UNIVERSITY OF CAMBRIDGE

Out-of-distribution generalisation in machine learning

Agnieszka Słowik

Lucy Cavendish College

This dissertation is submitted for the degree of Doctor of Philosophy.

December 2022

To my mum Izabela.

Declaration

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except as declared in the Preface and specified in the text. It is not substantially the same as any that I have submitted, or am concurrently submitting, for a degree or diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. I further state that no substantial part of my dissertation has already been submitted, or is being concurrently submitted, for any such degree, diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. This dissertation does not exceed the prescribed limit of 60 000 words.

Abstract

Out-of-distribution generalisation in machine learning

Agnieszka Słowik

Machine learning has proven extremely useful in many applications in recent years. However, a lot of these success stories stem from evaluating the algorithms on data very similar to that they were trained on. When applied to a new data distribution, machine learning algorithms have been shown to fail. Given the non-stationary and heterogeneous nature of real-world data, a better grasp of out-of-distribution generalisation is needed for algorithms to be widely deployed and trusted.

My thesis presents three research studies that aim to investigate and develop the field of out-of-distribution generalisation. The central goal of these research efforts is to produce new tools, such as algorithms, theoretical results, experimental results and datasets, to improve understanding and performance of machine learning methods in the face of distribution shift. The high-level idea that drives these research efforts across three machine learning scenarios is *modularity* – the quality of consisting of separate parts that form a whole when combined. Modular approaches are hypothesised to steer the machine learning methods away from rigid memorisation of examples and towards more flexible and 'more intelligent' learning that supports generalisation.

In my first contribution, I approach the thesis goal from the perspective of learning from multiple training distributions. The contribution to this line of research is twofold. First, I present a new standardised suite of tasks for evaluation and comparison of out-of-distribution generalisation algorithms. Second, I state a set of new theoretical results that fill an existing gap between data-centric and algorithmic approaches to out-of-distribution generalisation. These theoretical findings guide a new set of practical recommendations on how to employ the algorithmic approach.

In the second contribution, I tackle generalisation in the common learning setup of supervised image recognition. In this context, I first investigate the effect of multi-level feature aggregation on generalisation, and demonstrate that augmentation with one of the considered methods consistently improves the performance. Second, I propose a set of

simple image datasets that can be used as a stepping stone for evaluation and comparison of image classification methods in terms of out-of-distribution generalisation.

Finally, I delve into the learning scenarios where multiple neural networks communicate to solve a shared task. This work supports the thesis goal in two ways. First, I propose a new environment, *graph referential games*, and present results on the influence of data representation and the corresponding data representation learning methods on out-of-distribution generalisation. These results connect the previously disjoint fields of graph representation learning and emergent communication. Second, I tackle the challenging domain of population-based communication grounded in realistic images.

The datasets, algorithms, theorems and experimental results in this thesis represent a few steps towards understanding and improving out-of-distribution generalisation in machine learning. They provide researchers with new tools and results that aim to foster research in this field, some of which have already proved useful to the research community. Finally, this work suggests important future directions in the machine learning subfields of learning from multiple distributions, image classification and multi-agent communication.

Agnieszka Słowik
December 2022

Acknowledgements

Reaching the end of this journey has only been possible because of the support of so many.

Starting at the end, I would like to thank my examiners **Ferenc Huszár** and **Artur d'Avila Garcez** for taking their time to read my thesis, and to thoroughly discuss it with me (for nearly three hours!). I could not have hoped for a more satisfying conclusion to these four years of work. Thank you for helping me flesh out new links between this thesis and the most recent research developments in machine learning. Your valuable feedback allowed me to strengthen this thesis further before its publication.

This thesis would not have been possible without my supervisors, **Sean Holden** and **Mateja Jamnik**. First of all, thank you for taking me on as a PhD student, which gave me an opportunity to both benefit from and contribute to the community at the Department of Computer Science and Technology. Thank you for your unwithering support throughout the last four years and for your invaluable and detailed feedback on each chapter of this thesis. Apart from your contributions to the research that has made its way to this dissertation, I am also forever grateful for your support in my broader career development, which has allowed me to grow as a young scientist, and has reinforced my strong drive to pursue a career in research after my PhD. Even though doing a PhD in Artificial Intelligence at Cambridge probably exceeds my childhood dreams, Sean is definitely the 'I want to be like him when I grow up' material. My family used to oppose my music taste and anything remotely edgy when I was a teenager, yet since the start of my PhD my father has referred to Sean as the 'heavymetalowiec' with a lot of esteem, so thank you for changing their perspective on long hair, leather jackets and piercing.

Teaching, mentoring and science communication lead to two-way knowledge exchange. I am very grateful to everyone who has given me the opportunity to present and to discuss my research, to teach and to mentor students, and to manage research projects. I am thankful to the organisers and the students of **AI4Good Lab 2020** for two months of teaching (and learning!) ML and for being able to supervise the team of students behind *EthicAI*, the winning academic project of the year. Thank you for your challenging questions and

for your enthusiasm and diligence even despite the remote format of the school. I would also like to thank the organisers and the students who took part in the **TryAI** workshop co-located with *AAAI 2020* – they received the well-deserved 'most in-depth' presentation prize. I am excited to see where you go in your careers and I am certain that you will excel in any project you choose to pursue. I am also forever grateful to **Tobias Kohn** for asking me to give a lecture at the **Experience Cambridge** outreach event when I was only in my first year. His feedback on my teaching has meant a lot to me and I still cherish it as one of the highlights of my PhD. I am also thankful to all the students I taught in undergraduate supervisions at Cambridge. Last but not least, I thank all the people behind the initiatives and the venues who invited me to give a talk during my PhD. In no particular order, this has included: **Oxbridge Women in Computer Science Conference**, **ML in PL**, **Women@CL**, **Schmidt Data for Science Residency Programme**, **Cambridge Spark**, **AI Research Group Talks at the Univeristy of Cambridge**, **GMUM Seminar at the Jagiellonian University**, **AI & NLP Day**, **Polish Science Cafe by Cambridge University Polish Society**, **ClusterAI**, **Causality & Domain Adaptation Reading & Work Group led by Ferenc Huszár**, **Institute of Mathematics of the Polish Academy of Science**, **DELTA Research Group at UCL Statistical Science led by Omar Rivasplata**, **Perspektywy Women in Tech Summit**; finally, the workshops where I gave short/spotlight/contributed talks, including: **NeurIPS Workshop on Causal Discovery and Causality-Inspired Machine Learning (CDCI 2020)**, **NeurIPS workshop on Algorithmic Fairness through the Lens of Causality and Robustness (AFCR 2021)** and **NeurIPS Workshop on Optimization for Machine Learning (OPT 2021)**. The invaluable feedback and the skills obtained through these experiences strengthened my research and my confidence.

I have been incredibly fortunate to work with stellar scientists and kind people both before my PhD and throughout my PhD. The pre-PhD academic experiences that ignited my passion for machine learning research and set me on the right track for accomplishing my dream of pursuing a PhD were my BSc and MSc theses under the supervision of **Wojciech Czarnecki** and the **Julian Hall** and **Amos Storkey** duo. Later on, when I served as a supervisor of research projects myself, (for example, when I supervised a team of students at *AI4Good Lab 2020*) I was striving to be as responsible as Wojtek. Julian has continued to motivate me by checking up on me at various stages of my life. I am always humbled and encouraged by his faith in me as a researcher, and I have been inspired by his research rigor, humility and dedication to his students. The experiences in *BayesWatch*, the research group led by Amos, prepared me well for my subsequent jumps in at the deep ends, such as applying for a PhD at Cambridge and my collaborations with Léon Bottou and Yoshua Bengio during my PhD. Finally, it would be unfair not

to acknowledge my industrial pre-PhD mentors who helped me develop the necessary technical skills and the can-do attitude: **Raffael Strassnig** and **JinSung Kang**. I wish everyone had such great managers, especially at the early career stages. I am forever inspired by Raffael's dedication and strong work ethic – staying after hours to do pair programming with me and constant support from the day one until my final presentation – and by the truly family-like atmosphere that Jin was able to create in his team during my time at Architech, the memories of which I still cherish. Thanks to these excellent mentors, I was able to start developing crucial skills for my PhD (both technical skills and soft skills) well ahead of the start date of my PhD programme in January 2019.

Moving on in the chronological order, it is time to acknowledge my mentors, collaborators and friends from Mila and from the rich AI ecosystem based in Montreal: **Will Hamilton**, **Abhinav Gupta**, **Koustuv Sinha**, **Yoshua Bengio**, **Anirudh Goyal**, **Alex Lamb**, **Philippe Beaudoin**, **Joelle Pineau**, **Doina Precup**, **Irina Rish**. Joelle, Doina and Irina are real role models on so many levels and anyone would be lucky to work with them. They are those exceptional people who can have a huge positive impact and encourage you to be a better person even through a very brief interaction. I am grateful to Philippe for his interest and faith in me. Alex and Anirudh found times in their busy schedules to collaborate with me on the Neural Function Modules project, and to keep it going despite the complete chaos and uncertainty of the early months of COVID-19. Alex is great at communication and a very respectful, kind and patient individual, with an enormous knowledge on machine learning. Similarly, I was lucky to work with Koustuv, an incredibly warm and open researcher, who makes it so easy and pleasant to work with him. Abhinav spent time teaching me the ins and outs of modern machine learning that are difficult to acquire through courses and books. There is something magical about writing code at 2am in Starbucks while it snows outside, and there is even more magic in being able to share the deadline stress and Reviewer 2 woes with someone. Finally, none of the above would have been possible without Will Hamilton, who accepted me for the research visit at Mila under his mentorship. It is difficult to imagine someone with a better 'academic brilliance' to 'great communication and humility' ratio than Will. This was my experience with Will and I am certain that all his students and co-workers would agree. I have so much nostalgia for the brilliant and ambitious yet easy to work with people of Mila and Montreal – but perhaps, as the song goes, *Je reviendrai à Montréal*!

In terms of giving me the proper introduction to the fascinating and rich field of out-of-distribution generalisation in machine learning, which I barely had space to cover in this thesis, credit goes to **Léon Bottou**, **David Lopez-Paz** and **Benjamin Aubin**. I have been lucky to work with you and this research was pivotal both for my PhD and for the work I am planning to undertake after graduation. I see the entire collaboration with Léon

(including the DRO work) as my biggest academic achievement and the biggest career achievement up-to-date. I have learnt so much from the meetings with Léon and I was able to present the results of our work at multiple venues (*AAAI*, *AISTATS*, two contributed talks at *NeurIPS* workshops), and even collect awards based on this work. I am extremely proud of the results of our work and the skills I developed along the way, especially of the combination of mathematical rigour and the courage to tackle 'big questions' of high tangible impact, such as the question of the source of bias in machine learning.

I am also forever grateful to **Marco Baroni** and **Roberto Dessì**. We found a shared language (apart from Italian) thanks to our mutual interest in cognitive science, linguistics and philosophy. Great researchers and extremely cool people that I would love to meet in person. Their group is one of the best examples of 'how to work as a researcher in machine learning nowadays and remain sane', which I find very important. Same goes for **Kory Mathewson**, whom I was lucky to have met through our brief collaboration in emergent communication. Kory has been a huge source of inspiration on how to combine technical and human-oriented interests in a career in AI. Looking forward to finding out what he comes up with for his next birthday in 2023!

In Cambridge, I am lucky to have met **Larry Paulson**, **Javier Antorán**, **Nick Pawlowski** (ok, Nick and I had met at ICLR in Toulon prior to my PhD/during his PhD), **Krzysztof Maziarz**, **Marcin Machura**, **David Krueger**, **Ferenc Huszár**, **Paul Scherer**, **Raoul-Gabriel Urma** and **Women@CL crew**, among many other brilliant and kind people. Larry, Javier, Nick and Krzysztof have seemed to believe in me in a way which gave me a huge push at various low points of my PhD. I appreciate all the encouragement and practical help, and I have been lucky and honoured to have you all as friends.

At Cambridge I have also met my husband **Andrej Ivašković**. No words can describe how much he has supported me and how much he has improved my overall experience over the last years. He is the most reliable, loyal, caring, kind and hard-working person that I have ever known. He cares about what matters to me and he has largely inspired me to improve my coding and presentation skills, among others. Despite his lack of enthusiasm (ekhm) for certain aspects of modern machine learning research, he has patiently listened to my research ideas and plans, and did everything to support me and help me achieve my goals. We only met towards the end of 2018 and since then we have been through two PhDs, long-distance relationship, lockdowns, travels, multiple virtual and non-virtual conferences, health problems, multiple research groups, multiple teaching gigs and workplaces, multiple living addresses, getting married and organising a wedding, research and teaching awards and ICPC coaching, full-time job search times two, all sorts of ups and downs... It has been amazing to grow with you in the last years. I feel that I have known you since forever and everything about you feels so right.

To my mum **Izabela** and brother **Andrzej**. My mum is the answer to how I have managed to be so productive in my 20s as a single mum raised, first generation immigrant and first generation PhD student. I dedicate this thesis to her.

Lastly, I want to say thank you to all the people who made my life better during these challenging years, including my invaluable support network in and from Poland – my father **Witold Słowik**, **Filip Szczypiński** (thanks to whom my wedding was way better than I had ever imagined), **Mateusz Malinowski** (who advised me at a career crossroads and who is a huge source of inspiration and motivation for me in my career and beyond), my godfather **Jerzy Pisuliński**, **Ania Pisulińska-Kuśmierczyk**, **Barbara Gierat-Kucharzewska**, **Andrzej Konsor**, **Henryk Michalewski** (thank you for all the kind words, your support and for setting an example as a stellar machine learning scientist!), **Sanket Kamthe** (thank you for being my friend, I'm looking forward to the housewarming party), **Ewa Kluczewska**, **Benedek Rozemberczki** (thank you for your help and advice!), my cousin **Miriam Casali** and her family, my new Serbian family – **Milka Knežević Ivašković**, **Slobodan Ivašković** and **Stefan Ivašković** – **Gosia Sumlet** (for being – officially – the best physiotherapist and – unofficially – the best psychotherapist in the world), **Eva Karska** (thank you for checking up on me!), **Andrey Vasilyev**, **Gosia Nguyen**, **Tad Adamek**, **Adam Baranowski**, **Piotr Migdał**, **Aleksandra Przegalińska**, **Alessandro Nicola Malusà**, **Gopal Kotecha**, **Robert Pinsler** (the weekend with Robert and his friends during the *Science: Polish Perspectives* conference in 2017 gave me the final push to prioritise my PhD application with Sean in Cambridge), **Hayley Leeman**, **Miloš Stanojević** and other inspiring, helpful and kind people I am lucky to have in my life.

