 0893-6080
- [Duan u. a. 2016] DUAN, Yan ; SCHULMAN, John ; CHEN, Xi ; BARTLETT, Peter L. ; SUTSKEVER, Ilya ; ABBEEL, Pieter: RL2: Fast reinforcement learning via slow reinforcement learning. In: *arXiv preprint arXiv:1611.02779* (2016)
- [Dupont u. a. 2019] DUPONT, Emilien ; DOUCET, Arnaud ; TEH, Yee W.: Augmented Neural ODEs. In: *Advances in Neural Information Processing Systems*, 2019
- [Esteva u. a. 2019] ESTEVA, Andre ; ROBICQUET, Alexandre ; RAMSUNDAR, Bharath ; KULESHOV, Volodymyr ; DEPRISTO, Mark ; CHOU, Katherine ; CUI, Claire ; CORRADO, Greg ; THRUN, Sebastian ; DEAN, Jeff: A guide to deep learning in healthcare. In: *Nature medicine* 25 (2019), Nr. 1, S. 24–29
- [Evgeniou u. a. 2005] EVGENIOU, Theodoros ; MICCHELLI, Charles A. ; PONTIL, Massimiliano ; SHAWE-TAYLOR, John: Learning multiple tasks with kernel methods. In: *Journal of machine learning research* 6 (2005), Nr. 4
- [Fan u. a. 2019] FAN, Wenqi ; MA, Yao ; LI, Qing ; HE, Yuan ; ZHAO, Eric ; TANG, Jiliang ; YIN, Dawei: Graph neural networks for social recommendation. In: *The world wide web conference, 2019*, S. 417–426
- [Farebrother 2001] FAREBROTHER, R. W.: *Adrien-Marie Legendre*. S. 101–104. In: HEYDE, C. C. (Hrsg.) ; SENETA, E. (Hrsg.) ; CRÉPEL, P. (Hrsg.) ; FIENBERG, S. E. (Hrsg.) ; GANI, J. (Hrsg.): *Statisticians of the Centuries*. New York, NY : Springer New York, 2001. – ISBN 978-1-4613-0179-0
- [Ferraty und Vieu 2006] FERRATY, Frédéric ; VIEU, Philippe: *Non-parametric functional data analysis: theory and practice*. Bd. 76. Springer, 2006
- [Finlay u. a. 2020] FINLAY, Chris ; JACOBSEN, Jörn-Henrik ; NURBEKYAN, Levon ; OBERMAN, Adam M.: *How to train your neural ODE: the world of Jacobian and kinetic regularization*. 2020

- [Finn u. a. 2017a] FINN, Chelsea ; ABBEEL, Pieter ; LEVINE, Sergey: Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In: PRECUP, Doina (Hrsg.) ; TEH, Yee W. (Hrsg.): *Proceedings of the 34th International Conference on Machine Learning* Bd. 70, PMLR, 06–11 Aug 2017, S. 1126–1135
- [Finn und Levine 2017] FINN, Chelsea ; LEVINE, Sergey: Meta-learning and universality: Deep representations and gradient descent can approximate any learning algorithm. In: *arXiv preprint arXiv:1710.11622* (2017)
- [Finn u. a. 2017b] FINN, Chelsea ; YU, Tianhe ; ZHANG, Tianhao ; ABBEEL, Pieter ; LEVINE, Sergey: One-Shot Visual Imitation Learning via Meta-Learning. In: LEVINE, Sergey (Hrsg.) ; VANHOUCHE, Vincent (Hrsg.) ; GOLDBERG, Ken (Hrsg.): *Proceedings of the 1st Annual Conference on Robot Learning* Bd. 78, PMLR, 13–15 Nov 2017, S. 357–368
- [Fix und Hodges 1989] FIX, Evelyn ; HODGES, J. L.: Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties. In: *International Statistical Review / Revue Internationale de Statistique* 57 (1989), Nr. 3, S. 238–247
- [Friston 2005] FRISTON, Karl: A theory of cortical responses. In: *Philosophical transactions of the Royal Society B: Biological sciences* 360 (2005), Nr. 1456, S. 815–836
- [Friston 2010] FRISTON, Karl: The free-energy principle: a unified brain theory? In: *Nature reviews neuroscience* 11 (2010), Nr. 2, S. 127–138
- [Friston und Stephan 2007] FRISTON, Karl J. ; STEPHAN, Klaas E.: Free-energy and the brain. In: *Synthese* 159 (2007), S. 417–458
- [Fukushima 1980] FUKUSHIMA, Kunihiro: Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. In: *Biological cybernetics* 36 (1980), Nr. 4, S. 193–202
- [Gammerman u. a. 1998] GAMMERMAN, A ; VOVK, V ; VAPNIK, V: Learning by transduction. In: *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, 1998, S. 148–155
- [Garg u. a. 2022] GARG, Shivam ; TSIPRAS, Dimitris ; LIANG, Percy ; VALIANT, Gregory: *What Can Transformers Learn In-Context? A Case Study of Simple Function Classes*. 2022

- [Garland 2011] GARLAND, Michael: *NVIDIA GPU*. S. 1339–1345. In: PADUA, David (Hrsg.): *Encyclopedia of Parallel Computing*. Boston, MA : Springer US, 2011. – URL https://doi.org/10.1007/978-0-387-09766-4_276. – ISBN 978-0-387-09766-4
- [Garnelo u. a. 2018] GARNELO, Marta ; ROSENBAUM, Dan ; MADDISON, Christopher ; RAMALHO, Tiago ; SAXTON, David ; SHANAHAN, Murray ; TEH, Yee W. ; REZENDE, Danilo ; ESLAMI, SM A.: Conditional neural processes. In: *International conference on machine learning* PMLR (Veranst.), 2018, S. 1704–1713
- [Geirhos u. a. 2020] GEIRHOS, Robert ; MEDING, Kristof ; WICHMANN, Felix A.: Beyond accuracy: quantifying trial-by-trial behaviour of CNNs and humans by measuring error consistency. In: LAROCHELLE, H. (Hrsg.) ; RANZATO, M. (Hrsg.) ; HADSELL, R. (Hrsg.) ; BALCAN, M.F. (Hrsg.) ; LIN, H. (Hrsg.): *Advances in Neural Information Processing Systems* Bd. 33, Curran Associates, Inc., 2020, S. 13890–13902. – URL <https://proceedings.neurips.cc/paper/2020/file/9f6992966d4c363ea0162a056cb45fe5-Paper.pdf>
- [Gershman und Niv 2010] GERSHMAN, Samuel J. ; NIV, Yael: Learning latent structure: carving nature at its joints. In: *Current Opinion in Neurobiology* 20 (2010), S. 251–256
- [Gong u. a. 2019] GONG, Xinyu ; CHANG, Shiyu ; JIANG, Yifan ; WANG, Zhangyang: *AutoGAN: Neural Architecture Search for Generative Adversarial Networks*. 2019
- [Goodfellow u. a. 2020] GOODFELLOW, Ian ; POUGET-ABADIE, Jean ; MIRZA, Mehdi ; XU, Bing ; WARDE-FARLEY, David ; OZAIR, Sherjil ; COURVILLE, Aaron ; BENGIO, Yoshua: Generative Adversarial Networks. In: *Commun. ACM* 63 (2020), oct, Nr. 11, S. 139–144. – ISSN 0001-0782
- [Goswami u. a. 2022] GOSWAMI, Somdatta ; BORA, Aniruddha ; YU, Yue ; KARNIADAKIS, George E.: Physics-informed neural operators. In: *arXiv preprint arXiv:2207.05748* (2022)
- [Grant u. a. 2018] GRANT, Erin ; FINN, Chelsea ; LEVINE, Sergey ; DARRELL, Trevor ; GRIFFITHS, Thomas: Recasting Gradient-Based Meta-Learning as Hierarchical Bayes. In: *International Conference on Learning Representations*, URL https://openreview.net/forum?id=BJ_UL-k0b, 2018

-
- [Grathwohl u. a. 2019] GRATHWOHL, Will ; CHEN, Ricky T. Q. ; BETTENCOURT, Jesse ; DUVENAUD, David: Scalable Reversible Generative Models with Free-form Continuous Dynamics. In: *International Conference on Learning Representations*, 2019
- [Grathwohl u. a. 2020] GRATHWOHL, Will ; WANG, Kuan-Chieh ; JACOBSEN, Joern-Henrik ; DUVENAUD, David ; NOROUZI, Mohammad ; SWERSKY, Kevin: Your classifier is secretly an energy based model and you should treat it like one. In: *International Conference on Learning Representations*, URL <https://openreview.net/forum?id=Hkxzx0NtDB>, 2020
- [Gregor u. a. 2015] GREGOR, Karol ; DANIHELKA, Ivo ; GRAVES, Alex ; REZENDE, Danilo ; WIERSTRA, Daan: DRAW: A Recurrent Neural Network For Image Generation. In: BACH, Francis (Hrsg.) ; BLEI, David (Hrsg.): *Proceedings of the 32nd International Conference on Machine Learning* Bd. 37. Lille, France : PMLR, 07–09 Jul 2015, S. 1462–1471
- [Güçlü und van Gerven 2015] GÜÇLÜ, Umut ; GERVEN, Marcel A. van: Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream. In: *Journal of Neuroscience* 35 (2015), Nr. 27, S. 10005–10014
- [Gupta u. a. 2018] GUPTA, Abhishek ; MENDONCA, Russell ; LIU, YuXuan ; ABBEEL, Pieter ; LEVINE, Sergey: Meta-Reinforcement Learning of Structured Exploration Strategies. In: *Advances in Neural Information Processing Systems* 31 (2018), S. 5302–5311
- [Gupta u. a. 2021] GUPTA, Gaurav ; XIAO, Xiongye ; BOGDAN, Paul: Multiwavelet-based operator learning for differential equations. In: *Advances in Neural Information Processing Systems* 34 (2021), S. 24048–24062
- [Guyon u. a. 2015] GUYON, Isabelle ; BENNETT, Kristin ; CAWLEY, Gavin ; ESCALANTE, Hugo J. ; ESCALERA, Sergio ; HO, Tin K. ; MACIÀ, Núria ; RAY, Bisakha ; SAEED, Mehreen ; STATNIKOV, Alexander ; VIEGAS, Evelyne: Design of the 2015 ChaLearn AutoML challenge. In: *2015 International Joint Conference on Neural Networks (IJCNN)*, 2015, S. 1–8
- [Ha u. a. 2017] HA, David ; DAI, Andrew ; LE, Quoc V.: HyperNetworks, 2017