Contents

Chapter 1

Introduction

It is good to have an end to journey towards;
but it is the journey that matters, in the end.
–Ursula K. Le Guin,
The Left Hand of Darkness (1969)

My thesis explores the circumstances under which machine learning models fail when given inputs dissimilar from the ones encountered during their training, and explores techniques that address or aim to prevent such failures.

I divide this introduction into four sections: Section 1.1 motivates the research through an explanation and discussion of out-of-distribution generalisation, and failure modes when not addressed; Section 1.2 poses the set of research questions and the hypothesis investigated in this thesis; Section 1.3 provides an outline of the remainder of this thesis; finally, Section 1.4 presents the publications that I have produced and presented during my PhD, the content of which overlaps with the content of this thesis.

1.1 Motivation

Is memorisation the same as learning? *Funes the Memorious*, a short story by writer Jorge Luis Borges (translated to English by James E. Irby [71, p.59–66]), describes a boy called Funes who gains perfect memory after receiving a head injury. He starts to remember every moment of his life in great detail. At the same time, he loses the ability to *generalise*: his memories are disconnected from each other. For example, he sees the same dog from different angles only to consider each side of the same dog as an independent piece of information. He does not even understand what his own body looks like ('His own face in

the mirror, his own hands, surprised him every time he saw them'), which leads to the conclusion: 'To think is to forget a difference, to generalise, to abstract. In the overly replete world of Funes, there were nothing but details.'.

Much like Funes, modern *neural networks* with millions of parameters have been shown to memorise training examples, which can lead to an array of problems such as (1) high sensitivity to noisy data [150, 221], (2) falling for *adversarial attacks* [271, 87, 269, 287], (3) sample inefficiency in comparison to human learning [302, 303, 275], and (4) poor generalisation to new data [62], even when new data samples bare intuitive similarities to what the model has been trained on [61, 251]. These problems can arise in any domain where modern machine learning is applied. They can lead to further practical consequences over the course of use of machine learning systems, such as opaque failure modes, resulting in a lack of trust in machine learning systems [297].

Out-of-distribution generalisation is a missing ability in standard machine learning methods. These methods are backed by *statistical learning theory* [279], which justifies the use of average-based optimisation (Empirical Risk Minimisation [279]) and the practice of estimating generalisation error using a test set. However, this theory assumes that training (past) and test (future) data are *independent and identically distributed*. This assumption is incorrect in many practical domains where machine learning is applied: real-world data is heterogeneous and its distribution usually shifts over time. Practical sources of *distribution shifts* include changes in the characteristics of the users of machine learning systems, or changes in the environment where an embodied agent is placed. Another common example of distribution shift is a result of the dynamic nature of language, including the languages used online. The constant evolution of natural language has been shown to change the perplexity of language models when applied repeatedly across several months [164]. Section 2.4 in the Background chapter covers more types of distribution shift and corresponding examples. As a result of these shifts, achieving nearly 100% on the commonly used in-distribution test sets is not always indicative of future performance, as has been demonstrated by numerous papers [137, 15, 61, 235, 204, 62].

The topic of out-of-distribution generalisation (OOD generalisation) in machine learning is essentially as broad and complex as the field of machine learning itself, and equally prone to passing trends and opposing views within the research community. In my view, improved generalisation in the face of distribution shifts is needed because of the following sets of reasons:

- *engineering reasons* – to improve sample efficiency and to improve performance in low-resource domains without thousands of training examples [110];

- *scientific reasons* – to gain insight into how neural networks learn and to potentially bring machine learning closer to human learning;
- *business reasons* – to employ neural networks in increasingly nuanced tasks currently performed by humans;
- and *societal reasons* – to *debias* machine learning systems by controlling the *simplicity bias* [246]. Exploiting 'shortcuts' in data can lead to unfair solutions (for instance, this can be seen in recruitment tools exploiting gender information [59]).

During my PhD, I have been asking myself:

> What kind of tools would the machine learning research communities working on on out-of-distribution generalisation benefit from the most?

This thesis aims to provide such tools in the form of new datasets, new theoretical results, new testbeds, new experimental results and new algorithms. The concrete outcomes of these research efforts are summarised in Figure 1.1.

The research efforts that led to this thesis have been carried out within three subfields of machine learning: *learning from multiple distributions* (Chapter 3), *image classification* (Chapter 4) and *multi-agent communication* (Chapter 5). This broad perspective allows me to gather more evidence to support the central hypothesis and to explore the research questions (Section 1.2). At the same time, the tools presented in this thesis aim to be of use to several machine learning communities that I have been fortunate to work with and to learn from during my PhD: (1) the invariant learning and group robustness communities (Chapter 3), (2) the vision community (Chapter 4), and (3) the emergent communication community (Chapter 5). All of these communities have been independently investigating out-of-distribution generalisation in machine learning, as I have reviewed in the background chapter (Chapter 2) and in the respective contribution chapters. This thesis connects previously disjoint communities within which I pursued my research, such as graph neural networks [141] with emergent communication [43] (Chapter 5), as well as data-oriented approach to group robustness [36] with Distributionally Robust Optimisation [21] (Chapter 3).

1.2 Research questions

Out-of-distribution generalisation in machine learning is a broad and challenging topic. The contributions of this thesis are guided by a high-level idea:

> Modularity can improve out-of-distribution generalisation in machine learning.

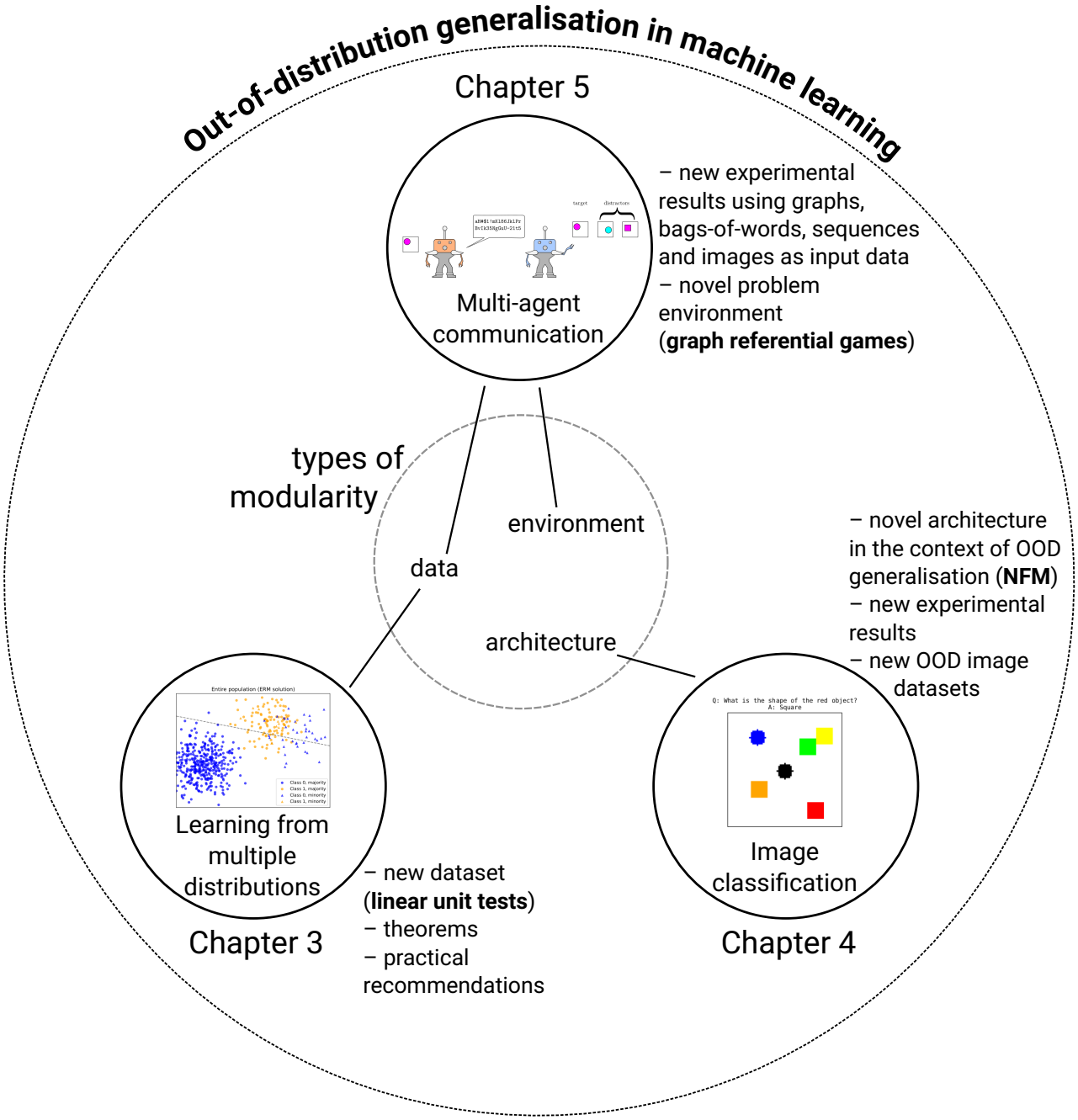


Figure 1.1: A visual summary of the research directions within out-of-distribution generalisation that are explored in this thesis.
Concrete research outputs are summarised and listed for each chapter.

This hypothesis is loosely inspired by cognitive neuroscience and *global workspace theory* [12], which posits that brain is composed of specialised modules ('experts') that communicate sparingly. In contrast, most neural networks are monolithic architectures that learn from a single dataset under the independent and identically distributed data assumption.

At the conceptual level, replacing monolithic approaches with modular solutions seems promising in the context of improving out-of-distribution generalisation. Many instances of OOD generalisation, such as *compositional generalisation* investigated in Chapter 4 and Chapter 5, are problems that can be solved by extracting *reusable pieces* from data and by recomposing them according to some rules. Having learnt the meaning of a *red square* and a *blue circle*, a human is capable of observing two distinct concepts that jointly define these examples: shape and colour. With these separate mental *modules*, it is possible to understand the new concept of a *red circle* by reusing the pieces of information obtained through different, yet systematically and structurally similar examples seen in the past. Non-modular understanding driven by raw memorisation of examples and their meanings can yield a high accuracy on familiar data, however, it fails to extract meaningful patterns, such as compositional rules, that can be used to generalise to a new data distribution. Modularity is only one of the possible *inductive biases* [90] that might allow humans to generalise – however, it is an important first step which can be incorporated at various stages of the standard machine learning setup.

Chapter 3 incorporates the concept of modularity *at the data level* by acknowledging that data often come from multiple heterogenous sources. This is in contrast to the standard monolithic approach of learning from a single training dataset under the independent and identically distributed data assumption. I discuss how a modular cost function (with components corresponding to different data sources) can be used to find a model with more consistent performance across different input distributions. Chapter 4 incorporates modularity *at the architecture level* by proposing Neural Function Modules [159], a design choice that increases modularity of a neural network and specialisation of its layers, along with new extensive results in the context of out-of-distribution generalisation. Finally, Chapter 5 investigates modularity *at the environment level* by using a learning setup with two or more neural networks separated by a discrete communication channel. Additionally, this chapter also approaches the main hypothesis by comparing data representations of the varying degree of modularity and structure in terms of OOD generalisation in the proposed graph referential games [259].

Apart from exploring the umbrella hypothesis on the benefits of modularity in the context of out-of-distribution generalisation, the research efforts presented in this thesis also aim to answer a set of focused, fine-grained research questions.

1. *What is the relation between existing data-centric and algorithmic approaches to improving OOD generalisation (and group robustness)?*

   As further described in Section 2.4, generalisation has been previously approached through data-centric methods, such as re-weighting the training examples, or algorithmic methods, such as Distributionally Robust Optimisation (DRO) [21]. The existing gap between these two approaches leads to contentious debates on the origins of machine learning bias. Chapter 3 provides new theorems (Theorem 1 and Theorem 2) that address this gap along with a set of new practical recommendations (Section 3.4.3) on how to use DRO in light of the new theoretical findings.

2. *Do methods that encourage multi-level feature aggregation help in improving OOD generalisation in image classification?*

   Convolutional Neural Networks (Section A.1.2) are the standard backbone of vision algorithms, and they are known to focus on local features in an image. However, there exist variants of convolution that aim to integrate both local and global features, such as dilated convolutions (Section 4.1.1.1). For example, Dilated DenseNets [119] (Section 4.1.1.2) have been shown to improve the performance in relational reasoning [6], which is a task that requires scene understanding through the use of both local and global image features. Neural Function Modules (Section 4.1.2) are another approach to multi-level feature aggregation by using attention (Section 2.3.2) and by combining bottom-up and top-down feedback (Figure 4.4). Chapter 4 investigates whether these approaches to multi-level feature aggregation are useful in the context of improving out-of-distribution generalisation, which leads to a new set of strong results in favour of using Neural Function Modules specifically in the context of out-of-distribution generalisation in image classification.

3. *Do graph reprentations induce a better OOD generalisation in multi-agent games? Does increasing the number and diversity of agents improve OOD generalisation?*

   These two questions are investigated in Chapter 5. Graph representations have been hypothesised to improve combinatorial generalisation broadly in machine learning [19]. The degree of structure in input data to multi-agent communication games has been hypothesised to be correlated with generalisation and *language compositionality* [262] – however, graph representations have not been previously attempted in emergent communication previously. I investigate this hypothesis by proposing graph referential games, where graph representations are compared with the corresponding baselines in terms of both out-of-distribution generalisation and the related concept of language compositionality. The final study in this chapter aims to explore the effect of the number and diversity of agents on the out-of-distribution generalisation in multi-agent

games grounded in images. The conclusion corroborates the results obtained in the similar context of reconstruction games [232]: unlike in the human studies [191, 181], the increase in the population size does not lead to an increase in in-distribution or out-of-distribution performance and language interpretability.

1.3 Thesis outline

Figure 1.1 shows a visual summary of the core research chapters and their roles in the main story.

This thesis is structured as described below. The order of chapters represents progression towards increasingly noisy and complex machine learning scenarios, where out-of-distribution generalisation is investigated. First, I approach the thesis goals using **mathematical tools** and **linearly separable data**, where spurious and invariant features can be perfectly disentangled. Next, I delve into out-of-distribution generalisation from the perspective of **image classification**, where meaningful features are derived from entangled pixel data by a neural network. Finally, I explore out-of-distribution generalisation in a setting where multiple neural networks solve a shared task through **multi-agent communication**.

Chapter 2: Background This chapter starts with a brief overview of the research methodology used throughout the contribution chapters. The next section is an overview of the necessary technical background that was required to carry out the research presented in Chapter 4 and Chapter 5. The last section defines out-of-distribution generalisation and reviews existing literature on this topic, with the focus on the perspectives that are later expanded in the contribution chapters: data-driven and algorithmic approaches (Chapter 3), image classification (Chapter 4) and multi-agent communication (Chapter 5).