- [Haber und Ruthotto 2017] HABER, Eldad ; RUTHOTTO, Lars: Stable architectures for deep neural networks. In: *Inverse Problems* 34 (2017), Dec, Nr. 1, S. 014004. – URL <http://dx.doi.org/10.1088/1361-6420/aa9a90>. – ISSN 1361-6420
- [Han u. a. 2018] HAN, Jiequn ; JENTZEN, Arnulf ; E, Weinan: Solving high-dimensional partial differential equations using deep learning. In: *Proceedings of the National Academy of Sciences* 115 (2018), Nr. 34, S. 8505–8510
- [Hariharan und Girshick 2017] HARIHARAN, Bharath ; GIRSHICK, Ross: Low-shot visual recognition by shrinking and hallucinating features. In: *Proceedings of the IEEE international conference on computer vision*, 2017, S. 3018–3027
- [Harlow 1949] HARLOW, Harry F.: The formation of learning sets. In: *Psychological review* 56 1 (1949), S. 51–65
- [Hassabis u. a. 2017] HASSABIS, Demis ; KUMARAN, Dharshan ; SUMMERFIELD, Christopher ; BOTVINICK, Matthew: Neuroscience-Inspired Artificial Intelligence. In: *Neuron* 95 (2017), Nr. 2, S. 245–258. – ISSN 0896-6273
- [Hastie u. a. 2009] HASTIE, Trevor ; TIBSHIRANI, Robert ; FRIEDMAN, Jerome: The elements of statistical learnin. In: *Cited on* 33 (2009)
- [He u. a. 2015] HE, Kaiming ; ZHANG, Xiangyu ; REN, Shaoqing ; SUN, Jian: Deep Residual Learning for Image Recognition, 2015
- [He u. a. 2016] HE, Kaiming ; ZHANG, Xiangyu ; REN, Shaoqing ; SUN, Jian: Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, S. 770–778
- [Hebb 1949] HEBB, DO: The organization of behavior; a neuropsychological theory. (1949)
- [Hendrycks und Gimpel 2016] HENDRYCKS, Dan ; GIMPEL, Kevin: Gaussian error linear units (gelus). In: *arXiv preprint arXiv:1606.08415* (2016)
- [Hersbach u. a. 2020] HERSBACH, Hans ; BELL, Bill ; BERRISFORD, Paul ; HIRAHARA, Shoji ; HORÁNYI, András ; MUÑOZ-SABATER, Joaquín ; NICOLAS, Julien ; PEUBEY, Carole ; RADU, Raluca ; SCHEPERS, Dinand ; SIMMONS, Adrian ; SOCI, Cornel ; ABDALLA, Saleh ;

- ABELLAN, Xavier ; BALSAMO, Gianpaolo ; BECHTOLD, Peter ; BIAVATI, Gionata ; BIDLOT, Jean ; BONAVITA, Massimo ; DE CHIARA, Giovanna ; DAHLGREN, Per ; DEE, Dick ; DIAMANTAKIS, Michail ; DRAGANI, Rossana ; FLEMMING, Johannes ; FORBES, Richard ; FUENTES, Manuel ; GEER, Alan ; HAIMBERGER, Leo ; HEALY, Sean ; HOGAN, Robin J. ; HÓLM, Elías ; JANISKOVÁ, Marta ; KEELEY, Sarah ; LALOYAUX, Patrick ; LOPEZ, Philippe ; LUPU, Cristina ; RADNOTI, Gabor ; ROSNAY, Patricia de ; ROZUM, Iryna ; VAMBORG, Freja ; VILLAUME, Sebastien ; THÉPAUT, Jean-Noël: The ERA5 global reanalysis. In: *Quarterly Journal of the Royal Meteorological Society* 146 (2020), Nr. 730, S. 1999–2049
- [Hinton 1984] HINTON, Geoffrey E.: Distributed representations. (1984)
- [Hjelm u. a. 2018] HJELM, R. D. ; FEDOROV, Alex ; LAVOIE-MARCHILDON, Samuel ; GREWAL, Karan ; BACHMAN, Phil ; TRISCHLER, Adam ; BENGIO, Yoshua: Learning deep representations by mutual information estimation and maximization. In: *arXiv preprint arXiv:1808.06670* (2018)
- [Hochreiter und Schmidhuber 1997a] HOCHREITER, Sepp ; SCHMIDHUBER, Jürgen: Long Short-Term Memory. In: *Neural Comput.* 9 (1997), nov, Nr. 8, S. 1735–1780. – ISSN 0899-7667
- [Hochreiter und Schmidhuber 1997b] HOCHREITER, Sepp ; SCHMIDHUBER, Jürgen: Long Short-Term Memory. In: *Neural Computation* (1997)
- [Hochreiter u. a. 2001] HOCHREITER, Sepp ; YOUNGER, A. S. ; CONWELL, Peter R.: Learning to learn using gradient descent. In: *International Conference on Artificial Neural Networks* Springer (Veranst.), 2001, S. 87–94
- [Hofer u. a. 2017] HOFER, Christoph ; KWITT, Roland ; NIETHAMMER, Marc ; UHL, Andreas: Deep Learning with Topological Signatures. In: GUYON, I. (Hrsg.) ; LUXBURG, U. V. (Hrsg.) ; BENGIO, S. (Hrsg.) ; WALLACH, H. (Hrsg.) ; FERGUS, R. (Hrsg.) ; VISHWANATHAN, S. (Hrsg.) ; GARNETT, R. (Hrsg.): *Advances in Neural Information Processing Systems* Bd. 30, Curran Associates, Inc., 2017. – URL <https://proceedings.neurips.cc/paper/2017/file/883e881bb4d22a7add958f2d6b052c9f-Paper.pdf>

- [Holl u. a.] HOLL, Philipp ; THUREY, Nils ; KOLTUN, Vladlen: Learning to Control PDEs with Differentiable Physics. In: *International Conference on Learning Representations*
- [Hopfield 1982a] HOPFIELD, J J.: Neural networks and physical systems with emergent collective computational abilities. In: *Proceedings of the National Academy of Sciences* 79 (1982), Nr. 8, S. 2554–2558. – ISSN 0027-8424
- [Hopfield 1982b] HOPFIELD, John J.: Neural networks and physical systems with emergent collective computational abilities. In: *Proceedings of the national academy of sciences* 79 (1982), Nr. 8, S. 2554–2558
- [Hornik u. a. 1989] HORNİK, Kurt ; STINCHCOMBE, Maxwell ; WHITE, Halbert: Multilayer feedforward networks are universal approximators. In: *Neural Networks* 2 (1989), Nr. 5, S. 359–366. – URL <https://www.sciencedirect.com/science/article/pii/0893608089900208>. – ISSN 0893-6080
- [Houthoof t u. a. 2018] HOUTHOOFT, Rein ; CHEN, Richard Y. ; ISOLA, Phillip ; STADIE, Bradly C. ; WOLSKI, Filip ; HO, Jonathan ; ABBEEL, Pieter: Evolved policy gradients. In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018, S. 5405–5414
- [Huang und Rao 2011] HUANG, Yanping ; RAO, Rajesh P.: Predictive coding. In: *Wiley Interdisciplinary Reviews: Cognitive Science* 2 (2011), Nr. 5, S. 580–593
- [Hyman u. a. 2012] HYMAN, James M. ; MA, Liya ; BALAGUER-BALLESTER, Emili ; DURSTEWITZ, Daniel ; SEAMANS, Jeremy K.: Contextual encoding by ensembles of medial prefrontal cortex neurons. In: *Proceedings of the National Academy of Sciences* 109 (2012), Nr. 13, S. 5086–5091
- [Hyvärinen 2005] HYVÄRINEN, Aapo: Estimation of Non-Normalized Statistical Models by Score Matching. In: *Journal of Machine Learning Research* 6 (2005), Nr. 24, S. 695–709. – URL <http://jmlr.org/papers/v6/hyvarinen05a.html>
- [Irie u. a. 2022a] IRIE, Kazuki ; FACCIO, Francesco ; SCHMIDHUBER, Jürgen: *Neural Differential Equations for Learning to Program Neural Nets Through Continuous Learning Rules*. 2022. – URL <https://arxiv.org/abs/2206.01649>

- [Irie u. a. 2022b] IRIE, Kazuki ; SCHLAG, Imanol ; CSORDÁS, Róbert ; SCHMIDHUBER, Jürgen: A Modern Self-Referential Weight Matrix That Learns to Modify Itself. In: *Proc. Int. Conf. on Machine Learning (ICML)*. Baltimore, MD, USA, Juli 2022
- [Ivakhnenko 1971] IVAKHNENKO, Alexey G.: Polynomial theory of complex systems. In: *IEEE transactions on Systems, Man, and Cybernetics* (1971), Nr. 4, S. 364–378
- [Jacot u. a. 2018] JACOT, Arthur ; GABRIEL, Franck ; HONGLER, Clement: Neural Tangent Kernel: Convergence and Generalization in Neural Networks. In: BENGIO, S. (Hrsg.) ; WALLACH, H. (Hrsg.) ; LAROCHELLE, H. (Hrsg.) ; GRAUMAN, K. (Hrsg.) ; CESABIANCHI, N. (Hrsg.) ; GARNETT, R. (Hrsg.): *Advances in Neural Information Processing Systems* Bd. 31, Curran Associates, Inc., 2018. – URL <https://proceedings.neurips.cc/paper/2018/file/5a4be1fa34e62bb8a6ec6b91d2462f5a-Paper.pdf>
- [Jaderberg u. a. 2019] JADERBERG, Max ; CZARNECKI, Wojciech M. ; DUNNING, Iain ; MARRIS, Luke ; LEVER, Guy ; CASTAÑEDA, Antonio G. ; BEATTIE, Charles ; RABINOWITZ, Neil C. ; MORCOS, Ari S. ; RUDERMAN, Avraham ; SONNERAT, Nicolas ; GREEN, Tim ; DEASON, Louise ; LEIBO, Joel Z. ; SILVER, David ; HASSABIS, Demis ; KAVUKCUOGLU, Koray ; GRAEPEL, Thore: Human-level performance in 3D multiplayer games with population-based reinforcement learning. In: *Science* 364 (2019), Nr. 6443, S. 859–865
- [Jaderberg u. a. 2015] JADERBERG, Max ; SIMONYAN, Karen ; ZISSERMAN, Andrew ; KAVUKCUOGLU, koray: Spatial Transformer Networks. In: *Advances in Neural Information Processing Systems*, 2015
- [Jaitly u. a. 2015] JAITLY, Navdeep ; SUSSILLO, David ; LE, Quoc V. ; VINYALS, Oriol ; SUTSKEVER, Ilya ; BENGIO, Samy: A neural transducer. In: *arXiv preprint arXiv:1511.04868* (2015)
- [Jia u. a. 2016] JIA, Xu ; DE BRABANDERE, Bert ; TUYTELAARS, Tinne ; GOOL, Luc V.: Dynamic Filter Networks. In: LEE, D. (Hrsg.) ; SUGIYAMA, M. (Hrsg.) ; LUXBURG, U. (Hrsg.) ; GUYON, I. (Hrsg.) ; GARNETT, R. (Hrsg.): *Advances in Neural Information Processing Systems*, 2016
- [Jin u. a. 2020] JIN, Pengzhan ; LU, Lu ; TANG, Yifa ; KARNIADAKIS, George E.: Quantifying the generalization error in deep learning in terms of data distribution and neural network smoothness. In: *Neural Networks* 130 (2020), S. 85–99

-
- [Jozefowicz u. a. 2016] JOZEFOWICZ, Rafal ; VINYALS, Oriol ; SCHUSTER, Mike ; SHAZEER, Noam ; WU, Yonghui: *Exploring the Limits of Language Modeling*. 2016
- [Jumper u. a. 2021] JUMPER, John ; EVANS, Richard ; PRITZEL, Alexander ; GREEN, Tim ; FIGURNOV, Michael ; RONNEBERGER, Olaf ; TUNYASUVUNAKOOL, Kathryn ; BATES, Russ ; ŽÍDEK, Augustin ; POTAPENKO, Anna u. a.: Highly accurate protein structure prediction with AlphaFold. In: *Nature* 596 (2021), Nr. 7873, S. 583–589
- [Kane 1994] KANE, G. S.: Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol 1: Foundations, vol 2: Psychological and Biological Models. In: *JAMA* 271 (1994), S. 956–957
- [Keskar u. a. 2016] KESKAR, Nitish S. ; MUDIGERE, Dheevatsa ; NOCEDAL, Jorge ; SMELYANSKIY, Mikhail ; TANG, Ping Tak P.: On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima. In: *CoRR* abs/1609.04836 (2016). – URL <http://arxiv.org/abs/1609.04836>
- [Khaligh-Razavi und Kriegeskorte 2014] KHALIGH-RAZAVI, Seyed-Mahdi ; KRIEGESKORTE, Nikolaus: Deep supervised, but not unsupervised, models may explain IT cortical representation. In: *PLoS computational biology* 10 (2014), Nr. 11, S. e1003915
- [Khamassi u. a. 2013] KHAMASSI, Mehdi ; ENEL, Pierre ; DOMINEY, Peter F. ; PROCYK, Emmanuel: Chapter 22 - Medial prefrontal cortex and the adaptive regulation of reinforcement learning parameters. In: PAMMI, V.S. C. (Hrsg.) ; SRINIVASAN, Narayanan (Hrsg.): *Decision Making* Bd. 202. Elsevier, 2013, S. 441–464. – ISSN 0079-6123
- [Kingma und Ba 2014] KINGMA, Diederik P. ; BA, Jimmy: *Adam: A Method for Stochastic Optimization*. 2014
- [Kingma und Welling 2013] KINGMA, Diederik P. ; WELLING, Max: Auto-encoding variational bayes. In: *arXiv preprint arXiv:1312.6114* (2013)
- [Kissas u. a. 2022] KISSAS, Georgios ; SEIDMAN, Jacob H. ; GUILHOTO, Leonardo F. ; PRECIADO, Victor M. ; PAPPAS, George J. ; PERDIKARIS, Paris: Learning operators with coupled attention. In: *Journal of Machine Learning Research* 23 (2022), Nr. 215, S. 1–63

- [Koiran 1994] KOIRAN, Pascal: Dynamics of Discrete Time, Continuous State Hopfield Networks. In: *Neural Computation* 6 (1994), Nr. 3, S. 459–468
- [Kotsiopoulos u. a. 2021] KOTSIPOULOS, Thanasis ; SARIGIANNIDIS, Panagiotis ; IOANNIDIS, Dimosthenis ; TZOVARAS, Dimitrios: Machine learning and deep learning in smart manufacturing: The smart grid paradigm. In: *Computer Science Review* 40 (2021), S. 100341
- [Krizhevsky u. a. 2012] KRIZHEVSKY, Alex ; SUTSKEVER, Ilya ; HINTON, Geoffrey E.: ImageNet Classification with Deep Convolutional Neural Networks. In: PEREIRA, F. (Hrsg.) ; BURGESS, C.J. (Hrsg.) ; BOTTOU, L. (Hrsg.) ; WEINBERGER, K.Q. (Hrsg.): *Advances in Neural Information Processing Systems* Bd. 25, Curran Associates, Inc., 2012. – URL <https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf>
- [Krizhevsky u. a. 2017] KRIZHEVSKY, Alex ; SUTSKEVER, Ilya ; HINTON, Geoffrey E.: ImageNet Classification with Deep Convolutional Neural Networks. In: *Commun. ACM* 60 (2017), may, Nr. 6, S. 84–90. – ISSN 0001-0782
- [Krotov und Hopfield 2016] KROTOV, Dmitry ; HOPFIELD, John J.: Dense Associative Memory for Pattern Recognition. In: *NIPS*, 2016
- [Kubilius u. a. 2019] KUBILIUS, Jonas ; SCHRIMPF, Martin ; KAR, Kohitij ; RAJALINGHAM, Rishi ; HONG, Ha ; MAJAJ, Najib ; ISSA, Elias ; BASHIVAN, Pouya ; PRESCOTT-ROY, Jonathan ; SCHMIDT, Kailyn ; NAYEBI, Aran ; BEAR, Daniel ; YAMINS, Daniel L. ; DICARLO, James J.: Brain-Like Object Recognition with High-Performing Shallow Recurrent ANNs. In: WALLACH, H. (Hrsg.) ; LAROCHELLE, H. (Hrsg.) ; BEYGELZIMER, A. (Hrsg.) ; ALCHÉ-BUC, F. d'(Hrsg.) ; FOX, E. (Hrsg.) ; GARNETT, R. (Hrsg.): *Advances in Neural Information Processing Systems* Bd. 32, Curran Associates, Inc., 2019. – URL <https://proceedings.neurips.cc/paper/2019/file/7813d1590d28a7dd372ad54b5d29d033-Paper.pdf>
- [Kuramoto 1975] KURAMOTO, Yoshiki: Self-entrainment of a population of coupled non-linear oscillators. In: *International Symposium on Mathematical Problems in Theoretical Physics: January 23–29, 1975, Kyoto University, Kyoto/Japan* Springer (Veranst.), 1975, S. 420–422
- [Köhler u. a. 2019] KÖHLER, Jonas ; KLEIN, Leon ; NOÉ, Frank: *Equivariant Flows: sampling configurations for multi-body systems with symmetric energies*. 2019

- [Lake und Baroni 2018] LAKE, Brenden ; BARONI, Marco: *Still not systematic after all these years: On the compositional skills of sequence-to-sequence recurrent networks*. 2018
- [Lake u. a. 2015] LAKE, Brenden M. ; SALAKHUTDINOV, Ruslan ; TENENBAUM, Joshua B.: Human-level concept learning through probabilistic program induction. In: *Science* 350 (2015), Nr. 6266, S. 1332–1338
- [Lake u. a. 2017] LAKE, Brenden M. ; ULLMAN, Tomer D. ; TENENBAUM, Joshua B. ; GERSHMAN, Samuel J.: Building machines that learn and think like people. In: *Behavioral and Brain Sciences* 40 (2017), S. e253
- [Larochelle und Hinton 2010] LAROCHELLE, Hugo ; HINTON, Geoffrey E.: Learning to combine foveal glimpses with a third-order Boltzmann machine. In: LAFFERTY, J. (Hrsg.) ; WILLIAMS, C. (Hrsg.) ; SHAWE-TAYLOR, J. (Hrsg.) ; ZEMEL, R. (Hrsg.) ; CULOTTA, A. (Hrsg.): *Advances in Neural Information Processing Systems* Bd. 23, Curran Associates, Inc., 2010. – URL <https://proceedings.neurips.cc/paper/2010/file/677e09724f0e2df9b6c000b75b5da10d-Paper.pdf>
- [Laskin u. a.] LASKIN, Michael ; WANG, Luyu ; OH, Junhyuk ; PARISOTTO, Emilio ; SPENCER, Stephen ; STEIGERWALD, Richie ; STROUSE, DJ ; HANSEN, Steven S. ; FILOS, Angelos ; BROOKS, Ethan u. a.: In-context Reinforcement Learning with Algorithm Distillation. In: *Deep Reinforcement Learning Workshop NeurIPS 2022*
- [Lecun u. a. 1998] LECUN, Y. ; BOTTOU, L. ; BENGIO, Y. ; HAFFNER, P.: Gradient-based learning applied to document recognition. In: *Proceedings of the IEEE* 86 (1998), Nr. 11, S. 2278–2324
- [LeCun u. a.] LECUN, Yann ; BENGIO, Yoshua u. a.: Convolutional networks for images, speech, and time series.
- [LeCun u. a. 2015] LECUN, Yann ; BENGIO, Yoshua ; HINTON, Geoffrey: Deep learning. In: *nature* 521 (2015), Nr. 7553, S. 436–444
- [LeCun u. a. 2006] LECUN, Yann ; CHOPRA, Sumit ; HADSELL, Raia ; RANZATO, M ; HUANG, Fufei: A tutorial on energy-based learning. In: *Predicting structured data* 1 (2006), Nr. 0
- [Lee u. a. 2014] LEE, Sang Wan ; SHIMOJO, Shinsuke ; O'DOHERTY, John P.: Neural Computations Underlying Arbitration between