Chapter 3: Learning from multiple distributions Training data often come from multiple systematically different sources. This chapter presents research on out-of-distribution generalisation in machine learning from the perspective of learning from multiple training distributions. I present two novel contributions: (1) *Linear unit tests*, a set of tasks that probe OOD algorithms, and (2) new theoretical results (Theorem 1 and Theorem 2) that fill the gap between data-driven and algorithmic approaches to generalisation, along with a set of practical recommnedations. This work aims to increase transparency and understanding in evaluation of OOD algorithms: on the one hand, by proposing a standardised battery of unit tests, and on the other hand, by explaining the relation between two previously disjoint approaches in the context of learning from multiple distributions.

Chapter 4: Out-of-distribution generalisation in image classification This contribution is focused on image classification under the assumption of a single training distribution, which is the most prevalent learning setup in machine learning. I expand this line of research by (1) presenting *Neural Function Modules* with an array of new experimental results (Section 4.2) that show the advantages that *Neural Function Modules* bring in the context of out-of-distribution generalisation in image classification, and (2) releasing a set of lightweight OOD image datasets that can be used as the first stepping stone in evaluation and comparison of new image classification methods. This work demonstrates the advantages and flexibility of *Neural Function Modules* in the context of out-of-distribution generalisation in image classification.

Chapter 5: Out-of-distribution generalisation in multi-agent systems This contribution is focused on out-of-distribution generalisation from the perspective of multi-agent communication, where agents are parameterised by neural networks. Firstly, I contribute to this field by introducing *graph referential games* along with the results on the influence of data representation and the corresponding data representation learning methods on out-of-distribution generalisation. The results bridge two previously disjoint fields of graph representation learning and emergent communication. Secondly, I investigate OOD generalisation in the challenging task of many-to-many communication grounded in realistic images. The results presented in this chapter corroborate the hypotheses that (1) more structured input data lead to more structured/compositional language [262, 163] and (2) graph representation learning can improve compositional generalisation in machine learning [19].

Chapter 6: Conclusion and further directions This final chapter summarises my contributions and the main takeaways of the thesis with respect to the research questions that were investigated. The conclusion is followed by a discussion of future research directions in out-of-distribution generalisation in machine learning.

1.4 Publications

The research efforts I carried out during my PhD have led to the publications and workshop presentations listed below in chronological order.

1. Antoniou, A., **Słowik, A.**, Crowley, E. J., and Storkey, A. J. (2019) **Dilated DenseNets for Relational Reasoning** [6]. Oral presentation. I also presented the same talk at the Artificial Intelligence Research Group Talks (Computer Laboratory) series.

2. **Słowik, A.**, Mangla, C., Jamnik, M., Holden, S. B., Paulson, L. C. (2019) **Bayesian**

Optimisation for Heuristic Configuration in Automated Theorem Proving [257]. Oral presentation.

3. **Słowik, A.**, Mangla, C., Jamnik, M., Holden, S. B., Paulson, L. C. (2020) **Bayesian Optimisation for Premise Selection in Automated Theorem Proving (Student Abstract)** [258]. Poster presentation.

4. Danel T., Spurek P., Tabor J., Śmieja M., Struski Ł., **Słowik A.**, Maziarka Ł. (2020) **Spatial Graph Convolutional Networks** [265, 58].

5. **Słowik, A.***, Gupta, A.*, Hamilton, W. L., Jamnik, M., Holden, S. B. (2020) **Towards Graph Representation Learning in Emergent Communication** [255]. Poster presentation.

6. Guo, S.*, **Słowik, A.***, Ren, Y.*, Mathewson, K. (2020) **Inductive Bias and Language Expressivity in Emergent Communication** [95]. Poster presentation.

7. Aubin, B., **Słowik, A.**, Arjovsky, M., Bottou, L., Lopez-Paz, D. (2020) **Linear unit-tests for invariance discovery** [10]. Oral presentation.

8. **Słowik, A.***, Gupta, A.*, Hamilton, W. L., Jamnik, M., Holden, S. B., Pal, C. (2021) **Structural Inductive Biases in Emergent Communication** [97, 256]. Oral presentation.

9. Lamb, A., Goyal, A., **Słowik, A.**, Beaudoin, P., Mozer, M., Bengio, Y. (2021) **Neural Function Modules with Sparse Arguments: A Dynamic Approach to Integrating Information across Layers** [160]. Oral presentation.[1]

10. **Słowik, A.**, Bottou, L. (2021) **Algorithmic Bias and Data Bias: Understanding the Relation between Distributionally Robust Optimization and Data Curation** [252]. Contributed talk.

11. **Słowik, A.**, Bottou L. (2022) **On Distributionally Robust Optimization and Data Rebalancing** [253]. Oral presentation.

12. **Słowik, A.**, Bottou, L., Holden, S. B., Jamnik, M. (2022) **On the Relation between Distributionally Robust Optimization and Data Curation (Student Abstract)** [260]. Oral presentation. **AAAI 2022 Best Student Abstract Honor-**

[1] Alex Lamb is the lead author of this paper. I contributed with research, implementation and experimental results on NFM in the context of visual-question answering and OOD generalisation, as well as by producing results that show that the benefits of using NFM extend beyond the benefit of additional model capacity for the AISTATS rebuttal. In Chapter 4, I expand substantially on my contribution to this paper and I include only the results of my individual follow-up research and experiments.

able Mention (conference acceptance rate: 15%, with 19 out of 111 accepted abstracts selected for an oral presentation, and 3 out of 19 receiving a mention).

All figures in this thesis are my original work unless otherwise credited in the captions.

The research carried out during my PhD led to the Young AI Researcher 2022 Award (https://www.cst.cam.ac.uk/news/award-students-work-addressing-bias-machine-learning) and Myson College Exhibition for Personal Achievement 2022 (awarded by Lucy Cavendish College).

I was also awarded the departmental Wiseman Prize (https://www.cst.cam.ac.uk/wiseman-prize) for my teaching, mentoring and community-building activities.

Chapter 2

Background

When I was a student, old professors called photocopiers a xerox machine. Today, we call `__call__` *the forward pass.*
–Ferenc Huszár (2022)

This chapter describes the notation used throughout the thesis (Section 2.1), the research methodology and metrics used throughout the contribution chapters (Section 2.2), an overview of concepts essential to the work undertaken in the thesis but not crucial in all chapters (Section 2.3.1), and a summary of an extensive literature review on out-of-distribution generalisation in machine learning (Section 2.4).

The research literature review for this dissertation is further organised into three sections. Sections 2.4.1 and 2.4.2 contain a review that is relevant to this dissertation as a single body of work, and which helps in placing this dissertation into a wider context of research on out-of-distribution generalisation in machine learning. These sections define the types of distribution shift studied in machine learning and link them to the new contributions presented in the core research chapters. This is a result of a bird's-eye, domain-agnostic view of the field of out-of-distribution generalisation in machine learning. Section 2.4.3 then presents a detailed, low-level summary of key papers relevant to each of the three main settings covered in this dissertation: learning from multiple distributions, image classification and multi-agent communication. Despite all these settings living under the umbrella of modern machine learning, each of them treats the topic of out-of-distribution generalisation in a slightly different way, and each of them comes with its own domain-specific characteristics (including the commonly used terminology).

2.1 Notation

In this section, I introduce consistent notation that is respected throughout the thesis. All of the descriptions are given with the assumption that the reader is familiar with the underlying concepts.

- Vectors and matrices are written *in boldface* when only one letter is used to represent them. Matrices are typically capitalised ($\boldsymbol{A}$, $\boldsymbol{M}$, $\boldsymbol{\Sigma}$), and vectors are written in lowercase ($\boldsymbol{u}$, $\boldsymbol{x}$, $\boldsymbol{\mu}$). The components of a vector (or matrix) are written using the same letter as the vector (or matrix), with added subscript(s):

$$\boldsymbol{v} = \begin{bmatrix} v_1 \\ \vdots \\ v_d \end{bmatrix} \qquad \boldsymbol{A} = \begin{bmatrix} A_{11} & \cdots & A_{1m} \\ \vdots & & \vdots \\ A_{n1} & \cdots & A_{nm} \end{bmatrix}$$

- Sets of numbers and vector spaces are written in *blackboard bold* – for example, $\mathbb{R}$ for the set of real numbers.

- The following are equivalent ways of stating the components of a vector $\boldsymbol{v} \in \mathbb{R}^d$:

$$\boldsymbol{v} = (v_1, \ldots, v_d) \qquad \boldsymbol{v} = \begin{bmatrix} v_1 & \cdots & v_d \end{bmatrix}^\mathsf{T} = \begin{bmatrix} v_1 \\ \vdots \\ v_d \end{bmatrix}$$

- If $\boldsymbol{a} \in \mathbb{R}^m$ and $\boldsymbol{b} \in \mathbb{R}^n$, then $\boldsymbol{c} = (\boldsymbol{a}, \boldsymbol{b})$ is the result of *concatenating* $\boldsymbol{a}$ and $\boldsymbol{b}$, which formally means $\boldsymbol{c} \in \mathbb{R}^{m+n}$ and:

$$\forall i \in \{1, \ldots, m+n\}. \qquad c_i = \begin{cases} a_i, & i \leq m \\ b_{i-m}, & i > m \end{cases}$$

- The univariate normal distribution with mean μ and variance σ^2 is written as $\mathcal{N}(\mu, \sigma^2)$. A d-dimensional normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ is written as $\mathcal{N}_d(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

  The calligraphic font is also used to represent other probability distributions: uniform ($\mathcal{U}(a, b)$), categorical ($\mathcal{C}(p_1, \ldots, p_n)$), Gumbel ($\mathcal{G}(\mu, \beta)$).

  Where appropriate, an upright serif font is used instead (for example, Geometric(p)).

- The notation $\mathbb{E}_{X \sim P}[f(X)]$ is the expectation of the quantity $f(X)$, where X is a random variable with probability distribution P.

- For a probability distribution P, p is either a probability mass function or probability density function (depending on whether P is discrete or continuous).

- Model parameters are represented by the letter $\boldsymbol{\Theta}$. When considering neural networks, $\boldsymbol{w}$ is used to represent the network weights (including biases). When discussing subsets/projections of $\boldsymbol{\Theta}$, we sometimes write $\boldsymbol{\theta}$ or $\boldsymbol{\omega}$.

- Functions that depend on data points, but also model parameters – for example, model errors – separate the two using semicolons: $\mathcal{L}(\boldsymbol{X}, \boldsymbol{y}; \boldsymbol{\Theta})$.

- Data point inputs are usually represented as matrices $\boldsymbol{X}$, with individual data points $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, …, $\boldsymbol{x}_n$. These are rows or columns of $\boldsymbol{X}$ depending on the setting. The j-th component of $\boldsymbol{x}_i$ is written $\boldsymbol{x}_i^{(j)}$ unless stated otherwise.

  Outputs are typically one-dimensional vectors $\boldsymbol{y} = (y_1, \ldots, y_n)$.

- Commonly used functions operating on vectors or neuron outputs are written in an upright, sans-serif font when they appear in equations: $\mathsf{Softmax}(\boldsymbol{v})$, $\mathsf{ReLU}(\boldsymbol{v})$. They are not written in this font when they appear in prose ('softmax', 'ReLU').

- All vector norms, represented by $\|\boldsymbol{x}\|$, refer to the Euclidean norm.

- In data generation descriptions, the notation $\boldsymbol{x} \sim P$ means that a vector $\boldsymbol{x}$ is randomly generated according to distribution P. For example, $\boldsymbol{x} \sim \mathcal{N}_3(0, 1)$.

  The notation $\boldsymbol{x} \leftarrow E$ means that $\boldsymbol{x}$ is assigned a value deterministically by evaluating the expression E. For example, $\boldsymbol{x} \leftarrow 2\boldsymbol{y} - \boldsymbol{z}$.

2.2 Methodology

This thesis contains both theoretical (Chapter 3) and experimental results (Chapter 4 and Chapter 5) on the topic of out-of-distribution generalisation in machine learning. Throughout the experiments, I aim to vary one 'parameter' at a time in order to draw conclusions about the impact of this parameter (for example, DenseNet is compared with the same model augmented with dilated convolutions in Chapter 4). Where existing (published) results from a comparable setting are available, they are included for comparison.

Training/validation/test splits The standard machine learning practice is to split the available data into random training, validation and test subsets. A training subset (the largest one, for instance, 80% of all the available data) is used to fit the parameters of a model (the process referred to as learning or training and described briefly in Section A.2). A validation set is used to evaluate the model throughout training (for example, after

every epoch in the case of a neural network), and to compare different model parameters before arriving at the final model. A test set is used to evaluate the model after training is completed.

As further discussed in Chapter 3, this way of partitioning the dataset relies on the assumption that data is identically and independently distributed. The samples are shuffled in order to bring the real data closer to the i.i.d assumption (even though in reality there might be multiple data sources that are systematically different). Training, validation and test splits are assumed to come from the same distribution as they are random partitions of the shuffled dataset.

In this dissertation, the main focus is the *out-of-distribution* (OOD) test performance rather than the standard in-distribution test performance. The OOD test samples are created in several purposeful ways to evaluate the robustness with respect to common distribution shifts. In the OOD tests, the test samples are systematically different from the training samples in a controlled way (for example, in Linear unit tests in Chapter 3, training samples contain spurious correlations that are removed from the test samples in the OOD evaluation).

Metrics *Accuracy* on the out-of-distribution subsets is the main quantitative metric throughout this dissertation. There are also domain-specific metrics where appropriate (for example, topographic similarity in Chapter 5). Randomly sampled qualitative examples are shown in each chapter for illustration.

Early stopping During training, the performance of the model on training data improves. On the test set, this trend is observed up to a point, after which the model performance on test data decreases. This is referred to as *overfitting* (Figure 2.1) and should be avoided in deployed machine learning models. A simple way of avoiding overfitting is by stopping training early, before training performance converges and when test performance is at the highest point. However, the test set must not be seen during training. The point at which training stops is determined by performance on the validation set: the validation performance of a model is used as a proxy for test performance, thus training stops once validation performance starts decreasing.