-
- Model-Based and Model-free Learning. In: *Neuron* 81 (2014), Nr. 3, S. 687–699. – ISSN 0896-6273
- [Lengyel und Dayan 2008] LENGYEL, Máté ; DAYAN, Peter: Hippocampal Contributions to Control: The Third Way. In: PLATT, J. (Hrsg.) ; KOLLER, D. (Hrsg.) ; SINGER, Y. (Hrsg.) ; ROWEIS, S. (Hrsg.): *Advances in Neural Information Processing Systems* Bd. 20, Curran Associates, Inc., 2008
- [Levine u. a. 2016] LEVINE, Sergey ; FINN, Chelsea ; DARRELL, Trevor ; ABBEEL, Pieter: End-to-end training of deep visuomotor policies. In: *The Journal of Machine Learning Research* 17 (2016), Nr. 1, S. 1334–1373
- [Li u. a. 2017] LI, Qianxiao ; CHEN, Long ; TAI, Cheng ; WEINAN, E: Maximum principle based algorithms for deep learning. In: *The Journal of Machine Learning Research* 18 (2017), Nr. 1, S. 5998–6026
- [Li u. a. 2019] LI, Qianxiao ; TAI, Cheng ; WEINAN, E: Stochastic modified equations and dynamics of stochastic gradient algorithms i: Mathematical foundations. In: *The Journal of Machine Learning Research* 20 (2019), Nr. 1, S. 1474–1520
- [Li u. a. 2020a] LI, Zongyi ; KOVACHKI, Nikola ; AZIZZADENESHELI, Kamyar ; LIU, Burigede ; BHATTACHARYA, Kaushik ; STUART, Andrew ; ANANDKUMAR, Anima: Fourier neural operator for parametric partial differential equations. In: *arXiv preprint arXiv:2010.08895* (2020)
- [Li u. a. 2020b] LI, Zongyi ; KOVACHKI, Nikola ; AZIZZADENESHELI, Kamyar ; LIU, Burigede ; STUART, Andrew ; BHATTACHARYA, Kaushik ; ANANDKUMAR, Anima: Multipole Graph Neural Operator for Parametric Partial Differential Equations. In: LAROCHELLE, H. (Hrsg.) ; RANZATO, M. (Hrsg.) ; HADSELL, R. (Hrsg.) ; BALCAN, M.F. (Hrsg.) ; LIN, H. (Hrsg.): *Advances in Neural Information Processing Systems* Bd. 33, Curran Associates, Inc., 2020, S. 6755–6766
- [Liberzon 2011] LIBERZON, Daniel: *Calculus of variations and optimal control theory: a concise introduction*. Princeton university press, 2011
- [Lillicrap u. a. 2016] LILLICRAP, Timothy P. ; HUNT, Jonathan J. ; PRITZEL, Alexander ; HEES, Nicolas ; EREZ, Tom ; TASSA, Yuval ; SILVER, David ; WIERSTRA, Daan: Continuous control with deep reinforcement learning. In: *ICLR (Poster)*, 2016

- [Lin u. a. 2022] LIN, Rong R. ; ZHANG, Hai Z. ; ZHANG, Jun: On reproducing kernel Banach spaces: Generic definitions and unified framework of constructions. In: *Acta Mathematica Sinica, English Series* 38 (2022), Nr. 8, S. 1459–1483
- [Lin u. a. 2018] LIN, Zichuan ; ZHAO, Tianqi ; YANG, Guangwen ; ZHANG, Lintao: Episodic Memory Deep Q-Networks. In: *IJCAI*, 2018
- [Linnainmaa 1976] LINNAINMAA, Seppo: Taylor expansion of the accumulated rounding error. In: *BIT Numerical Mathematics* 16 (1976), S. 146–160
- [Liu u. a. 2018] LIU, Chenxi ; ZOPH, Barret ; NEUMANN, Maxim ; SHLENS, Jonathon ; HUA, Wei ; LI, Li-Jia ; FEI-FEI, Li ; YUILLE, Alan ; HUANG, Jonathan ; MURPHY, Kevin: Progressive neural architecture search. In: *Proceedings of the European conference on computer vision (ECCV)*, 2018, S. 19–34
- [Liu und Theodorou 2019] LIU, Guan-Horng ; THEODOROU, Evangelos A.: Deep learning theory review: An optimal control and dynamical systems perspective. In: *arXiv preprint arXiv:1908.10920* (2019)
- [Liu u. a. 2019] LIU, Hanxiao ; SIMONYAN, Karen ; YANG, Yiming: DARTS: Differentiable Architecture Search, 2019
- [Lu u. a. 2019] LU, Lu ; JIN, Pengzhan ; KARNIADAKIS, George E.: DeepONet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. In: *arXiv preprint arXiv:1910.03193* (2019)
- [Lu u. a. 2021] LU, Lu ; JIN, Pengzhan ; PANG, Guofei ; ZHANG, Zhongqiang ; KARNIADAKIS, George E.: Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. In: *Nature Machine Intelligence* 3 (2021), Nr. 3, S. 218–229
- [Lu u. a. 2018] LU, Yiping ; ZHONG, Aoxiao ; LI, Quanzheng ; DONG, Bin: Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations. In: *International Conference on Machine Learning* PMLR (Veranst.), 2018, S. 3276–3285
- [Lu u. a. 2017] LU, Zhou ; PU, Hongming ; WANG, Feicheng ; HU, Zhiqiang ; WANG, Liwei: The Expressive Power of Neural Networks: A View from the Width. (2017), 09
- [Lyu und Simoncelli 2008] LYU, Siwei ; SIMONCELLI, Eero P.: Nonlinear image representation using divisive normalization. In: *2008 IEEE*

-
- Conference on Computer Vision and Pattern Recognition* IEEE (Veranst.), 2008, S. 1–8
- [Marr 2010] MARR, David: *Vision: A computational investigation into the human representation and processing of visual information*. MIT press, 2010
- [Massaroli u. a. 2020a] MASSAROLI, Stefano ; POLI, Michael ; BIN, Michelangelo ; PARK, Jinkyoo ; YAMASHITA, Atsushi ; ASAMA, Hajime: Stable neural flows. In: *arXiv preprint arXiv:2003.08063* (2020)
- [Massaroli u. a. 2020b] MASSAROLI, Stefano ; POLI, Michael ; PARK, Jinkyoo ; YAMASHITA, Atsushi ; ASAMA, Hajime: *Dissecting Neural ODEs*. 2020
- [Mathis und Mathis 2020] MATHIS, Mackenzie W. ; MATHIS, Alexander: Deep learning tools for the measurement of animal behavior in neuroscience. In: *Current Opinion in Neurobiology* 60 (2020), S. 1–11. – URL <https://www.sciencedirect.com/science/article/pii/S0959438819301151>. – Neurobiology of Behavior. – ISSN 0959-4388
- [McCloskey und Cohen 1989] MCCLOSKEY, Michael ; COHEN, Neal J.: Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. Academic Press, 1989, S. 109–165
- [McCulloch und Pitts 1943] MCCULLOCH, Warren S. ; PITTS, Walter: A logical calculus of the ideas immanent in nervous activity. In: *The bulletin of mathematical biophysics* 5 (1943), S. 115–133
- [Miconi 2016] MICONI, Thomas: Learning to learn with backpropagation of Hebbian plasticity. In: *arXiv: Neural and Evolutionary Computing* (2016)
- [Miconi 2021] MICONI, Thomas: *Learning to acquire novel cognitive tasks with evolution, plasticity and meta-meta-learning*. 2021. – URL <https://arxiv.org/abs/2112.08588>
- [Miconi u. a. 2018a] MICONI, Thomas ; RAWAL, Aditya ; CLUNE, Jeff ; STANLEY, Kenneth O.: Backpropamine: training self-modifying neural networks with differentiable neuromodulated plasticity. In: *International Conference on Learning Representations*, 2018
- [Miconi u. a. 2018b] MICONI, Thomas ; STANLEY, Kenneth ; CLUNE, Jeff: Differentiable plasticity: training plastic neural networks with backpropagation. In: *Proceedings of the 35th International Conference on Machine Learning*, 2018

- [Mikolov u. a. 2013a] MIKOLOV, Tomas ; CHEN, Kai ; CORRADO, Greg ; DEAN, Jeffrey: Efficient estimation of word representations in vector space. In: *arXiv preprint arXiv:1301.3781* (2013)
- [Mikolov u. a. 2013b] MIKOLOV, Tomas ; SUTSKEVER, Ilya ; CHEN, Kai ; CORRADO, Greg S. ; DEAN, Jeff: Distributed Representations of Words and Phrases and their Compositionality. In: BURGESS, C.J. (Hrsg.) ; BOTTOU, L. (Hrsg.) ; WELLING, M. (Hrsg.) ; GHAHRAMANI, Z. (Hrsg.) ; WEINBERGER, K.Q. (Hrsg.): *Advances in Neural Information Processing Systems* Bd. 26, Curran Associates, Inc., 2013. – URL <https://proceedings.neurips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf>
- [Mildenhall u. a. 2020] MILDENHALL, Ben ; SRINIVASAN, Pratul P. ; TANCIK, Matthew ; BARRON, Jonathan T. ; RAMAMOORTHI, Ravi ; NG, Ren: NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16* Springer (Veranst.), 2020, S. 405–421
- [Mishra u. a. 2018] MISHRA, Nikhil ; ROHANINEJAD, Mostafa ; CHEN, Xi ; ABBEEL, Pieter: A Simple Neural Attentive Meta-Learner. In: *International Conference on Learning Representations*, 2018
- [Misra und Maaten 2020] MISRA, Ishan ; MAATEN, Laurens van d.: Self-Supervised Learning of Pretext-Invariant Representations. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020
- [Mnih u. a. 2016] MNIH, Volodymyr ; BADIA, Adria P. ; MIRZA, Mehdi ; GRAVES, Alex ; LILICRAP, Timothy ; HARLEY, Tim ; SILVER, David ; KAVUKCUOGLU, Koray: Asynchronous methods for deep reinforcement learning. In: *International conference on machine learning* PMLR (Veranst.), 2016, S. 1928–1937
- [Mnih u. a. 2014] MNIH, Volodymyr ; HEESS, Nicolas ; GRAVES, Alex u. a.: Recurrent models of visual attention. In: *Advances in neural information processing systems 27* (2014)
- [Mnih u. a. 2013] MNIH, Volodymyr ; KAVUKCUOGLU, Koray ; SILVER, David ; GRAVES, Alex ; ANTONOGLU, Ioannis ; WIERSTRA, Daan ; RIEDMILLER, Martin A.: Playing Atari with Deep Reinforcement Learning. In: *ArXiv abs/1312.5602* (2013)

-
- [Mnih u. a. 2015] MNIH, Volodymyr ; KAVUKCUOGLU, Koray ; SILVER, David ; RUSU, Andrei A. ; VENESS, Joel ; BELLEMARE, Marc G. ; GRAVES, Alex ; RIEDMILLER, Martin ; FIDJELAND, Andreas K. ; OSTROVSKI, Georg u. a.: Human-level control through deep reinforcement learning. In: *nature* 518 (2015), Nr. 7540, S. 529–533
- [Moore 2006] MOORE, Gordon E.: Progress in digital integrated electronics [Technical literature, Copyright 1975 IEEE. Reprinted, with permission. Technical Digest. International Electron Devices Meeting, IEEE, 1975, pp. 11-13.]. In: *IEEE Solid-State Circuits Society Newsletter* 11 (2006), Nr. 3, S. 36–37
- [Munkhdalai und Yu 2017] MUNKHDALAI, Tsendsuren ; YU, Hong: Meta networks. In: *International Conference on Machine Learning* PMLR (Veranst.), 2017, S. 2554–2563
- [Nadaraya 1964] NADARAYA, Elizbar A.: On estimating regression. In: *Theory of Probability & Its Applications* 9 (1964), Nr. 1, S. 141–142
- [Najarro und Risi 2020] NAJARRO, Elias ; RISI, Sebastian: Meta-Learning through Hebbian Plasticity in Random Networks. In: LAROCHELLE, H. (Hrsg.) ; RANZATO, M. (Hrsg.) ; HADSELL, R. (Hrsg.) ; BALCAN, M. F. (Hrsg.) ; LIN, H. (Hrsg.): *Advances in Neural Information Processing Systems* Bd. 33, Curran Associates, Inc., 2020, S. 20719–20731
- [Nakahara und Hikosaka 2012] NAKAHARA, Hiroyuki ; HIKOSAKA, Okihide: Learning to represent reward structure: A key to adapting to complex environments. In: *Neuroscience research* 74 (2012), Nr. 3-4, S. 177–183
- [Nelsen und Stuart 2021] NELSEN, Nicholas H. ; STUART, Andrew M.: The Random Feature Model for Input-Output Maps between Banach Spaces. In: *SIAM Journal on Scientific Computing* 43 (2021), Nr. 5, S. A3212–A3243
- [Nguyen u. a. 2018] NGUYEN, Hoang ; KIEU, Le-Minh ; WEN, Tao ; CAI, Chen: Deep learning methods in transportation domain: a review. In: *IET Intelligent Transport Systems* 12 (2018), Nr. 9, S. 998–1004
- [Norcliffe u. a. 2020] NORCLIFFE, Alexander ; BODNAR, Cristian ; DAY, Ben ; SIMIDJIEVSKI, Nikola ; LIÒ, Pietro: *On Second Order Behaviour in Augmented Neural ODEs*. 2020

- [Olsson u. a. 2022] OLSSON, Catherine ; ELHAGE, Nelson ; NANDA, Neel ; JOSEPH, Nicholas ; DASARMA, Nova ; HENIGHAN, Tom ; MANN, Ben ; ASKELL, Amanda ; BAI, Yuntao ; CHEN, Anna u. a.: In-context learning and induction heads. In: *arXiv preprint arXiv:2209.11895* (2022)
- [Oord u. a. 2018] OORD, Aaron van d. ; LI, Yazhe ; VINYALS, Oriol: Representation learning with contrastive predictive coding. In: *arXiv preprint arXiv:1807.03748* (2018)
- [Ortega u. a. 2019] ORTEGA, Pedro A. ; WANG, Jane X. ; ROWLAND, Mark ; GENEWEIN, Tim ; KURTH-NELSON, Zeb ; PASCANU, Razvan ; HEES, Nicolas ; VENESS, Joel ; PRITZEL, Alex ; SPRECHMANN, Pablo u. a.: Meta-learning of sequential strategies. In: *arXiv preprint arXiv:1905.03030* (2019)
- [Ott u. a. 2021] OTT, Katharina ; KATIYAR, Prateek ; HENNIG, Philipp ; TIEMANN, Michael: ResNet After All: Neural {ODE}s and Their Numerical Solution. In: *International Conference on Learning Representations*, URL <https://openreview.net/forum?id=HxzSxSxL0JZ>, 2021
- [Panzeri u. a. 2010] PANZERI, Stefano ; MONTANI, Fernando ; NOTARO, Giuseppe ; MAGRI, Cesare ; PETERSON, Rasmus S.: *Population Coding*. In: GRÜN, Sonja (Hrsg.) ; ROTTER, Stefan (Hrsg.): *Analysis of Parallel Spike Trains*. Boston, MA : Springer US, 2010
- [Park und Reuter-Lorenz 2009] PARK, Denise C. ; REUTER-LORENZ, Patricia: The adaptive brain: aging and neurocognitive scaffolding. In: *Annual review of psychology* 60 (2009), S. 173–196
- [Paszke u. a. 2019] PASZKE, Adam ; GROSS, Sam ; MASSA, Francisco ; LERER, Adam ; BRADBURY, James ; CHANAN, Gregory ; KILLEEN, Trevor ; LIN, Zeming ; GIMELSHEIN, Natalia ; ANTIGA, Luca ; DESMAISON, Alban ; KÖPF, Andreas ; YANG, Edward ; DEVITO, Zach ; RAISON, Martin ; TEJANI, Alykhan ; CHILAMKURTHY, Sasank ; STEINER, Benoit ; FANG, Lu ; BAI, Junjie ; CHINTALA, Soumith: *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. 2019
- [Pathak u. a. 2022] PATHAK, Jaideep ; SUBRAMANIAN, Shashank ; HARRINGTON, Peter ; RAJA, Sanjeev ; CHATTOPADHYAY, Ashesh ; MARDANI, Morteza ; KURTH, Thorsten ; HALL, David ; LI, Zongyi ; AZIZZADENESHELI, Kamyar ; HASSANZADEH, Pedram ; KASHINATH,

- Karthik ; ANANDKUMAR, Animashree: *FourCastNet: A Global Data-driven High-resolution Weather Model using Adaptive Fourier Neural Operators*. 2022
- [Pedregosa u. a. 2011] PEDREGOSA, F. ; VAROQUAUX, G. ; GRAMFORT, A. ; MICHEL, V. ; THIRION, B. ; GRISEL, O. ; BLONDEL, M. ; PRETTENHOFER, P. ; WEISS, R. ; DUBOURG, V. ; VANDERPLAS, J. ; PASSOS, A. ; COURNAPEAU, D. ; BRUCHER, M. ; PERROT, M. ; DUCHESNAY, E.: Scikit-learn: Machine Learning in Python. In: *Journal of Machine Learning Research* 12 (2011), S. 2825–2830
- [Pesme u. a. 2021] PESME, Scott ; PILLAUD-VIVIEN, Loucas ; FLAMMARION, Nicolas: Implicit Bias of SGD for Diagonal Linear Networks: a Provable Benefit of Stochasticity. In: RANZATO, M. (Hrsg.) ; BEYGELZIMER, A. (Hrsg.) ; DAUPHIN, Y. (Hrsg.) ; LIANG, P.S. (Hrsg.) ; VAUGHAN, J. W. (Hrsg.): *Advances in Neural Information Processing Systems* Bd. 34, Curran Associates, Inc., 2021, S. 29218–29230. – URL <https://proceedings.neurips.cc/paper/2021/file/f4661398cb1a3abd3ffe58600bf11322-Paper.pdf>
- [Pickering u. a. 2022] PICKERING, Ethan ; GUTH, Stephen ; KARNADAKIS, George E. ; SAPSIS, Themistoklis P.: Discovering and forecasting extreme events via active learning in neural operators. In: *Nature Computational Science* 2 (2022), Nr. 12, S. 823–833
- [Poggio u. a. 2020] POGGIO, Tomaso ; LIAO, Qianli ; BANBURSKI, Andrzej: Complexity control by gradient descent in deep networks. In: *Nature communications* 11 (2020), Nr. 1, S. 1027
- [Pontes-Filho u. a. 2022] PONTES-FILHO, Sidney ; WALKER, Kathryn ; NAJARRO, Elias ; NICHELE, Stefano ; RISI, Sebastian: A Single Neural Cellular Automaton for Body-Brain Co-Evolution. In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion, Association for Computing Machinery, 2022 (GECCO '22)*, S. 148–151. – ISBN 9781450392686
- [Pontryagin Lev Semyonovich ; Boltyanskii V G und F 1962] PONTRYAGIN LEV SEMYONOVICH ; BOLTYANSKII V G, Gamkrelidze R V. ; F, Mishchenko E.: *The mathematical theory of optimal processes*. 1962
- [Prigogine und Nicolis 1985] PRIGOGINE, I. ; NICOLIS, G.: *Self-Organisation in Nonequilibrium Systems: Towards A Dynamics of Complexity*. S. 3–12. In: HAZEWINKEL, M. (Hrsg.) ; JURKOVICH,

- R. (Hrsg.) ; PAELINCK, J. H. P. (Hrsg.): *Bifurcation Analysis: Principles, Applications and Synthesis*. Dordrecht : Springer Netherlands, 1985
- [Prim 1957] PRIM, Robert C.: Shortest connection networks and some generalizations. In: *The Bell System Technical Journal* 36 (1957), Nr. 6, S. 1389–1401
- [Pritzel u. a. 2017a] PRITZEL, Alexander ; URIA, Benigno ; SRINIVASAN, Sriram ; BADIA, Adria P. ; VINYALS, Oriol ; HASSABIS, Demis ; WIERSTRA, Daan ; BLUNDELL, Charles: Neural episodic control. In: *International Conference on Machine Learning* PMLR (Veranst.), 2017, S. 2827–2836
- [Pritzel u. a. 2017b] PRITZEL, Alexander ; URIA, Benigno ; SRINIVASAN, Sriram ; BADIA, Adrià P. ; VINYALS, Oriol ; HASSABIS, Demis ; WIERSTRA, Daan ; BLUNDELL, Charles: Neural Episodic Control. In: PRECUP, Doina (Hrsg.) ; TEH, Yee W. (Hrsg.): *Proceedings of the 34th International Conference on Machine Learning* Bd. 70, PMLR, 06–11 Aug 2017, S. 2827–2836
- [Pritzel u. a. 2017c] PRITZEL, Alexander ; URIA, Benigno ; SRINIVASAN, Sriram ; BADIA, Adrià P. ; VINYALS, Oriol ; HASSABIS, Demis ; WIERSTRA, Daan ; BLUNDELL, Charles: Neural Episodic Control. In: PRECUP, Doina (Hrsg.) ; TEH, Yee W. (Hrsg.): *Proceedings of the 34th International Conference on Machine Learning* Bd. 70, PMLR, 06–11 Aug 2017, S. 2827–2836
- [Qiu u. a. 2018] QIU, Jiezhong ; TANG, Jian ; MA, Hao ; DONG, Yuxiao ; WANG, Kuansan ; TANG, Jie: Deepinf: Social influence prediction with deep learning. In: *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, S. 2110–2119
- [Quinlan 1986] QUINLAN, J. R.: Induction of decision trees. In: *Machine learning* 1 (1986), Nr. 1, S. 81–106
- [Raissi u. a. 2017] RAISSI, Maziar ; PERDIKARIS, Paris ; KARNADAKIS, George E.: Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. In: *arXiv preprint arXiv:1711.10561* (2017)
- [Rakelly u. a. 2019] RAKELLY, Kate ; ZHOU, Aurick ; FINN, Chelsea ; LEVINE, Sergey ; QUILLEN, Deirdre: Efficient off-policy meta-reinforcement learning via probabilistic context variables. In: *In-*