2.3 Key techniques

In this section, I describe several algorithms and mathematical concepts that are referenced throughout the thesis. These are specialised topics that are not found in standard machine

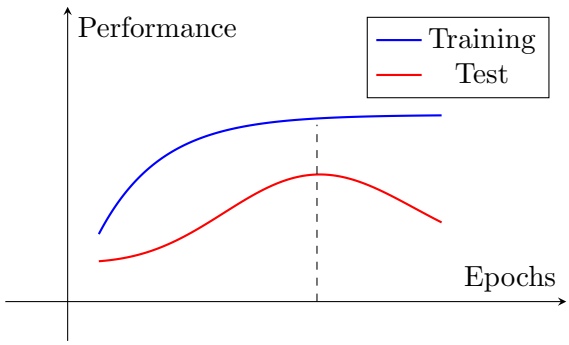


Figure 2.1: An illustration of overfitting (for an abstract notion of performance). The dashed line represents the point at which training should stop. In early stopping, validation performance is used as a proxy for test performance.

learning textbooks as of writing this thesis. More fundamental ideas that are also referenced throughout the thesis but are well-known are included in Appendix A.1.

This section is divided into two parts. Section 2.3.1 covers several operations and methods: Softmax,[1] Gumbel-Softmax trick, and Gaussian Blobs. In contrast to these concepts, *attention* requires more motivation and description, given in Section 2.3.2.

2.3.1 Simple concepts

2.3.1.1 Softmax

In many machine learning tasks, it is required to produce a discrete probability distribution from a real vector. This is especially common when neural networks are used for classification tasks: the values output by the n output neurons need to be converted into probabilities that the input belong to the corresponding class, out of n in total. To this end, we use the $\mathsf{Softmax}$ function. When applied to a vector $\boldsymbol{v} = (v_1, \ldots, v_n) \in \mathbb{R}^n$, it produces another vector $\mathsf{Softmax}(\boldsymbol{v}) = \boldsymbol{p} = (p_1, \ldots, p_n) \in \mathbb{R}^n$ using the following formula (for all i, $1 \leq i \leq n$):

$$p_i = \frac{e^{\lambda v_j}}{\sum_{j=1}^{n} e^{\lambda v_j}},$$

for some choice of $\lambda \geq 0$.

It is not difficult to verify that the components of $\boldsymbol{p}$ give the probability mass function of a categorical distribution $\mathcal{C}(p_1, \ldots, p_n)$: the value of $e^{\lambda v_j}$ is positive for all values of λ and v_j, so the values of all p_i are non-negative, less than 1, and they sum to 1.

[1] Softmax is important in every classification task, including those presented in Chapters 3, 4 and 5, as well as in all "fits" to probability distributions discussed and proposed in this thesis. These include the Gumbel-softmax trick, attention mechanisms, and Neural Function Modules.

The role of λ is in determining how differences in the values of v_i translate into differences in probabilities p_i. When $\lambda = 0$, all values of p_i equal $1/n$. As $\lambda \to +\infty$, all values of p_i approach to 0 except for one, corresponding to the largest value of v_i.[2]

All problems presented in Chapter 4 and Chapter 5 are variants of multi-class classification. The Softmax function is then used in all the methods presented in these chapters. The Softmax function is also necessary for describing how the agents are trained in the referential games presented in Chapter 5 (Section 5.1.2.1 on game dynamics).

2.3.1.2 Gumbel-softmax relaxation

Consider a categorical distribution $\mathcal{C}(p_1, \ldots, p_n)$. Rather than having one-hot outputs $\boldsymbol{w}$ of this distribution, obtain n samples $\{u_k\}_{k=1}^n$ from the uniformly distributed random variable $U \sim \mathcal{U}(0, 1)$ and transform each sample to obtain $g_k = -\log(-\log u_k)$. These g_k follow the *Gumbel distribution* $\mathcal{G}(0, 1)$. Use these samples to obtain a continuous relaxation $\mathsf{GumbelSoftmax}(p_1, \ldots, p_n) = \tilde{\boldsymbol{w}} = (\tilde{w}_1, \ldots, \tilde{w}_n)$ of the one-hot vector, defined as follows:

$$\tilde{w}_k = \frac{e^{(\log p_k + g_k)/\tau}}{\sum_{i=1}^{n} e^{(\log p_i + g_i)/\tau}}$$

where τ is the *temperature* of the trick. The role of the temperature is to control the accuracy of the approximation of $\arg\max$ (which outputs a one-hot vector) with Softmax: when $\tau \to 0$, the samples from the Gumbel-softmax distribution become one-hot vectors; when $\tau \to +\infty$, all components becomes $1/n$.

Gumbel-softmax relaxation (sometimes referred to as the *Gumbel-softmax trick*) was first introduced independently by Jang et al. and Maddison et al. [126, 183] in order to turn discrete inputs – one-hot encoding of category membership based on applying the $\arg\max$ function – into continuous and easily differentiable inputs. In this thesis, it is used for the same reason – in the context of multi-agent communication.

2.3.1.3 Gaussian Blobs

In both synthetic and real-world data, noise follows a Gaussian distribution. A model that fits and interprets the data well is expected to be robust to the noise. Finding a pattern in such high-entropy noise is indicative of overfitting, and it is likely that the model fails to generalise.

Communicating normally-distributed noise is used in multi-agent communication in order to provide a sanity check of whether agents are learning to successfully interpret messages

[2] modulo the pathological cases where there are multiple maxima among the components of $\boldsymbol{v}$

or if they are erroneously learning the noise [163, 29]. This is the context in which *Gaussian Blobs* are used in this thesis.

2.3.2 Attention

The idea behind the design of many neural networks is to mimic the cognitive processes with which humans solve tasks. A common mechanism in reasoning is to use important historical and contextual information, requiring mental focus. Humans 'pay attention' to the relevant parts and roles of an input in order to solve a particular problem. Based on this idea, the powerful concept of *attention* in a neural network has been developed.

The need for attention is best seen in the setting of natural language processing, where humans infer the meaning and roles of words based on relevant parts of the rest of the sentence. Consider the following example of machine translation, from English to Polish:

$$\text{I like strawberries.} \quad \rightsquigarrow \quad \text{Lubię truskawki.}$$

Translating each part of the sentence and the sentence as a whole relies mostly on translating the main concept ('strawberries'), but also its relations to other words. The word 'strawberries' serves the role of the object in this sentence and its translation needs to be in the accusative case (as seen from 'lubię'), so we use the form 'truskawki', as opposed to, for instance, 'truskawkom' or 'truskawkami'. On the other hand, the word 'I' does not affect the translation of the word 'strawberries' and attention should not be paid to it while translating.

In short, the effect of using attention is to enhance some parts of the input data in computing the output while diminishing the others.

I first give an outline of attention mechanisms (Section 2.3.2.1), before focusing on the attention mechanism of the Transformer architecture – multi-headed scaled dot-product attention – used by Neural Function Modules and the most popular attention mechanism as of time of writing this paragraph[3] (Section 2.3.2.2).

2.3.2.1 Brief overview of attention mechanisms

The basic idea of attention can arguably be seen in the design of recurrent neural networks (RNNs) [236]. Recurrent neural networks typically operate on sequential inputs such as sentences – seen as sequences of words – and their operation considers each symbol in

[3]The most talked-about topic in machine learning in 2023, attracting unprecedented levels of public attention, are large language models such as GPT-4. This has resulted in widespread interest in the Transformer architecture.

the sequence in turn. They maintain a hidden state which gets updated as symbols are processed: the new hidden state $\boldsymbol{h}_{t+1}$ depends on the previous hidden state $\boldsymbol{h}_t$ and the next symbol, encoded as $\boldsymbol{x}_{t+1}$: $\boldsymbol{h}_{t+1} = f(\boldsymbol{x}_{t+1}, \boldsymbol{h}_t)$ for some function f. The main disadvantage of RNNs is that too much emphasis is placed on words being close to one another and long-range dependencies can be missed. Early references to mimicking attention (for example, Align & Translate by Bahdanau et al. [13]) are based on RNNs and fix this problem. In the setting of machine translation, attention establishes connections between parts of a sentence even if the word order is not the same in two languages. In short, a model is capable of learning complex relationships between the input and the output.

Some attention mechanisms go beyond the basic sequence-to-sequence (Seq2Seq) problems, as in machine translation. Attention can be applied in the setting of visual reasoning. Xu et al. [295] use attention mechanisms in the context of automatic image captioning. Their method is based on feature extraction in convolutional neural networks, after which the caption (word sequence) is constructed.

Using RNNs and CNNs as the basis for attention mechanisms has a significant flaw: recurrences and convolutions are non-linear operations (cannot be expressed as matrix multiplication) that are difficult to parallelise. Early neural networks with attention mechanisms were mostly based on an encoder-decoder architectures based on these operations. The Transformer model [280] is an influential architecture which dispensed with these operations. The Transformer is based only on attention mechanisms ('Attention is all you need', as the title of the seminal paper goes) and is thus easily parallelisable on a GPU.

2.3.2.2 Attention mechanism of Transformer

I describe the Transformer attention mechanism both to provide an example to the previous discussion and to provide background for Chapter 4, where it plays a key role in the design of Neural Function Modules.

The attention function is based on the analogy with querying the map (sometimes also called a table) data structure. A map is a set of key-value pairs (a value associated with each key), and a query might request retrieval of a value associated with a particular key. In the context of attention, the query, keys and values are input vectors, and the result is the output vector. The result is a weighted sum of the values, where the weight associated with each value is based on the compatibility of the query with the corresponding key.

Scaled dot-product attention Suppose each key $\boldsymbol{k}$ is a d_k-dimensional real vector and that values $\boldsymbol{v}$ are d_v-dimensional real vectors ($\boldsymbol{k} \in \mathbb{R}^{d_k}$, $\boldsymbol{v} \in \mathbb{R}^{d_v}$). The compatibility of a query $\boldsymbol{q} \in \mathbb{R}^{d_k}$ with a key $\boldsymbol{k} \in \mathbb{R}^{d_k}$ is their dot product $\boldsymbol{q}^\mathsf{T}\boldsymbol{k}$. The whole map and a

set of queries are considered in parallel: the n keys and values are packed into matrices $\boldsymbol{K} \in \mathbb{R}^{n \times d_k}$ and $\boldsymbol{V} \in \mathbb{R}^{n \times d_v}$, and a set of m queries is also packed into a matrix $\boldsymbol{Q} \in \mathbb{R}^{m \times d_k}$. The attention function is thus defined as follows:

$$\mathsf{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \mathsf{Softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\mathsf{T}}{\sqrt{d_k}}\right)\boldsymbol{V}.$$

The intuition is that $\boldsymbol{Q}\boldsymbol{K}^\mathsf{T}$ determines the significance of each key-value pair for all queries. The $\mathsf{Softmax}$ function is here assumed to operate on each row separately and its goal is to smooth the coefficients associated with values and make them sum up to 1. The scaling factor $\sqrt{d_k}$ is used in practice in order to prevent the softmax function from being applied to regions with extremely small gradients when d_k is large (which tends to worsen performance). Finally, the values are selected and their weighted sum is the result for each query.

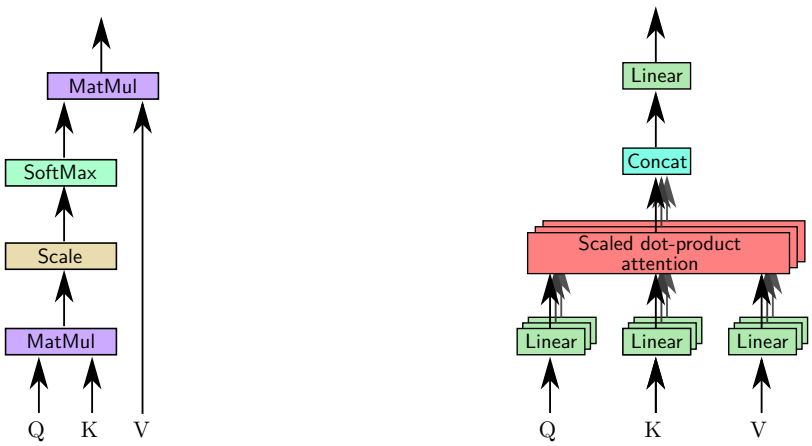
Multi-head attention The Transformer architecture introduced the idea of *multi-head attention*. Instead of performing a single attention function with all keys, values and queries being d_{model}-dimensional vectors (where d_{model} is the dimension of the input embedding), each query, key and value is projected to smaller subspaces h times. These projections are represented by matrices $\boldsymbol{W}_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $\boldsymbol{W}_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$ and $\boldsymbol{W}_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$ with learned parameters, where $i \in \{1, \ldots, h\}$. Attention is then separately performed on projected versions by the multiple *heads*, the output vectors are all concatenated and then finally projected to give the final values using a matrix $\boldsymbol{W}^O \in \mathbb{R}^{(hd_v) \times d_{\text{model}}}$:

$$\begin{aligned} \mathsf{MultiHead}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) &= (head_1, \ldots, head_h)\boldsymbol{W}^O \\ \text{where } head_i &= \mathsf{Attention}(\boldsymbol{Q}\boldsymbol{W}_i^Q, \boldsymbol{K}\boldsymbol{W}_i^K, \boldsymbol{V}\boldsymbol{W}_i^V). \end{aligned}$$

2.4 Out-of-distribution generalisation

Historically, research on neural networks was focused on improving the in-distribution rather than out-of-distribution performance. Recent widespread adoption of machine learning algorithms, encountered every day by a variety of users, has shown the limitations of this approach.

The first part of this section (Section 2.4.1) describes the common changes in the distribution between the training and the test scenarios (*distribution shift*) and their relation to the contributions presented in this dissertation. The second part (Section 2.4.2) discusses *compositional generalisation*, the ability to combine known components in order to systematically generalise when a distribution shift occurs, and gives the context in which this is studied in the thesis. The third part (Section 2.4.3) systematises existing



(a) Scaled dot-product attention (b) Multi-head attention

Figure 2.2: A diagrammatic representation of the attention mechanism used in the Transformer architecture.

research on out-of-distribution generalisation in a way that corresponds to the three core contribution chapters: existing work on algorithmic and data-driven approaches to OOD generalisation (expanded in Chapter 3), existing work on OOD generalisation in image classification (expanded in Chapter 4) and finally, prior work on OOD generalisation in multi-agent communication (expanded in Chapter 5).

2.4.1 Types of distribution shift

Distribution shift refers to the change in the input data to a trained model, which causes it to become less accurate when evaluated on the test set.

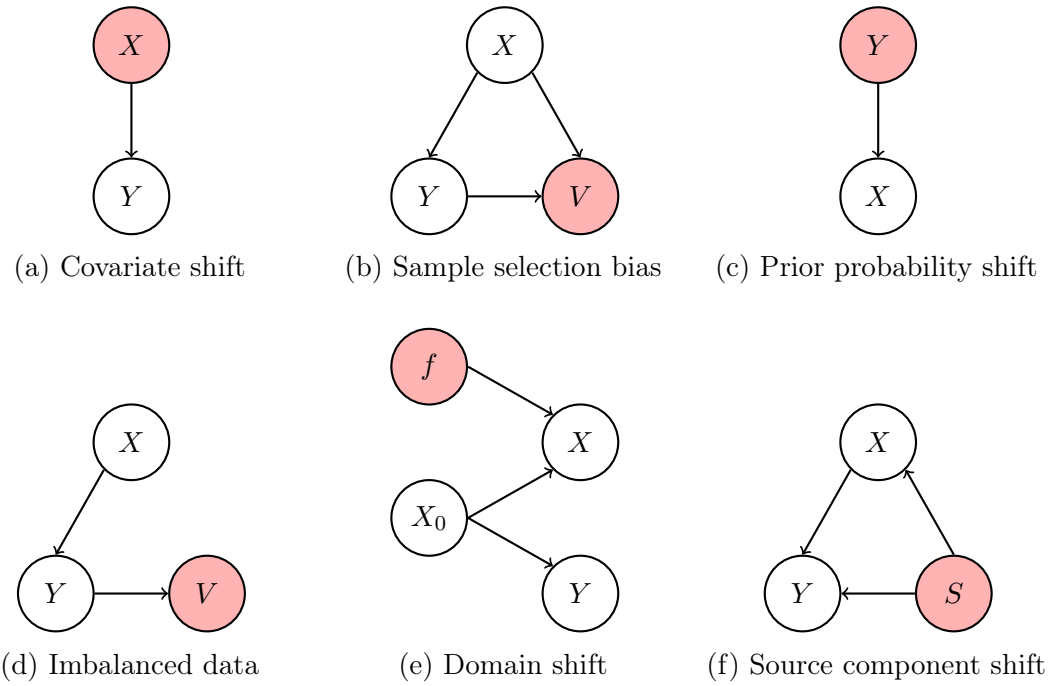
Assume input data X and the corresponding outputs Y. In supervised learning, the training data can be viewed as a set of samples from the joint distribution $\mathbb{P}(X, Y)$. The purpose of a standard supervised model is to represent $\mathbb{P}(Y|X)$, that is, the conditional probability of an output given an input. The joint distribution $\mathbb{P}(X, Y)$ can be decomposed in two ways:

$$\begin{aligned} \mathbb{P}(X, Y) &= \mathbb{P}(Y|X)\mathbb{P}(X) \\ \mathbb{P}(X, Y) &= \mathbb{P}(X|Y)\mathbb{P}(Y) \end{aligned}$$

where $\mathbb{P}(X)$ denotes the probability distribution of the input and $\mathbb{P}(Y)$ denotes the probability distribution of the output.[4]

There is an infinite number of possible changes in the data distribution. Based on the previously defined concepts and the classification of distribution shifts from the seminal

[4]Distribution' refers to the probability mass function when referring to discrete random variables and the probability density function when referring to continuous random variables. In classification problems Y is discrete, while in regression Y is continuous. X is usually continuous in both cases.



(a) Covariate shift (b) Sample selection bias (c) Prior probability shift

(d) Imbalanced data (e) Domain shift (f) Source component shift

Figure 2.3: A diagrammatic representation of the six common types of distribution shift. In each example, the arrows represent the invariant conditional probabilities. The shaded circles represent the data that changes between the training and test set: for instance, in the case of a covariate shift $\mathbb{P}(X)$ changes and the conditional probabilities $\mathbb{P}(Y|X)$ remain the same. Note that more variables than the shaded one can change, but the shaded variable is the defining shift that drives the change in each type of distribution shift. V is a selection variable, which indicates if an example is included in the respective dataset or not. f is a representation function, for instance, whether an image is framed or not. S refers to source proportions which can vary between training and testing.

book 'Dataset shift in Machine Learning' [220], Figure 2.3 shows the common differences between the training and the test distribution.

The following subsections discuss the various types of distribution shifts (illustrated in Figure 2.3): covariate shift (Section 2.4.1.1), sample selection bias (Section 2.4.1.2), prior probability shift (Section 2.4.1.3), imbalanced data (Section 2.4.1.4), domain shift (Section 2.4.1.5) and source component shift (Section 2.4.1.6). Finally, I include an aside on adversarial machine learning as an example of robustness to distribution shifts (Section 2.4.1.7).

2.4.1.1 Covariate shift

Covariate shift (Figure 2.3a) means that $\mathbb{P}(X)$ changes and the conditional probabilities $\mathbb{P}(Y|X)$ remain the same. Covariate shift is one of the most widely studied distribution shifts in machine learning. It commonly occurs as a result of a *selection bias* (Figure 2.3b),

or missing data. For instance, consider the task of detecting breast cancer[5]. The risk of breast cancer is higher for women over the age of 40. Consider a simple model that is meant to predict the risk based on the age of the patient. If there are more women over the age of 40 in the training data than in the inference data (for instance, because the women over the age of 40 are encouraged to get tested and they provide training samples, but random people can sign up to be tested using the trained model), $\mathbb{P}(X)$ changes at the inference stage. However, for an example with a given age, such as above 40, the probability that this example has breast cancer $\mathbb{P}(Y|X)$ is constant. This is an example of a covariate shift. Covariate shift notoriously occurs in practical applications when a model trained on the past data (for example, the records from the past 5 years) is meant to be used in the future (for example, a year from now). Even if $\mathbb{P}(Y|X)$ is assumed to stay the same, there will often be differences in the future samples in comparison to the samples from the past due to various changes that occur in the meantime.

Connection to the thesis Covariate shift is the canonical instance of distribution shift (at times used as a synonym of 'distribution shift') and it is relevant to all of the contribution chapters in this dissertation. Chapter 3 presents the approaches to mitigate such shift based on learning an *invariant representation* and ignoring *spurious features* (Section 3.3), as well as the approaches based on optimising for the most challenging data distribution that can be modelled at the training stage (Section 3.4). Chapter 4 shows results of the experiments using a wide array of the instances of a covariate shift in image data (for example, changes in the colour or object count between the training and the test images). Chapter 5 shows experiments that use several examples of covariate shift in the domain of graphs and realistic images in a multi-agent setting.

2.4.1.2 Selection bias

Selection bias (Figure 2.3b) refers to the situation when the training and test distributions differ as a result of a sample rejection process [109]. For instance, public opinion polling conducted via landline phones can lead to an under-representation of the views of the young people who do not have landline phones. Another example that links selection bias to covariate shift: women over the age of 40 are reminded to have breast scans, which might lead to their over-representation in the training dataset in comparison to the entire population of women.

Connection to the thesis Chapter 3 mentions the problem of selection bias in the context of the prevalence of the 'minority' and the 'majority' groups that are often

[5]An example from the *Machine Learning Systems Design* course at Stanford University (CS 329S by Chip Huyen, 2022).

combined into a single training dataset. Due to selection bias that favours the majority group, the samples of certain properties might be more prevalent in the training data, which means that the solution based on the average error on the entire training dataset might be similar to a solution that represents only the majority group (Figure 3.1 in Chapter 3). Survival bias is a common intuitive instance of selection bias – for instance, the approach of using only the records of successful start-ups as an indicator of economy without including the failed companies that fell out of view. Section 3.4 presents my contributions to the research on addressing the problems often caused by selection bias, such as the issue of over- and under-representation.

2.4.1.3 Prior probability shift

Prior probability shift (also referred to as label shift; Figure 2.3c) refers to the situation when $\mathbb{P}(Y)$ changes and $\mathbb{P}(X \mid Y)$ remains unchanged. In other words, the output distribution changes but for a given output, the input distribution stays the same. In the breast cancer example, if there are more women over 40 in the training data than in the test data, the percentage of POSITIVE labels might be higher during training than at test time. Another example of the prior probability shift occurs in various ML-based failure detection systems, when a change in the maintenance policy results in fewer failures and a new distribution of the positive and negative examples [148]. Covariate shift and prior probability shift often occur simultanously – if one occurs without the other, it means that at least one of the conditional probabilities changes (as seen from the joint probability decomposition).

Connection to the thesis Both covariate shift and prior probability shift occur in the *compositional generalisation/systematic generalisation* (Section 2.4.2) tasks studied in Chapter 4 and Chapter 5: the distribution of features $\mathbb{P}(X)$ shifts due to the test examples that contain new feature combinations (for instance, an algorithm trained on the examples of a 'red square' and 'blue circle' is tested on the 'red circle' example), and the label distribution $\mathbb{P}(Y)$ changes since the label is assigned based on the features that are recomposed (for example, the figures are assigned the labels corresponding to their shapes and colours). Section 4.2 presents new results in compositional generalisation using image data, and Section 5.1.4 presents new results in compositional generalisation in the context of multi-agent communication.

2.4.1.4 Imbalanced data

Imbalanced data (Figure 2.3d) refers to a multi-class machine learning problem where one or more classes are very rare compared to the other classes. It happens by design in systems that predict very rare events. In the case of imbalanced data, the shift depends

on the labels alone, whereas in the case of selection bias, the shift depends both on the features and the labels (Figure 2.3b and Figure 2.3d). Since it might be infeasible to collect more samples from the rare class, the samples from the more prevalent class are sometimes discarded at the training stage to create a more balanced dataset. However, this leads to the distribution shift between the training stage that uses an artificially balanced dataset and the evaluation stage when the model is deployed on the real, imbalanced population.

Connection to the thesis Imbalanced datasets are not the focus of this dissertation; however, there is ample literature on addressing this problem [218, 129].

2.4.1.5 Domain shift

Domain shift (Figure 2.3e) occurs when the representation of the problem changes. Consider a latent variable X_0 referring to the unchanging meaning of the input data. The output Y is dependent solely on the underlying latent variable X_0. However, X_0 is not explicitly observed, as the input X is the result of a mapping from X_0: $X = f(X_0)$ for some function f. Domain shift refers to a different f in the test scenario compared to the training stage. Using an example from Chapter 3, X_0 might refer to information that uniquely identifies the type of an animal in an image classification system, Y is the label corresponding to the animal type, and X is the image data. However, f might change if it represents a different environment (meadow or beach, day or night). The feature distribution $\mathbb{P}(X)$ changes if the representation mapping f changes. Unlike in the case of covariate shift, there is no requirement that $\mathbb{P}(Y \mid X)$ remains the same. The conditional probability $\mathbb{P}(Y \mid X_0)$ stays the same, but $\mathbb{P}(Y \mid X)$ changes if the transformation f changes.

Connection to the thesis Linear unit tests (Section 3.3 in Chapter 3) are all examples of domain shift, where the domain change from training to testing is manifested in shuffling spurious features to remove the spurious links. The test domain includes only the invariant links to test whether the algorithms are able to learn them from the training domain, while ignoring spurious links.

2.4.1.6 Source component shift

Source component shift (Figure 2.3f) is a type of distribution shift that directly relates to the problem of data coming from multiple sources rather than being independently and identically distributed. Each data source can have its own characteristics, and the proportions of the samples from those sources can vary between the training and test stages. The data source (also referred to as the 'environment' [7]) is a confounding factor that affects the X and Y values.

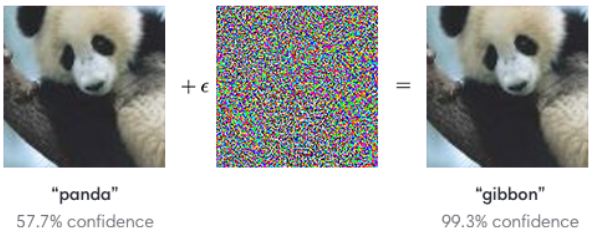


Figure 2.4: An example that illustrates vulnerability of neural networks in the face of distribution shifts (even those undetectable by humans).
The authors [88] generate this example using a powerful GoogLeNet architecture [273], ImageNet dataset [65], the weight $\epsilon = 0.007$ and a noise perturbation. The perturbed input image results in the model outputting an incorrect answer ('gibbon') with high confidence.
Vulnerability to insignificant input perturbations further motivates the research into understanding distribution shifts in machine learning.

Connection to the thesis The entire Chapter 3 assumes the setting of data coming from multiple sources. Section 3.3 presents a set of tasks that evaluate whether algorithms can learn invariant features across multiple data sources. Section 3.4 shows new results in the field of research focused on minimising the error on the most difficult data source.

These types of distribution shifts are closely related, and a model can suffer from multiple types of shifts at the same time. In order to generalise in the presence of a distribution shift, the training and test distributions have to be related in an exploitable way. A practical example of such a relation between the training distribution and the test distribution is exploited in the case of compositional generalisation [133, 4, 208, 173, 24, 201, 291, 9, 238].

2.4.1.7 Digression: adversarial machine learning

The fragility of neural networks optimised for the highest in-distribution accuracy came to light with the work on *adversarial robustness* showing that neural networks cannot generalise even to visually indistinguishable changes in image data [272]. There is a close interplay between adversarial robustness and out-of-distribution generalisation, with methods such as Distributionally Robust Optimisation (Chapter 3) being adapted to tackle both problems [7, 267]. Figure 2.4 shows a motivating example.

2.4.2 Compositional generalisation

The idea of compositional generalisation in machine learning is inspired *by the principle of compositionality* from logic and semantics: *The meaning of a complex expression is determined by the meanings of its parts and the way they are syntactically combined* [212]. The principle appears in formal reasoning about both programming and natural languages,

in which an interpretation function $[\![\cdot]\!]$ maps expressions to their semantics and its definition follows the syntax. For example, the meaning of 'two plus three' depends on the meanings of 'two', 'plus' and 'three', so its translation into the semantic domain (the domain of numbers) is as follows:

$$[\![\text{two plus three}]\!] = [\![\text{plus}]\!]([\![\text{two}]\!], [\![\text{three}]\!]) = add(2, 3) = 5.$$

In some cases, the meaning of a complex expression can be predicted in a rigorous way based on the meanings of its constituents (for example, when mathematical functions or operations in a computer program are composed, as is the case in Scott's *denotational* approach to formal semantics of programming languages [245]). In other cases the inference can be noisy and complex, such as in human language (for example, in the sentences 'time flies like an arrow' and 'fruit flies like a banana' the meaning of a single constituent 'flies' changes depending on the context). Rule-based, symbolic approaches (such as pre-2010 research in natural language processing) have struggled with semantic compositionality 'in the wild' due to the number of exceptions and subtleties in interpreting the combinations of real-world data, such as words or visual stimuli.