- ternational conference on machine learning* PMLR (Veranst.), 2019, S. 5331–5340
- [Ramesh u. a. 2021] RAMESH, Aditya ; PAVLOV, Mikhail ; GOH, Gabriel ; GRAY, Scott ; VOSS, Chelsea ; RADFORD, Alec ; CHEN, Mark ; SUTSKEVER, Ilya: *Zero-Shot Text-to-Image Generation*. 2021
- [Ramsauer u. a. 2020] RAMSAUER, Hubert ; SCHÄFL, Bernhard ; LEHNER, Johannes ; SEIDL, Philipp ; WIDRICH, Michael ; GRUBER, Lukas ; HOLZLEITNER, Markus ; ADLER, Thomas ; KREIL, David ; KOPP, Michael K. u. a.: Hopfield Networks is All You Need. In: *International Conference on Learning Representations*, 2020
- [Rao und Ballard 1999] RAO, Rajesh P. ; BALLARD, Dana H.: Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. In: *Nature neuroscience* 2 (1999), Nr. 1, S. 79–87
- [Ravi und Larochelle 2017] RAVI, Sachin ; LAROCHELLE, H.: Optimization as a Model for Few-Shot Learning. In: *ICLR*, 2017
- [Reed u. a. 2022] REED, Scott ; ZOLNA, Konrad ; PARISOTTO, Emilio ; COLMENAREJO, Sergio G. ; NOVIKOV, Alexander ; BARTH-MARON, Gabriel ; GIMÉNEZ, Mai ; SULSKY, Yury ; KAY, Jackie ; SPRINGENBERG, Jost T. ; ECCLES, Tom ; BRUCE, Jake ; RAZAVI, Ali ; EDWARDS, Ashley ; HEES, Nicolas ; CHEN, Yutian ; HADSELL, Raia ; VINYALS, Oriol ; BORDBAR, Mahyar ; FREITAS, Nando de: A Generalist Agent. In: *Transactions on Machine Learning Research* (2022). – URL <https://openreview.net/forum?id=1ikK0kHjvj>. – Featured Certification. – ISSN 2835-8856
- [Reed u. a. 2016] REED, Scott E. ; AKATA, Zeynep ; MOHAN, Santosh ; TENKA, Samuel ; SCHIELE, Bernt ; LEE, Honglak: Learning What and Where to Draw. In: LEE, D. (Hrsg.) ; SUGIYAMA, M. (Hrsg.) ; LUXBURG, U. (Hrsg.) ; GUYON, I. (Hrsg.) ; GARNETT, R. (Hrsg.): *Advances in Neural Information Processing Systems* Bd. 29, Curran Associates, Inc., 2016. – URL <https://proceedings.neurips.cc/paper/2016/file/a8f15eda80c50adb0e71943adc8015cf-Paper.pdf>
- [Ricci u. a. 2021] RICCI, Matthew ; JUNG, Minju ; ZHANG, Yuwei ; CHALVIDAL, Mathieu ; SONI, Aneri ; SERRE, Thomas: *KuraNet: Systems of Coupled Oscillators that Learn to Synchronize*. 2021
- [Ritter u. a. 2018a] RITTER, Samuel ; WANG, Jane ; KURTH-NELSON, Zeb ; JAYAKUMAR, Siddhant ; BLUNDELL, Charles ; PASCANU, Razvan ; BOTVINICK, Matthew: Been There, Done That: Meta-Learning

- with Episodic Recall. In: DY, Jennifer (Hrsg.) ; KRAUSE, Andreas (Hrsg.): *Proceedings of the 35th International Conference on Machine Learning* Bd. 80, PMLR, 10–15 Jul 2018, S. 4354–4363
- [Ritter u. a. 2018b] RITTER, Samuel ; WANG, Jane X. ; KURTH-NELSON, Zeb ; BOTVINICK, M: Episodic control as meta-reinforcement learning. In: *bioRxiv* (2018), S. 360537
- [Robbins und Monro 1951] ROBBINS, Herbert ; MONRO, Sutton: A stochastic approximation method. In: *The annals of mathematical statistics* (1951), S. 400–407
- [Roberts u. a. 2021] ROBERTS, Nicholas C. ; KHODAK, Mikhail ; DAO, Tri ; LI, Liam ; RE, Christopher ; TALWALKAR, Ameet: Rethinking Neural Operations for Diverse Tasks. In: BEYGELZIMER, A. (Hrsg.) ; DAUPHIN, Y. (Hrsg.) ; LIANG, P. (Hrsg.) ; VAUGHAN, J. W. (Hrsg.): *Advances in Neural Information Processing Systems, 2021*
- [Rosenblatt 1958] ROSENBLATT, Frank: The perceptron: a probabilistic model for information storage and organization in the brain. In: *Psychological review* 65 (1958), Nr. 6, S. 386
- [Rosenblatt 1961] ROSENBLATT, Frank: Principles of neurodynamics. perceptrons and the theory of brain mechanisms / Cornell Aeronautical Lab Inc Buffalo NY. 1961. – Forschungsbericht
- [Rubanova u. a. 2019] RUBANOVA, Yulia ; CHEN, Ricky T. Q. ; DU-VENAUD, David: *Latent ODEs for Irregularly-Sampled Time Series*. 2019
- [Rumelhart u. a. 1985] RUMELHART, David E. ; HINTON, Geoffrey E. ; WILLIAMS, Ronald J.: Learning internal representations by error propagation / California Univ San Diego La Jolla Inst for Cognitive Science. 1985. – Forschungsbericht
- [Rumelhart und McClelland 1987] RUMELHART, David E. ; MCCLELLAND, James L.: *Learning Internal Representations by Error Propagation*. S. 318–362. In: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*, 1987
- [Ruthotto und Haber 2018] RUTHOTTO, Lars ; HABER, Eldad: *Deep Neural Networks Motivated by Partial Differential Equations*. 2018
- [Ruthotto u. a. 2019] RUTHOTTO, Lars ; OSHER, Stanley ; LI, Wuchen ; NURBEKYAN, Levon ; FUNG, Samy W.: *A Machine Learning Framework for Solving High-Dimensional Mean Field Game and Mean Field Control Problems*. 2019

-
- [Ruthotto u. a. 2020] RUTHOTTO, Lars ; OSHER, Stanley J. ; LI, Wuchen ; NURBEKYAN, Levon ; FUNG, Samy W.: A machine learning framework for solving high-dimensional mean field game and mean field control problems. In: *Proceedings of the National Academy of Sciences* (2020)
- [Salakhutdinov u. a. 2007] SALAKHUTDINOV, Ruslan ; MNIH, Andriy ; HINTON, Geoffrey: Restricted Boltzmann machines for collaborative filtering. In: *Proceedings of the 24th international conference on Machine learning*, 2007, S. 791–798
- [Santoro u. a. 2016a] SANTORO, Adam ; BARTUNOV, Sergey ; BOTVINICK, Matthew ; WIERSTRA, Daan ; LILICRAP, Timothy: Meta-Learning with Memory-Augmented Neural Networks. In: *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, JMLR.org, 2016 (ICML'16), S. 1842–1850
- [Santoro u. a. 2016b] SANTORO, Adam ; BARTUNOV, Sergey ; BOTVINICK, Matthew ; WIERSTRA, Daan ; LILICRAP, Timothy: Meta-learning with memory-augmented neural networks. In: *International conference on machine learning* PMLR (Veranst.), 2016, S. 1842–1850
- [Schlag u. a. 2020] SCHLAG, Imanol ; MUNKHDALAI, Tsendsuren ; SCHMIDHUBER, Jürgen: Learning Associative Inference Using Fast Weight Memory. In: *International Conference on Learning Representations*, 2020
- [Schlag und Schmidhuber 2018a] SCHLAG, Imanol ; SCHMIDHUBER, Jürgen: Learning to reason with third-order tensor products. In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018, S. 10003–10014
- [Schlag und Schmidhuber 2018b] SCHLAG, Imanol ; SCHMIDHUBER, Jürgen: Learning to Reason with Third Order Tensor Products. In: *NeurIPS*, 2018, S. 10003–10014
- [Schmidhuber u. a. 1996] SCHMIDHUBER, Juergen ; ZHAO, Jieyu ; WIERING, Marco A.: Simple Principles of Metalearning, 1996
- [Schmidhuber 1992] SCHMIDHUBER, Jürgen: Learning to Control Fast-Weight Memories: An Alternative to Dynamic Recurrent Networks. In: *Neural Computation* 4 (1992), S. 131–139