2.4.2.1 General machine learning perspective

In machine learning, compositional generalisation (also referred to as systematic generalisation or systematic compositionality) refers to the algebraic capacity to understand and produce novel combinations from known components (a definition by Brenden Lake, one of the main contributors to this field)[6]. On the other hand, Dieuwke et al. [122] propose an extended definition of compositional generalisation in the context of natural language processing and machine learning as an umbrella term for five compositional properties: (i) *systematicity*: the ability to recombine known components and rules (used in a synonymous way to 'compositional generalisation' in Chapter 4 and Chapter 5 in this dissertation), (ii) *productivity*: the ability to generalise to longer data samples than those seen in training (most applicable in the context of language data), (iii) *substitutivity*: robustness to synonym substitutions (related to the substitutions of similar colours and shapes in Chapter 4), (iv) *localism*: are local compositions evaluated before the global compositions? and (v) *overgeneralisation*: the ability to extract rules from data and the ability to accommodate exceptions. Figure 2.5 shows a reproduced illustration of this classification of the main types of compositional generalisation.

[6]Source: the talk 'Compositional generalisation beyond the training distribution in minds and machines' at ICLR 2021.

Connection to the thesis Chapter 4 is mostly focused on compositional generalisation in image recognition in the form of systematicity (Figure 2.5 (a)), where the recomposable building blocks are interpretable visual features such as colour and shape. Specifically, systematicity with respect to shape and colour is studied in visual question answering tasks (Section 4.2.1). The experiments on generalisation in font and letter recognition (Section 4.2.3) are also an example of systematicity. The experiments on generalisation with respect to a varying number of objects in a scene (Section 4.2.4) can be seen as an instance of productivity (Figure 2.5 (b)) because models are tested on their ability to extrapolate to a larger number of objects than seen in training. Section 5.1.4 presents a study on systematicity in multi-agent games. The idea of localism (Figure 2.5 (d)) relates to the hierarchical learning of local and global features, which is the motivation behind the idea to probe Dilated DenseNets (Section 4.1.1.2) in their ability to generalise to a systematically different distribution.

2.4.2.2 Neural networks context

Neural networks are a promising alternative to previously popular rule-based systems in terms of compositional generalisation on real-world data due to their data-driven approach to memory and learning. The principle of learning from data might allow neural networks to grasp the complex contextual information that cannot be hard-coded as a set of fixed rules, and this context can be helpful in distinguishing between subtle rules that govern compositional inputs in the domains of vision, language and real-world concepts. For instance, humans know that a 'wine hangover' is a hangover *caused* by wine, a 'college town' is a town that *has* a college, and 'honey bee' *produces* honey.[7] Moreover, we can usually infer the role of a new word in a familiar context, for example, based on the sentence 'we will gfdgsdf like there is no tomorrow' we can infer that 'gfdgsdf' is a verb. Early research on neural networks and generalisation also points to the role of distributed representations in the ability to generalise to a new yet somewhat similar distribution.[8]

Despite the conceptual advantages of neural networks in terms of compositional generalisation, empirical evidence shows that standard neural networks with no additional mechanisms to account for the compositionality of the input data (such as inductive biases [19]) fail even at simple compositional generalisation tasks that humans can do very well. To quote Hudson and Manning, 2018 [120]:

> Most neural networks are essentially very large correlation engines that will

[7] Examples from the a post by Felix Hill: `https://fh295.github.io/noncompositional.html`.

[8] Geoffrey Hinton writes about the advantages of *distributed representations* learned by neural networks (the many-to-many mapping between input concepts and neurons in a network) in terms of generalisation through the ability to exploit the similarity of new information to the previous observations [113].

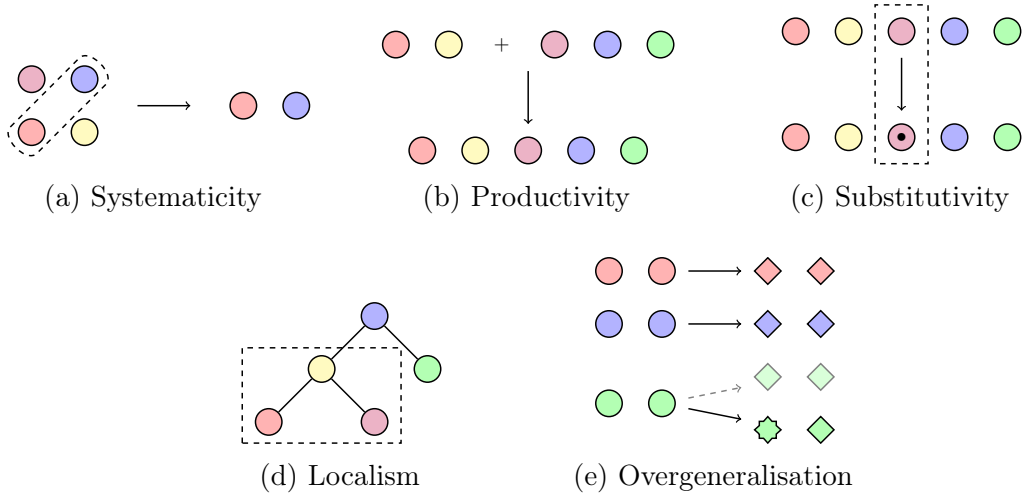


Figure 2.5: Classification of the main instances of compositional generalisation [123].
(a) Systematicity refers to the most common understanding of compositional generalisation, where familiar components are recombined according to a familiar rule. Humans are able to infer meanings for sentences they have never seen before, which means they use the notion of sytematicity rather than memorising each sentence they encounter in a brute force way.
(b) Productivity is a related skill that allows a model to extend beyond the sample length encountered in training. It is best seen in the case of language data ('language makes infinite use of finite means' [54]; under the natural constraints such as human memory or human lifespan).
(c) Substitivity is also closely related to systematicity and language: it extends the idea of systematicity to the ability of replacing words with previously unseen synonyms. Substitivity covers the case when the meaning of a complex whole is not changed as a result of the replacement – in this case, the distribution of features $\mathbb{P}(X)$ might change but the distribution of the descriptive labels that capture the meaning of a sample $\mathbb{P}(Y)$ stays the same.
(d) Localism tests whether a model prioritises global or local compositions. It can be tested by comparing the outputs of a model (for example, a sequence-to-sequence network) for certain stand-alone sentences to the outputs the model assigns to the same sentences, this time presented as part of a longer complex sentence. Finally,
(e) overgeneralisation is tested by using a presumably compositional model on a non-compositional sample (an exception). Most real-world tasks, such as learning a foreign language, require the ability to balance compositional skills (for example, adding the suffix '-ed' to a verb in English tends to result in a past-tense form), and the awereness of exceptions ('broke', 'went').

hone in on any statistical, potentially spurious pattern that allows them to model the observed data more accurately.

Brute memorisation hinders out-of-distribution generalisation, including compositional generalisation.

For instance, Johnson et al. [128] show the results of compositional generalisation experiments designed by swapping the colour palette in a visual question answering task [185] based on images containing 3D shapes. In the training set ('Condition A') all cubes are gray, blue, brown, or yellow and all cylinders are red, green, purple, or cyan. In the OOD test set ('Condition B') colour palettes are swapped. The authors find that the performance of the visual-language model (CNN+LSTM+SA) drops from 85% to 51% due to the change of the colour palettes, which shows that the model fails to generalise to new combinations of familiar shapes and colours.

Connection to the thesis Section 4.2.1 and Section 4.2.2 present the results of two studies based on variants of the visual question answering task introduced by Johnson et al. [128]. In terms of compositional generalisation, Section 4.2.1 studies a more complex task than Johnson et al. [128]: instead of swapping the colour palettes based on the shape, I consider all possible colour-shape combinations and include the results of generalisation to an increasing number of omitted shape-colour combinations up to the maximum number of shape-colour combinations that can be excluded without changing the total number of objects in a scene (Table 4.1). Section 4.2.1 additionally compares the accuracy of answering the relational and non-relational questions in the face of new colour-shape combinations in the test set. Section 4.2.2 is loosely related to Johnson et al. [128]: the dataset contains only two opposite questions ('Are they different?' and 'Are they the same?') paired with images depicting two objects.

In the context of sequence-to-sequence models [270], the results about compositionality and systematicity are mixed. Lake and Baroni [152] show that standard sequence-to-sequence models perform very well in terms of systematicity based on familiar commands (being trained on the examples such as 'jump opposite right after turn opposite right' and 'jump right twice after walk around right thrice', the model was tested on new combinations of familiar subcommands such as 'jump opposite right after walk around right thrice'). However, using the same dataset of commands and actions, the authors found that the models cannot generalise to a longer sequence of actions than seen in training (with the best accuracy of only approx. 20%). Systematicity was also lower when the commands were presented only in their basic form in the training dataset ('jump', 'walk', 'run twice' etc), which suggests that combining basic commands is more challenging than recombining previously observed complex commands.

Connection to the thesis Section 5.1 uses sequence-to-sequence models as baseline methods to explore the hypothesis that graph representation learning induces better results in terms of compositional generalisation. This hypothesis was coined by Battaglia et al. [19] in the context of machine learning in general, and Section 5.1 empirically investigates it in the context of multi-agent communication. Productivity is studied in Section 4.2.4 with the promising results of using Neural Function Modules to improve generalisation to a larger number of objects in the context of multi-object classification (Table 4.4).

A recent paper in the domain of video data by Kipf et al. [140] introduces an extension of Slot Attention [177] to achieve productivity with respect to the video length and generalisation to new objects, new backgrounds, and a combination of both. This work shows the potential of methods based on attention (see Section 2.3.2) and *object-centric learning* [91] in compositional generalisation in challenging setups such as analysing video data. SIMONe [131] is another example of research on compositional generalisation in video. The authors present a Transformer-based model [280] that learns to separate object-specific features (such as size and position) from the features that vary in time as the video progresses.

Connection to the thesis Similarly to the methods mentioned above, Neural Function Modules (Section 4.1.2) use Transformer-based attention (Section 2.3.2.2) as one of the key components. The works such as Slot Attention [177] and SIMONe [131] motivate the use of Neural Function Modules and their variants in the new tasks involving compositional generalisation with respect to interpretable visual features, such as colour and shape (Section 4.2.1 and Section 4.2.2), the font and size of letters (Section 4.2.3) and the number of objects in an image (Section 4.2.4).

To conclude, there is an ongoing debate in the research community on whether neural networks can or cannot generalise in a systematic or compositional way [112], whether such ability is necessary [134] and even what compositional generalisation means [123]. Regardless of the machine learning research, compositionality is studied in semantics, logic and cognitive science, and it is often argued to be one of the most important characteristics of human intelligence [197, 79, 244, 215, 157][9]. Practical advantages of compositional generalisation in the context of machine learning include improved sample efficiency and a higher potential for transfer learning and domain adaptation.

[9]Schulz et al. [244] conducted human experiments based on extrapolation and completion of mathematical functions, and concluded that the human problem-solving strategy is best described as compositional. They also show that people perceive compositional functions as more predictable than their non-compositional counterparts. Piantadosi and Aslin [215] have shown that children as young as 3.5 years old can generalise the idea of function composition with the accuracy above chance, even when they have not been explicitly taught how to compose functions.

2.4.3 OOD generalisation: literature review

So far, this chapter has covered the background that is important for placing the entire dissertation in a wider research context in machine learning. The following part of the chapter systematises more specialised and recent knowledge (prior, concurrent and subsequent papers relevant to the new contributions presented in this dissertation), and breaks it down into sections that correspond to the focus of each of the contribution chapters.

- Chapter 3 contributes to the research on model-agnostic, data-driven and algorithmic approaches to OOD generalisation, in particular to the 'pessimism-based' Distributionally Robust Optimisation approach [21] and to the approach of learning 'invariant features' (where the key previous work is Invariant Risk Minimisation [7]). Section 2.4.3.1 and Section 2.4.3.2 include the part of the literature review focused on Chapter 3.
- Chapter 4 focuses on image recognition, arguably the key application of modern neural networks. Section 2.4.3.3 provides a review of existing work in out-of-distribution generalisation in image classification.
- Chapter 5 addresses the setting of multi-agent games, in particular multi-agent communication using the setting of 'emergent communication'. Section 2.4.3.4 covers existing work on out-of-distribution generalisation in emergent communication.

2.4.3.1 Data-oriented approaches to OOD generalisation

Simple changes in the available training data can lead to an increase of out-of-distribution generalisation and a decrease in the observable *bias* [114] in machine learning models. For example, an under-represented group can be upsampled to create a training dataset that is more representative of the possible future samples by duplicating the rare samples or removing a number of the most common samples. This approach is motivated further in various instances of problem detection: upsampling the rare problematic events (fraud in the banking context, or a malignant cancer in healthcare applications) is beneficial if the goal is to improve *recall* and false negatives have greater consequences than false positives.

Hooker [114] argues that it is often infeasible to weigh the training dataset in an appropriate way in practical applications. In order to increase representation of the under-represented subsets at the training stage, we need *a priori* knowledge of which groups might be under-represented with respect to the future uses of the model. Such information is sometimes included in the metadata (for instance, additional information on the image source), or it is encoded in the features (for instance, a flag that indicates scarcity or low data quality

for a certain distribution, or the presence of known sensitive attributes such as race or gender).

If the probability density function of the future data distribution Q is available and has a density $q(x)$, and P is the source data distribution with density $\mathbb{P}(x)$ (from which training samples are drawn), the training samples can be weighed in a principled way according to the likelihood ratio $q(x)/p(x)$ (often referred to as Importance Sampling [116]). A machine learning model is then found using *Weighted Empirical Risk Minimisation* (WERM) instead of standard empirical risk minimisation (ERM) based on an unweighted average of the values of the loss function on the entire training dataset. In the case of covariate shift (Figure 2.3a), the weights per sample are obtained as a function of the input features x ($q(x)/p(x)$), whereas in the presence of label shift (Figure 2.3c) the probabilities of labels y are used ($q(y)/p(y)$). WERM in the former case starts from the assumption that the desired loss function can be seen as an estimate of the expectation $\mathbb{E}_{x\sim Q}[\ell(x, y; \mathbf{\Theta})]$, which is equivalently expressed as:

$$\begin{aligned}\mathbb{E}_{x\sim Q}[\ell(x, y; \mathbf{\Theta})] &= \int_D q(x)\ell(x, y; \mathbf{\Theta})dx = \int_D p(x)\frac{q(x)}{p(x)}\ell(x, y; \mathbf{\Theta})dx \\ &= \mathbb{E}_{x\sim P}\left[\frac{q(x)}{p(x)}\ell(x, y; \mathbf{\Theta})\right].\end{aligned}$$

In the equation above, D is the domain over which x ranges – meaning $(-\infty, +\infty)$ for real scalars and simple product domains when x is a vector.

As Vogel et al. [282] point out, it is unrealistic to assume that $q(x)/p(x)$ or $q(y)/p(y)$ ratios are known in real machine learning scenarios, where out-of-distribution generalisation is of high importance. Nevertheless, variants of Importance Sampling and Weighted Empirical Risk Minimisation are used to train neural networks and to compensate for different sizes of treatment groups in healthcare applications [247], to counter known label shift [172], and in off-policy reinforcement learning [217], among others. Byrd and Lipton [37] investigate the effect of Importance Sampling on the training of overparametrised neural networks, and find that the impact of Importance Sampling diminishes over consecutive epochs, and so Importance Sampling needs to be combined with *early stopping* to avoid the asymptotic vanishing effect.