- [Schmidhuber 2005] SCHMIDHUBER, Jürgen: Completely Self-referential Optimal Reinforcement Learners. In: *International Conference on Artificial Neural Networks*, 2005
- [Schmidhuber u. a. 1997] SCHMIDHUBER, Jürgen ; ZHAO, Jieyu ; WIERING, Marco: Shifting Inductive Bias with Success-Story Algorithm, Adaptive Levin Search, and Incremental Self-Improvement. In: *Machine Learning* 28 (1997), 01
- [Schrimpf u. a. 2018] SCHRIMPF, Martin ; KUBILIUS, Jonas ; HONG, Ha ; MAJAJ, Najib J. ; RAJALINGHAM, Rishi ; ISSA, Elias B. ; KAR, Kohitij ; BASHIVAN, Pouya ; PRESCOTT-ROY, Jonathan ; GEIGER, Franziska ; SCHMIDT, Kailyn ; YAMINS, Daniel L. K. ; DICARLO, James J.: Brain-Score: Which Artificial Neural Network for Object Recognition is most Brain-Like? In: *bioRxiv preprint* (2018). – URL <https://www.biorxiv.org/content/10.1101/407007v2>
- [Schulman u. a. 2018] SCHULMAN, John ; MORITZ, Philipp ; LEVINE, Sergey ; JORDAN, Michael ; ABBEEL, Pieter: *High-Dimensional Continuous Control Using Generalized Advantage Estimation*. 2018
- [Schulman u. a. 2017] SCHULMAN, John ; WOLSKI, Filip ; DHARIWAL, Prafulla ; RADFORD, Alec ; KLIMOV, Oleg: *Proximal Policy Optimization Algorithms*. 2017
- [Serre u. a. 2007] SERRE, Thomas ; WOLF, Lior ; BILESCHI, Stanley ; RIESENHUBER, Maximilian ; POGGIO, Tomaso: Robust object recognition with cortex-like mechanisms. In: *IEEE transactions on pattern analysis and machine intelligence* 29 (2007), Nr. 3, S. 411–426
- [Siegelmann und Sontag 1995] SIEGELMANN, H.T. ; SONTAG, E.D.: On the Computational Power of Neural Nets. In: *Journal of Computer and System Sciences* 50 (1995), Nr. 1, S. 132–150. – ISSN 0022-0000
- [Silver u. a. 2017] SILVER, David ; SCHRITTWIESER, Julian ; SIMONYAN, Karen ; ANTONOGLU, Ioannis ; HUANG, Aja ; GUEZ, Arthur ; HUBERT, Thomas ; BAKER, Lucas ; LAI, Matthew ; BOLTON, Adrian ; CHEN, Yutian ; LILICRAP, Timothy P. ; HUI, Fan ; SIFRE, L. ; DRIESSCHE, George van den ; GRAEPEL, Thore ; HASSABIS, Demis: Mastering the game of Go without human knowledge. In: *Nature* 550 (2017), S. 354–359
- [Simonyan und Zisserman 2014] SIMONYAN, Karen ; ZISSERMAN, Andrew: Very deep convolutional networks for large-scale image recognition. In: *arXiv preprint arXiv:1409.1556* (2014)

- [Snell u. a. 2017a] SNELL, Jake ; SWERSKY, Kevin ; ZEMEL, Richard: Prototypical networks for few-shot learning. In: *Advances in neural information processing systems* 30 (2017)
- [Snell u. a. 2017b] SNELL, Jake ; SWERSKY, Kevin ; ZEMEL, Richard: Prototypical Networks for Few-shot Learning. In: GUYON, I. (Hrsg.) ; LUXBURG, U. V. (Hrsg.) ; BENGIO, S. (Hrsg.) ; WALLACH, H. (Hrsg.) ; FERGUS, R. (Hrsg.) ; VISHWANATHAN, S. (Hrsg.) ; GARNETT, R. (Hrsg.): *Advances in Neural Information Processing Systems* Bd. 30, Curran Associates, Inc., 2017
- [Song und Kingma 2021] SONG, Yang ; KINGMA, Diederik P.: *How to Train Your Energy-Based Models*. 2021
- [Song u. a.] SONG, Yang ; SOHL-DICKSTEIN, Jascha ; KINGMA, Diederik P. ; KUMAR, Abhishek ; ERMON, Stefano ; POOLE, Ben: Score-Based Generative Modeling through Stochastic Differential Equations. In: *International Conference on Learning Representations*
- [Song u. a. 2020] SONG, Yang ; SOHL-DICKSTEIN, Jascha ; KINGMA, Diederik P. ; KUMAR, Abhishek ; ERMON, Stefano ; POOLE, Ben: Score-based generative modeling through stochastic differential equations. In: *arXiv preprint arXiv:2011.13456* (2020)
- [Song u. a. 2021] SONG, Yang ; SOHL-DICKSTEIN, Jascha ; KINGMA, Diederik P. ; KUMAR, Abhishek ; ERMON, Stefano ; POOLE, Ben: Score-Based Generative Modeling through Stochastic Differential Equations. In: *International Conference on Learning Representations*, URL <https://openreview.net/forum?id=PXTIG12RRHS>, 2021
- [Sousa u. a. 2022] SOUSA, João P. ; TAVARES, Rogério ; GOMES, João P. ; MENDONÇA, Vitor: Review and analysis of research on Video Games and Artificial Intelligence: a look back and a step forward. In: *Procedia Computer Science* 204 (2022), S. 315–323. – International Conference on Industry Sciences and Computer Science Innovation. – ISSN 1877-0509
- [Soviany u. a. 2022] SOVIANY, Petru ; IONESCU, Radu T. ; ROTA, Paolo ; SEBE, Nicu: Curriculum learning: A survey. In: *International Journal of Computer Vision* 130 (2022), Nr. 6, S. 1526–1565
- [Srivastava u. a. 2014] SRIVASTAVA, Nitish ; HINTON, Geoffrey ; KRIZHEVSKY, Alex ; SUTSKEVER, Ilya ; SALAKHUTDINOV, Ruslan: Dropout: A Simple Way to Prevent Neural Networks from Overfitting. In: *Journal of Machine Learning Research* 15 (2014), Nr. 56, S. 1929–1958. – URL <http://jmlr.org/papers/v15/srivastava14a.html>

- [Stanley u. a. 2019] STANLEY, Kenneth O. ; CLUNE, Jeff ; LEHMAN, Joel ; MIIKKULAINEN, Risto: Designing neural networks through neuroevolution. In: *Nature Machine Intelligence* 1 (2019), Nr. 1, S. 24–35
- [Storrs und Kriegeskorte 2019] STORRS, Katherine R. ; KRIEGESKORTE, Nikolaus: Deep learning for cognitive neuroscience. In: *arXiv preprint arXiv:1903.01458* (2019)
- [Sun 2019a] SUN, Ruoyu: *Optimization for deep learning: theory and algorithms*. 2019. – URL <https://arxiv.org/abs/1912.08957>
- [Sun 2019b] SUN, Ruoyu: *Optimization for deep learning: theory and algorithms*. 2019
- [Sung u. a. 2018] SUNG, Flood ; YANG, Yongxin ; ZHANG, Li ; XIANG, Tao ; TORR, Philip H. ; HOSPEDALES, Timothy M.: Learning to Compare: Relation Network for Few-Shot Learning. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018
- [Sutskever u. a. 2014] SUTSKEVER, Ilya ; VINYALS, Oriol ; LE, Quoc V.: Sequence to sequence learning with neural networks. In: *Advances in neural information processing systems* 27 (2014)
- [Szegedy u. a. 2013] SZEGEDY, Christian ; ZAREMBA, Wojciech ; SUTSKEVER, Ilya ; BRUNA, Joan ; ERHAN, Dumitru ; GOODFELLOW, Ian ; FERGUS, Rob: Intriguing properties of neural networks. In: *arXiv preprint arXiv:1312.6199* (2013)
- [Team u. a. 2021] TEAM, Open Ended L. ; STOOKE, Adam ; MAHAJAN, Anuj ; BARROS, Catarina ; DECK, Charlie ; BAUER, Jakob ; SYGNOWSKI, Jakub ; TREBACZ, Maja ; JADERBERG, Max ; MATHIEU, Michaël ; MCALEESE, Nat ; BRADLEY-SCHMIEG, Nathalie ; WONG, Nathaniel ; PORCEL, Nicolas ; RAILEANU, Roberta ; HUGHES-FITT, Steph ; DALIBARD, Valentin ; CZARNECKI, Wojciech M.: Open-Ended Learning Leads to Generally Capable Agents. In: *CoRR* abs/2107.12808 (2021). – URL <https://arxiv.org/abs/2107.12808>
- [Tenenbaum u. a. 2011] TENENBAUM, Joshua B. ; KEMP, Charles ; GRIFFITHS, Thomas L. ; GOODMAN, Noah D.: How to Grow a Mind: Statistics, Structure, and Abstraction. In: *Science* 331 (2011), S. 1279 – 1285
- [Thrun und Pratt 1998] THRUN, Sebastian ; PRATT, Lorien: *Learning to Learn: Introduction and Overview*. S. 3–17. In: THRUN, Sebastian

-
- (Hrsg.) ; PRATT, Lorien (Hrsg.): *Learning to Learn*. Boston, MA : Springer US, 1998
- [Tishby und Zaslavsky 2015] TISHBY, Naftali ; ZASLAVSKY, Noga: Deep learning and the information bottleneck principle. In: *2015 ieee information theory workshop (itw)* IEEE (Veranst.), 2015, S. 1–5
- [Tsalakanidou 2020] TSALAKANIDOU, Filareti: AI4Media - A European Excellence Centre for Media, Society and Democracy. (2020), 09
- [Van Den Oord u. a. 2016] VAN DEN OORD, Aäron ; KALCHBRENNER, Nal ; KAVUKCUOGLU, Koray: Pixel recurrent neural networks. In: *International conference on machine learning* PMLR (Veranst.), 2016, S. 1747–1756
- [VanRullen 2016] VANRULLEN, Rufin: Perceptual Cycles. In: *Trends in Cognitive Sciences* 20 (2016), Nr. 10, S. 723–735. – ISSN 1364-6613
- [Vapnik 1999] VAPNIK, Vladimir: *The nature of statistical learning theory*. Springer science & business media, 1999
- [Vapnik 2006] VAPNIK, Vladimir: *Estimation of dependences based on empirical data*. Springer Science & Business Media, 2006
- [Vaswani u. a. 2017] VASWANI, Ashish ; SHAZEER, Noam ; PARMAR, Niki ; USZKOREIT, Jakob ; JONES, Llion ; GOMEZ, Aidan N. ; KAISER, Ł u. ; POLOSUKHIN, Illia: Attention is All you Need. In: GUYON, I. (Hrsg.) ; LUXBURG, U. V. (Hrsg.) ; BENGIO, S. (Hrsg.) ; WALLACH, H. (Hrsg.) ; FERGUS, R. (Hrsg.) ; VISHWANATHAN, S. (Hrsg.) ; GARNETT, R. (Hrsg.): *Advances in Neural Information Processing Systems* Bd. 30, Curran Associates, Inc., 2017
- [Vilalta und Drissi 2002] VILALTA, Ricardo ; DRISSI, Youssef: A Perspective View and Survey of Meta-Learning. In: *Artif. Intell. Rev.* 18 (2002), oct, Nr. 2, S. 77–95. – ISSN 0269-2821
- [Vinuesa und Brunton 2022] VINUESA, Ricardo ; BRUNTON, Steven L.: Enhancing computational fluid dynamics with machine learning. In: *Nature Computational Science* 2 (2022), Nr. 6, S. 358–366
- [Vinyals u. a. 2016] VINYALS, Oriol ; BLUNDELL, Charles ; LILLICRAP, Timothy ; KAVUKCUOGLU, koray ; WIERSTRA, Daan: Matching Networks for One Shot Learning. In: LEE, D. (Hrsg.) ; SUGIYAMA, M. (Hrsg.) ; LUXBURG, U. (Hrsg.) ; GUYON, I. (Hrsg.) ; GARNETT, R.

- (Hrsg.): *Advances in Neural Information Processing Systems* Bd. 29, Curran Associates, Inc., 2016
- [Vinyals u. a. 2015] VINYALS, Oriol ; KAISER, Ł u. ; KOO, Terry ; PETROV, Slav ; SUTSKEVER, Ilya ; HINTON, Geoffrey: Grammar as a Foreign Language. In: CORTES, C. (Hrsg.) ; LAWRENCE, N. (Hrsg.) ; LEE, D. (Hrsg.) ; SUGIYAMA, M. (Hrsg.) ; GARNETT, R. (Hrsg.): *Advances in Neural Information Processing Systems* Bd. 28, Curran Associates, Inc., 2015. – URL <https://proceedings.neurips.cc/paper/2015/file/277281aada22045c03945dcb2ca6f2ec-Paper.pdf>
- [Wang u. a. 2018a] WANG, Haoran ; ZARIPHOUPOULOU, Thaleia ; ZHOU, Xunyu: Exploration versus exploitation in reinforcement learning: a stochastic control approach. In: *arXiv preprint arXiv:1812.01552* (2018)
- [Wang 2021] WANG, Jane X.: Meta-learning in natural and artificial intelligence. In: *Current Opinion in Behavioral Sciences* 38 (2021), S. 90–95
- [Wang u. a. 2018b] WANG, Jane X. ; KURTH-NELSON, Zeb ; KUMARAN, Dharshan ; TIRUMALA, Dhruva ; SOYER, Hubert ; LEIBO, Joel Z. ; HASSABIS, Demis ; BOTVINICK, Matthew: Prefrontal cortex as a meta-reinforcement learning system. 2018
- [Wang u. a. 2021] WANG, Sifan ; WANG, Hanwen ; PERDIKARIS, Paris: Learning the solution operator of parametric partial differential equations with physics-informed DeepONets. In: *Science Advances* 7 (2021), Nr. 40, S. eabi8605
- [Watson 1964] WATSON, Geoffrey S.: Smooth regression analysis. In: *Sankhyā: The Indian Journal of Statistics, Series A* (1964), S. 359–372
- [Wei u. a. 2022] WEI, Jason ; TAY, Yi ; BOMMASANI, Rishi ; RAFFEL, Colin ; ZOPH, Barret ; BERGEAUD, Sebastian ; YOGATAMA, Dani ; BOSMA, Maarten ; ZHOU, Denny ; METZLER, Donald u. a.: Emergent abilities of large language models. In: *arXiv preprint arXiv:2206.07682* (2022)
- [Wen u. a. 2022] WEN, Gege ; LI, Zongyi ; AZIZZADENESHELI, Kamyar ; ANANDKUMAR, Anima ; BENSON, Sally M.: U-FNO—An enhanced Fourier neural operator-based deep-learning model for multi-phase flow. In: *Advances in Water Resources* 163 (2022), S. 104180
- [Werbos 1982] WERBOS, P.J: Applications of advances in nonlinear sensitivity analysis. (1982)

-
- [Weston u. a. 2014] WESTON, Jason ; CHOPRA, Sumit ; BORDES, Antoine: Memory networks. In: *arXiv preprint arXiv:1410.3916* (2014)
- [Williams und Rasmussen 1995] WILLIAMS, Christopher ; RASMUSSEN, Carl: Gaussian processes for regression. In: *Advances in neural information processing systems* 8 (1995)
- [Wiskott und Sejnowski 2002] WISKOTT, Laurenz ; SEJNOWSKI, Terrence J.: Slow feature analysis: Unsupervised learning of invariances. In: *Neural computation* 14 (2002), Nr. 4, S. 715–770
- [Wittenberg 2010] WITTENBERG, George F.: Experience, cortical remapping, and recovery in brain disease. In: *Neurobiology of Disease* 37 (2010), Nr. 2, S. 252–258. – *Frontiers in Brain Repair*. – ISSN 0969-9961
- [Wu u. a. 2013] WU, Si ; WONG, KY M. ; TSODYKS, Misha: *Neural information processing with dynamical synapses*. 2013
- [Xie u. a. 2021] XIE, Sang M. ; RAGHUNATHAN, Aditi ; LIANG, Percy ; MA, Tengyu: *An Explanation of In-context Learning as Implicit Bayesian Inference*. 2021
- [Xu u. a.] XU, Keyulu ; LI, Jingling ; ZHANG, Mozhi ; DU, Simon S. ; KAWARABAYASHI, Ken-ichi ; JEGELKA, Stefanie: What Can Neural Networks Reason About? In: *International Conference on Learning Representations*
- [Xu u. a. 2018] XU, Zhongwen ; HASSELT, Hado van ; SILVER, David: Meta-gradient reinforcement learning. In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018, S. 2402–2413
- [Yamins und DiCarlo 2016] YAMINS, Daniel L. ; DICARLO, James J.: Using goal-driven deep learning models to understand sensory cortex. In: *Nature neuroscience* 19 (2016), Nr. 3, S. 356–365
- [Yamins u. a. 2014] YAMINS, Daniel L. ; HONG, Ha ; CADIEU, Charles F. ; SOLOMON, Ethan A. ; SEIBERT, Darren ; DICARLO, James J.: Performance-optimized hierarchical models predict neural responses in higher visual cortex. In: *Proceedings of the national academy of sciences* 111 (2014), Nr. 23, S. 8619–8624
- [Yoon u. a. 2022] YOON, Tehrim ; SHIN, Sumin ; YANG, Eunho: Learning Polymorphic Neural ODEs With Time-Evolving Mixture. In: *IEEE*

-
- Transactions on Pattern Analysis and Machine Intelligence* 45 (2022), Nr. 1, S. 712–721
- [Yu u. a. 2019] YU, Tianhe ; QUILLEN, Deirdre ; HE, Zhanpeng ; JULIAN, Ryan ; HAUSMAN, Karol ; FINN, Chelsea ; LEVINE, Sergey: Meta-World: A Benchmark and Evaluation for Multi-Task and Meta Reinforcement Learning. In: *Conference on Robot Learning (CoRL)*, URL <https://arxiv.org/abs/1910.10897>, 2019
- [Zaremba und Sutskever 2016] ZAREMBA, Wojciech ; SUTSKEVER, Ilya: *Reinforcement Learning Neural Turing Machines - Revised*. 2016
- [Zhang u. a. 2021] ZHANG, Chiyuan ; BENGIO, Samy ; HARDT, Moritz ; RECHT, Benjamin ; VINYALS, Oriol: Understanding deep learning (still) requires rethinking generalization. In: *Communications of the ACM* 64 (2021), Nr. 3, S. 107–115
- [Zhang 2013] ZHANG, Haizhang: Vector-valued reproducing kernel Banach spaces with applications to multi-task learning. In: *Journal of Complexity* 29 (2013), Nr. 2, S. 195–215
- [Zhang u. a. 2009] ZHANG, Haizhang ; XU, Yuesheng ; ZHANG, Jun: Reproducing Kernel Banach Spaces for Machine Learning. In: *Journal of Machine Learning Research* 10 (2009), Nr. 12
- [Zhang u. a. 2020] ZHANG, Han ; GAO, Xi ; UNTERMAN, Jacob ; AR-ODZ, Tom: *Approximation Capabilities of Neural ODEs and Invertible Residual Networks*. 2020
- [Zhang u. a. 2019] ZHANG, Tianjun ; YAO, Zhewei ; GHOLAMI, Amir ; GONZALEZ, Joseph ; KEUTZER, Kurt ; MAHONEY, Michael W. ; BIROS, George: ANODEV2: A Coupled Neural ODE Framework. In: *Neural Information Processing Systems*, 2019
- [Zhang und Zhou 2017] ZHANG, Wei ; ZHOU, Bowen: *Learning to update Auto-associative Memory in Recurrent Neural Networks for Improving Sequence Memorization*. 2017
- [Zhou u. a. 2019] ZHOU, Teng ; SONG, Zhen ; SUNDMACHER, Kai: Big data creates new opportunities for materials research: a review on methods and applications of machine learning for materials design. In: *Engineering* 5 (2019), Nr. 6, S. 1017–1026
- [Zhu u. a. 2021] ZHU, Qunxi ; GUO, Yao ; LIN, Wei: Neural delay differential equations. In: *arXiv preprint arXiv:2102.10801* (2021)

[Zintgraf u. a. 2019] ZINTGRAF, Luisa ; SHIARLIS, Kyriacos ; IGL, Maximilian ; SCHULZE, Sebastian ; GAL, Yarin ; HOFMANN, Katja ; WHITESON, Shimon: Varibad: A very good method for bayes-adaptive deep rl via meta-learning. In: *arXiv preprint arXiv:1910.08348* (2019)

Title: Learning higher-order functions for computation, memorization and control in artificial neural networks

Keywords: Meta-learning, neural networks, control theory, working memory, synaptic plasticity, reinforcement learning, neural operators

Abstract: After decades of research, the question of how biological neural networks synthesize experience to serve higher-level cognitive processes (concept acquisition, systematic decision making, evaluative thinking, creativity...) still resists to a complete scientific understanding. At the same time, advances in large scale computation have enabled the development of their artificial counterparts, which currently power a revolution in machine intelligence. These simplified neural models developed through a global scientific trial-and-error process, present however evident behavioral differences and limitations compared to biological neural networks and provide unsatisfactory to little insight into the computational principles that evolution condensed in biological neurons. One evident limitation is the lack of compositionality and systematicity in artificial neural computation, revealing their current failure as a true functional programming paradigm. In an attempt to identify key neural mechanisms able to support more abstractive

machine intelligence, this thesis propose to explore the problem of learning higher-order functions (i.e functions of functions) with neural networks. Namely, we propose to marry modern machine learning techniques with the perspective of neural networks as controllable *functionals*, unearthing original computational mechanisms able to support powerful faculties such as dynamic associative memorization or functional operator regression. In particular, building on the clear epistemological trend that hand-designed methods are eventually replaced by computerized and self-executing solutions, we will consider the specific higher-order problem of learning to learn, i.e *meta-learning*. We show that granting neural networks the aptitude to design learning strategies with minimal solution constraints as a function of data, activity or choice history, can help explore new forms of adaptive programs, with application to low-shot learning or robotic control.