Liu et al [174] describe a simple approach to finding the data points that should be upsampled. First, they train a neural network on the entire dataset, and identify the samples for which the model produces incorrect outputs. Next, these data points are upsampled and a final, new model is trained on an updated dataset. The authors find that this method performs as well as Group DRO [240] on several datasets. They also find that removing the majority (over-represented) samples from the dataset harms the

performance in the minority group, which suggests that there is useful information in the over-represented samples as well.

Finally, a common approach to improving generalisation in machine learning is to collect 'more data'. This might refer to adding more data samples (to counter the errors caused by high variance; for example, language models are prone to high variance due to a large feature space [100]), or adding new features to the existing samples (to counter the errors caused by high bias). Overparametrised neural networks defy the intuition derived from the classical distinction of underfitting (high bias models) and overfitting (high variance models), with performance gains observed past the point of overfitting (*double descent* [205] and *'grokking'* [216]). It is possible that increasing both the amount of data and the model size suffices to achieve many practical aspects of generalisation. However, it is not yet fully understood how such generalisation is achieved. New caveats are regularly discovered: for example, Transformers [280] can achieve few-shot generalisation based on scale alone [35] but only when the data is distributed in a particular way [46]. A popular school of thought in machine learning in 2022 is to work towards combining the advantages of scale (where scale is feasible [27]) with the research developments in theory, interpretability and fairness.

Connection to the thesis Chapter 3 provides new theoretical results on the topic of re-weighting the training distribution, which clarify the relationship between algorithmic approaches to out-of-distribution generalisation (based on Distributionally Robust Optimisation), data-oriented approaches to out-of-distribution generalisation (based on Importance Sampling) and bias mitigation in order to increase fairness in machine learning systems. These results fill the existing gap between the work on algorithmic approaches to OOD generalisation (based on Distributionally Robust Optimisation) and the various approaches based on re-weighting the training data mentioned above [116, 247]. The theoretical results followed by practical recommendations presented in Section 3.4 also aim to increase the existing understanding of the algorithmic and data-based biases mentioned in the literature on fairness in machine learning [114], and in particular to contribute to the fundamental 'is machine learning bias a data problem or a model problem' debate.

2.4.3.2 Algorithmic approaches to OOD generalisation

Algorithmic tools for addressing distribution shift can be divided into methods based on the principles of *pessimism*, *adaptation* and *anticipation*[10].

The pessimism principle assumes the future distribution to be similar to the most difficult subset of the training data and it is implemented under the name of Distributionally

[10] A classification by Chelsea Finn [77].

Robust Optimisation [21]. In the general case, the Distribution Robust Optimisation (DRO) objective is as follows:

$$\min_{\boldsymbol{\Theta}} \sup_{Q \in \mathcal{Q}} \mathbb{E}_{(\boldsymbol{x},y)\sim Q}[\ell(f(\boldsymbol{x};\boldsymbol{\Theta}), y)]$$

where $\mathcal{Q}$ is a set of probability distributions, and f is a model.

Connection to the thesis As discussed later in Section 3.4, we often know that a certain amount of error is unavoidable for a given distribution. The unavoidable error per distribution is expressed via *calibration coefficients* [193, 93]. The efficacy of using DRO largely depends on setting the calibration coefficients, yet, there is no principled way of doing it. Section 3.4.3 proposes a guideline for using DRO including the choice of calibration coefficients based on the newly established theoretical results (Section 3.4.2).

Applications of DRO The definitions of $\mathcal{Q}$ vary depending on the use case of DRO. In one example, for a fixed distribution P we define $\mathcal{Q} = \{Q \mid W_p(P, Q) \leq \epsilon\}$ where W_p is the Wasserstein distance between two distributions (Wasserstein DRO [149]). In Wasserstein DRO, the distributions considered in $\mathcal{Q}$ are small perturbations of the empirical distribution P. In another instance of DRO called Conditional Value at Risk (α-CVaR DRO) [300], the loss is defined as the average loss over the worst $\alpha \in (0, 1)$ fraction of the training samples. In α-CVaR DRO, the set of probability distributions $\mathcal{Q}$ is defined as all possible distributions over the α fraction of the dataset. Finally, Group DRO [240] assumes that data comes from distinct (and known) groups, and the performance on the most difficult group is optimised. The group membership per data point has to be known at the training time, but not at test time. During training, the respective group accuracies are re-evaluated and the 'worst' group can change based on the recent results (to encourage good performance across all groups rather than only on the group that is the most difficult initially).[11] A possible disadvantage of Group DRO is that it requires access to group membership at training time (it works in the setting of learning from multiple data distributions, as described in Chapter 3). However, Group DRO is shown to lead to a higher robustness to spurious correlations (that harm generalisation) than Empirical Risk Minimisation, even in realistic text and image datasets such as CelebA [176] and MultiNLI [289].

Adaptation-based approaches assume access to unlabelled 'out-of-distribution' data at test time. The unlabelled samples might come only from one of the possible future distributions (for instance, handwriting samples from a single new user in a digit recognition system

[11] See Algorithm 1 for an illustration of how to obtain a DRO solution.

that strives to be robust with respect to individual handwriting styles). Adaptive Risk Minimisation (ARM) [305] implements this approach by using meta learning [78].

Anticipation-based approaches assume that the data distribution changes over time. This is most applicable in the context of repetitive distribution shifts in reinforcement learning. For example, a goal that the agent is approaching might oscillate, or the physical conditions (wind, velocity) might change in the context of robotics or autonomous driving. Formally, classical reinforcement learning assumes that the state at step $t+1$, s_{t+1}, depends on the state at step t and the action taken by the agent: $s_{t+1} = S(s_t, a_t)$; the state transition function S is assumed to be constant across episodes. When the state transition function varies in each episode, meaning we have a sequence of state transition functions $S_1, \ldots,$ $S_i, \ldots,$ any useful inference relies on our ability to find a link between the consecutive transition functions S_i and S_{i+1}. In cases when model dynamics are known, it is possible to train the agent with a suitable choice of objective (as in Xie et al. [293]). When the environment changes are predictable, the trained agent can thus 'get ahead' of the environment shift by anticipating it.

2.4.3.3 OOD generalisation in image classification

Modern image classification mostly involves the use of Convolutional Neural Networks (CNNs; Section A.1.2.2) as the main building block of vision models. Motivated by the biological provenance of the idea behind CNNs, Geirhos et al. [85] published an extensive comparison of the generalisation ability of humans and CNNs. They find that humans are more robust than state-of-the-art CNNs to all image manipulations considered (noise, contrast, rotation and more). Interestingly, they show that CNNs trained on a certain image distortion surpass human performance on the exact same distortion type. However, CNNs do not generalise across different distortion types such as different types of noise (salt-and-pepper versus white noise) that are trivial for humans. There are multiple other tasks where CNNs have been shown to exceed human in-distribution performance [146, 105, 250], yet at the same time these models struggle on trivial, visually indistinguishable changes in the image distribution (as shown in adverarial learning; Figure 2.4).

Madan et al. [182] shows that increasing the diversity of in-distribution training data helps CNNs generalise out-of-distribution, even when the number of training samples does not increase (there is a similar existing result in the setting of emergent communication [44]; Section 2.4.3.4). They also show that increased specialisation of parts of a neural network helps in improving the out-of-distribution performance.

Connection to the thesis The design of Neural Function Modules (NFMs; Section 4.1.2) and the idea to evaluate them in the context of out-of-distribution generalisation (Sec-

tion 4.2) relies on the benefits of specialised components in a neural network [44, 182]. NFMs allow the mechanism of specialisation to be easily added to any neural network (not only CNNs). The previous approaches to specialisation of the individual layers are limited in that they still process the entire hidden state [119, 52], while NFMs allow for dynamic selection of the parts of the current hidden state via attention. NFMs also extend the idea of layer specialisation by combining them with a proposal on how to implement the biologically-inspired 'top-down' feedback in neural networks (Section 4.1.2).

Failure modes of existing CNNs: a broader context Apart from the tasks where CNNs perform to a high standard in the in-distribution case but struggle in the out-of-distribution case (such as image classification given enough data), there are also problems where CNNs fall short of human performance even in the in-distribution scenario. Brenden Lake investigates *concept learning* from a small number of image examples [155] in humans and CNNs. While humans can obtain a rich representation of a concept from a few examples (for example, after seeing a single segway for the first time, humans can parse it into the most important parts such as wheels and the handlebar). On the contrary, much of the progress in CNNs has been enabled by training them on large datasets such as ImageNet [146] (1.2 million images in total and around 1200 images per class). The authors propose a dataset for probing concept learning and generalisation in visual models (Omniglot stimulus set [155]) with 'visual Turing tests' that require generalisation, and they aim to recreate the human approach to decomposing unfamiliar concepts using Bayesian Program Learning [155]. Finally, Lake et al. [156] argue that the ability to infer causal relations is missing from the existing CNNs used in image classification (as an example, a picture of people running away from a house destroyed by the waves is captioned by an image captioning model as 'a group of people standing on top of a beach' [156]).

Chapter 4 aims to separate the problem of studying out-of-distribution generalisation in image recognition from the tasks of processing a noisy image and extracting complex visual features. This is approached by using a suite of synthetic tasks (Section 4.2) that allow a full control over the distribution shift between training and test scenarios.

2.4.3.4 OOD generalisation in multi-agent communication

Existing research on emergent communication is focused on symbolic input data (vectors interpreted as properties) [143, 38, 30, 42, 98, 43, 230, 96] and image input data [162, 163, 104, 75, 130, 29, 103, 135, 95, 69]. The work presented in Chapter 5 shows the first results of using graphs (including trees) as input data in the emergent communication setting.

There are a few studies that use out-of-distribution input data in emergent communication.

Bouchacourt and Baroni [30] use out-of-domain data in a cooperative game with a symmetric communication channel, where agents learn to choose an appropriate tool to eat a given fruit. One agent observes a fruit (for example, a 'pear') and the other agent observes two possible tools (for example, a 'spoon' and a 'hammer'). There are multiple rounds of communication and at each stage either of the agents can choose to stop (and choose one of the tools) or to continue the exchange by sending another message. Tools and fruits are represented by property vectors (symbolic data). The agents are rewarded if the tool choice is optimal given a fruit. In this game, the authors test the ability to generalise to five unseen fruit types. The authors find that the performance on the new fruit types nearly matches the standard in-distribution case. This might be the case because all of the fruit types are represented by fixed properties such as 'is crunchy', 'is small' etc (the agents might learn a property-to-tool mapping), as well as due to the limited number of out-of-domain categories (five).

Connection the the thesis In contrast to Bouchacourt and Baroni [30], Chapter 5 studies problems where the accuracy of a random baseline is much lower than 50% (in the games analysed in Figure 5.10, the accuracies of random baselines are 2%, 5% and 10%). However, the methodology and observations overlap: the out-of-distribution accuracy in graph referential games is relatively good (approx. 80% on average in Figure 5.10) and the 'properties' in the baseline methods are modeled in a similar way to Bouchacourt and Baroni [30] with the key addition of graph representations in Game-1 and Game-2 (Section 5.1.2.2).

Chaabouni et al. [44] contribute to the discussion on language compositionality and compositional generalisation in emergent communication. In the generalisation experiments, the set of all possible distinct input samples is partitioned so that the OOD test samples contain only the feature combinations not seen in training (similar to Section 5.1.4). The agent that recives the message has to reconstruct the original input rather than recognising it among similar samples, and the input data is symbolic. While some authors use compositional generalisation (the agents' accuracy on the samples consisting of new combinations of familiar features) as a measure for how compositional the emerged language is [57, 142], Chaabouni et al. [44] show that empirically observed compositional generalisation can be achieved without a compositional language (that is, without the signs that the agents learn to communicate and disentagle the compositional input in a compositional way). The authors find that the agents in an emergent communication game defined over symbols can generalise to unseen combinations of familiar features if the

input space is 'sufficiently large'. For instance, if the agents see 900 distinct combinations in the training set, they achieve a higher accuracy on the held-out 100 out-of-domain distinct combinations than in the case of observing 90 combinations and being evaluated on 10. This is shown to hold regardless of the number of training samples. A large variety of compositional inputs might give the agents a stronger cue about the compositional nature of the data. The authors conclude that the languages emerged in these games are not compositional because the channel capacity C to input size $|I|$ ratio is $\frac{C}{|I|} \approx 6$, whereas it is $\frac{C}{|I|} = 1$ for a perfectly compositional language. The authors also stipulate that compositionality is a stronger condition than compositional generalisation, with language compositionality being a sufficient but not necessary condition for compositional generalisation. Finally, the authors show that compositional languages are easier to teach to new agents (Spearman correlation 0.9), and these new agents are more likely to be able to generalise in a compositional way if they are taught a compositional language (Spearman correlation 0.8).

Connection to the thesis The experiments on graph referential games in Chapter 5 (Section 5.1) use a similar high-level design of OOD generalisation experiments, however, input data includes graphs, sequences and bag-of-words, and the goal of the agent who receives the message is to recognise the original input among similar samples (distractors).

A recent paper by Mu and Goodman [203] proposes communication over a set of images rather than a single image at a time (the idea previously investigated by Guo et al. [94]). The authors use the idea that human language might have emerged in order to communicate 'generalisations', for instance, to use the word 'lion' for as many real examples of a lion as possible, including those that have not yet been seen [158]. In the games proposed by Mu and Goodman [203], the agents are explicitly encouraged to generalise: the agent with the role of a 'teacher' communicates a group of images belonging to a single concept (for instance, images of a 'red triangle'), and the games vary depending on how many agents see the images and how different the images belonging to a single concept are (for instance, the 'red square' on a dark background might have a different rotation and size). In the games defined over images of geometric shapes, the authors use 10 images per target and 10 distractors, and the communication channel has the same bandwith as the number of features needed to recognise the ground-truth concept combinations. Apart from the geometrical shapes, the authors also use images of birds (100 classes at training time and 50 classes at test time, 40-60 images per class). In terms of language compositionality, the authors find that in the games defined over a set the language is more systematic, regardless of whether the input images are shared or unshared between the agents. However, the authors find that the accuracy achieved by the emergent language fails to approach the

accuracy of referential games: it is more difficult to learn to communicate based on sets than to communicate specific references.

Connection to the thesis Section 5.2 investigates image-based emergent communication. However, it is focused on learning to describe an image at a time, and the images are sourced from a dataset of realistic images. Additionally, the main research questions in Section 5.2 concern the effect of model complexity and population size on generalisation in image-based emergent communication.

2.5 Summary

This chapter describes relevant background for the thesis. Section 2.2 presents methodology used to obtain the experimental results presented throughout the contribution chapters. Section 2.1 sets the notation used throughout the thesis. Section 2.3 describes important concepts such as attention, which are some of the building blocks of the novel work in this thesis. Section 2.4 presents an overview of the existing research on out-of-distribution generalisation in machine learning, with the focus on the learning scenarios investigated in this thesis.

The remaining relevant background material is in the Appendix – it contains minimal background on neural networks, the main building block of the work shown in Chapter 4 and Chapter 5.

2.5.1 Gaps in existing work

The contribution chapters address the following gaps in existing work:

- The practice of processing the entire hidden state (the entire unprocessed output of a hidden layer is the input to the consecutive layer; Figure A.2) in neural networks as described in Section A.1 and Section 2.4.3.3. This is addressed by the introduction of Neural Function Modules (Section 4.1.2) and by providing new results on their ability to improve out-of-distribution generalisation across an extensive lists of tasks and models in the context of image classification (Section 4.2).

- The lack of studies on graph representation learning in emergent communication; in particular, on the influence of data representation and the corresponding data representation method in the face of a covariate shift in input data (Section 2.4.3.4). This is addressed in Section 5.1.4 by the introduction of graph referential games, and by providing the results of a comparison of representation learning methods in terms of compositional generalisation and language compositionality.

- The lack of a standardised set of tasks (unit tests) to evaluate algorithms in terms of the ability to tackle domain shifts by learning invariant features. This is addressed by the introduction of Linear unit tests and by providing the results of a comparison of existing methods using Linear unit tests (Section 3.3).
- The lack of a full understanding of the connection between re-weighting training examples (Section 2.4.3.1) and applying the algorithmic approach to bias mitigation (Distributionally Robust Optimisation; Section 2.4.3.2). This is addressed by providing a set of theoretical results on the equivalence of these approaches (Section 3.4).
- The lack of a principled way of setting calibrated coefficients in DRO (Section 2.4.3.2). This is remedied by providing a set of practical recommendations on using DRO and setting calibration coefficients based on the new theoretical findings (Section 3.4.3).
- The lack of studies on the influence of population size and model diversity on generalisation in emergent communication (in the most common setting of referential games) with realistic images (Section 2.4.3.4). These research questions are investigated in Section 5.2.

The list above is a brief summary of the ways existing work presented in this chapter motivates the contributions made in this thesis. Each of the contribution chapters expands on its respective contributions.

Chapter 3

Learning from multiple distributions

Nature does not shuffle the data,
so we should not either.
–Léon Bottou (2019)

This chapter focuses on *algorithms that learn from multiple training distributions simultaneously*, as opposed to the standard practice of an algorithm learning from one training dataset under *independent and identically distributed* (i.i.d) data assumptions. In addition to exploring the framework of multiple training distributions, this chapter is model-agnostic, as opposed to the later chapters focused on neural networks. The main results presented in Section 3.3 and Section 3.4 reveal limitations of the error minimisation algorithms that are currently used to improve distribution robustness in machine learning. On one hand, a new dataset reveals that existing optimisation methods fail to generalise in presence of spurious correlations, even if data is linearly separable (Section 3.3). On the other hand, new theoretical results show that a commonly used practical formulation of out-of-distribution generalisation, *Distributionally Robust Optimisation* [21], is equivalent to standard *Empirical Risk Minimisation* [279] on an adequately weighted training dataset (Section 3.4).

Chapter structure

Section 3.1 briefly describes the limitations of the standard approach of learning under the i.i.d assumption and motivates the framework of using multiple distinct training distributions instead of shuffling all the available data at random, and assuming a single training dataset that represents a single distribution.

Section 3.2 formally defines the framework of learning from multiple training distributions, which is used in practice in the results sections, Section 3.3 and Section 3.4.

Section 3.3 describes a set of new datasests developed to standardise the evaluation of out-of-distribution generalisation algorithms. I present the datasets and an extensive evaluation of the existing algorithms. This work shows that problems typically considered to be domain-specific challenges (for example, recognising the actual object of interest in an image rather than exploiting image background in image recognition tasks [20, 292]) are challenging even in simplified, linearly separable instances. It also provides the first standardised testbed for comparing and evaluating algorithms from the fields of causality [301], invariace discovery and out-of-distribution generalisation, with the goal of reducing the risk of duplicating efforts in these closely related lines of work.

Section 3.4 presents a set of new theorems, which establish the conditions under which a robust solution to learning from multiple training distributions corresponds to optimising a mixture of distributions. The approach considered here is *Distributionally Robust Optimisation* (DRO) [21]. Based on the theorems, I propose new guidelines for applying DRO in scenarios where there is an imbalance between the training distributions; for instance, when there is data scarcity or lack of quality data to represent one of the distributions. This work shows that the performance of DRO measured on a particular dataset from the set of training datasets (that might represent different distributions) can be no better than the performance of the best solution optimised specifically for the given dataset. I discuss ramifications of this result for fairness in machine learning systems.

Related publications I presented some of the results from Section 3.3 at the competitive *NeurIPS Workshop on Causal Discovery and Causality-Inspired Machine Learning* (oral presentation).

I presented the results from Section 3.4 at the *NeurIPS Workshop on Optimization for Machine Learning (OPT 2021)* (oral presentation) and at the *NeurIPS workshop on Algorithmic Fairness through the Lens of Causality and Robustness (AFCR 2021)* (oral presentation). The results were later published at *The Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI 2022)* as a student abstract (oral presentation), which received the AAAI-22 Best Student Abstract Honorable Mention (conference acceptance rate: 15%, with 19 out of 111 accepted abstracts selected for an oral presentation, and 3 out of 19 receiving a mention).

I also gave a presentation on existing work in learning from multiple distributions at the *Causality & Domain Adaptation Reading & Work Group* (the group of Ferenc Huszár).

3.1 How the i.i.d assumption fails

The standard practice of shuffling the training and test examples at random brings the test distribution closer to the training distribution, and it allows the use of algorithms that assume i.i.d data [279]. On the other hand, this practice is also motivated by the use of data-hungry methods such as neural networks, which encourages the curation of large datasets. The pooled, shuffled and randomly split samples are convenient, however, they might not correspond to real future examples, to which an algorithm deployed in practice is exposed. In fact, it has been frequently shown that optimising for the in-distribution test performance does not translate to good performance in presence of realistic distribution shifts [20, 187, 189, 86, 228, 231]. The discrepancy between the performance on shuffled data in the synthetic research scenario, and the performance on the future examples when deploying a model in practice, plays the critical role in machine learning failing to fulfil the promises of aritificial intelligence. Paraphrasing Zoubin Ghahramani (the quote from the panel discussion at the Workshop on Advances in Approximate Bayesian Inference, 2017):

> *The big lie in machine learning is that testing data comes from the same distribution as training data.*

Apart from the risk of obtaining a lower future performance than estimated based on the in-distribution test accuracy, in certain scenarios pooling all the available examples and shuffling them at random can reinforce systemic biases against pre-existing minority subpopulations represented in the data. Algorithms that optimise for the minimum average error over the entire dataset, such as the ubiquitous *Empirical Risk Minimisation* (ERM) [278], yield models that perform poorly on subpopulations that are already at risk due to pre-existing biases. This is most pronounced when ERM (based on minimising the average prediction error) produces solutions that privilege the latent majority populations over the latent minority groups (Figure 3.1). This is a simple phenomenon in terms of the underlying mathematics, however, it can have serious negative practical consequences. The solutions that are skewed towards the majority subpopulations have been shown to be consequential in scenarios such as court verdicts, loan applications and healthcare interventions [219, 56, 223, 194, 5]. For example, hiring systems and ad-targeting algorithms based on minimising average error were found to discriminate against female users by more frequently proposing executive and technical jobs to men [59, 60].

There is clearly a challenging trade-off between the wish to collect as many relevant samples as possible and the wish to keep the i.i.d assumption. An alternative is to assume the presence of several disjoint training datasets, each representing a meaningful subpopulation

(for instance, each subpopulation corresponds to the samples collected in a different country, or under different technical conditions).

Figure 3.1 compares these two approaches using the same classifier (linear SVM [108]). When using the i.i.d assumption, the inferred model misclassifies the visible minority subpopulation at the level of a random choice model, while performing nearly perfectly when evaluated on the majority subpopulation. When the assumption of multiple disjoint datasets is used, both the majority and the minority subpopulations are classified with the accuracy of approx. 85%. The latter solution is preferred in the applications where fairness and *group robustness* [118] are crucial; for example, when the majority and the minority groups represent different demographic groups in machine learning applications such as disease prediction.

3.2 Generalisation from multiple distributions

This section defines a framework for learning to generalise from multiple distributions, which will be used for the rest of this chapter. This section explains how the multi-distribution setup differs from the standard practice of using Empirical Risk Minimisation, and how it is motivated by out-of-distribution generalisation.

Consider a dataset of N examples $\boldsymbol{x}_n \in \mathbb{R}^d$ (all d-dimensional vectors) and corresponding labels $y_n \in \mathbb{R}$. Here, we focus on the *supervised learning* setting with access to a set of pairs $\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N)\}$. The goal of learning is to estimate parameters $\boldsymbol{\Theta}$ of a function $f(\cdot, \boldsymbol{\Theta}) : \mathbb{R}^d \to \mathbb{R}$ based on training data. While out-of-distribution generalisation is equally important in unsupervised, semi-supervised and reinforcement learning settings, the contributions presented in this chapter consider the setting of supervised learning.

In supervised learning, the i.i.d assumption implies that 1) training and test data samples stem from the same data generative process (distribution) $\mathcal{D}$, meaning all $(\boldsymbol{x}_i, y_i) \sim \mathcal{D}$; 2) the generative process has no memory of past generated samples, that is, any two data points $(\boldsymbol{x}_i, y_i)$ and $(\boldsymbol{x}_j, y_j)$ are statistically independent.[1] This assumption is used when measuring the ubiquitous *in-distribution generalisation* performance metrics (for instance, standard test accuracy in classification) that assume no difference between the data generating process for the training and test examples. The i.i.d assumption holds for the outcomes of a generative process (usually unknown) that produces both the training and test examples. However, the assumption is unrealistic when applied to data coming from multiple generative processes in practical applications. Large datasets used in modern

[1] Recall the definition of independence is stronger than pairwise independence, which is what the formulation of 'no memory' means here. For i.i.d random samples, we have that for joint distribution p of n variables with density $p_{\mathcal{D}}$, $p(\boldsymbol{X}, \boldsymbol{y}) = \prod_{i=1}^{n} p_{\mathcal{D}}(\boldsymbol{x}_i, y_i)$.

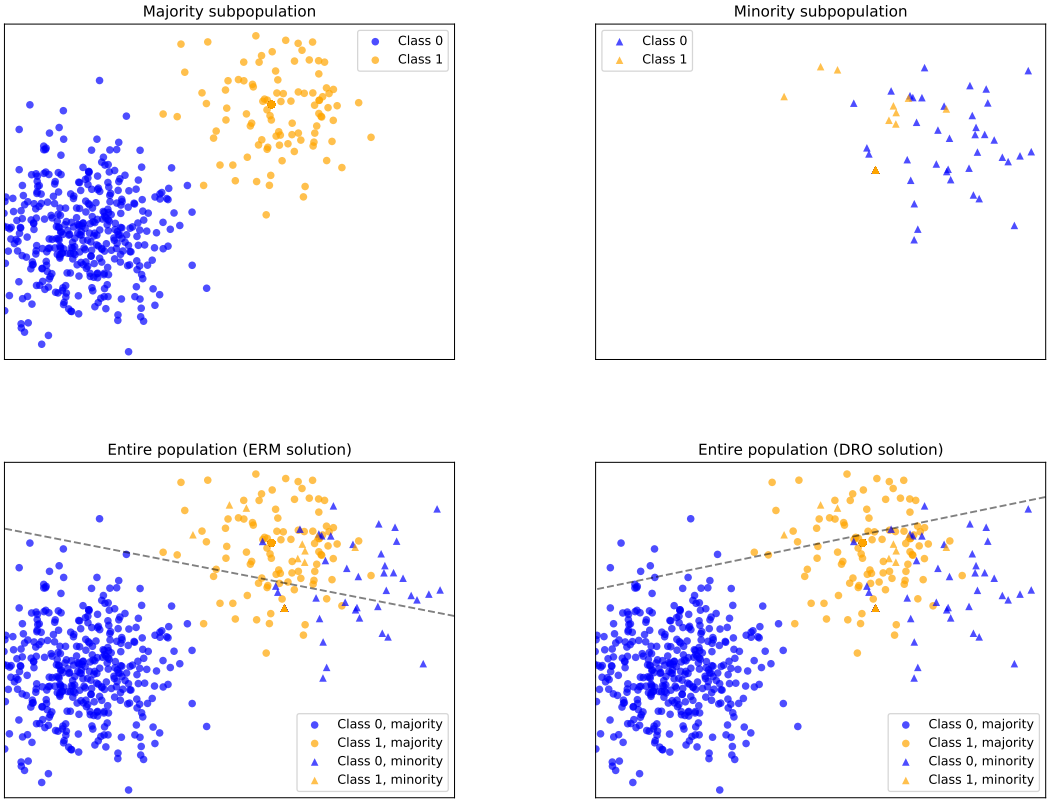


Figure 3.1: Illustration of the failings of ERM and i.i.d assumption in a linear binary classification problem, where the training dataset can be partitioned to a *majority* and a *minority* subpopulation.
When ERM is applied to the entire dataset, a significant proportion of the minority subpopulation is misclassified. The training accuracy of the ERM solution (bottom left plot) is 95.8% for the majority subpopulation and 48% for the minority subpopulation. DRO (investigated in Section 3.4) uses the approach of learning from multiple distributions instead of assuming i.i.d data, which improves the performance in the minority subpopulation (bottom right plot, 84%) while keeping the performance on the majority subpopulation at an acceptable level (85%). The criterion of 'acceptable' performance and the trade-off between the performance on individual subpopulations will be discussed in Section 3.4. In this example, the two decision boundaries are obtained using a linear Support Vector Machine (SVM) [108].
DRO leads to a more balanced solution than ERM with respect to the majority and minority subpopulations.

machine learning are often curated based on samples coming from systematically different generative processes (for instance, MNIST dataset [168] was collected by pooling and shuffling smaller datasets corresponding to different authors, each with an individual handwriting style). Practical sources of statistical differences between training and test data include changes in the behaviour of machine learning system's users and systematically reduced (or improved) quality of data over time. For instance, if a system works poorly for a certain group of users, those users are more likely to avoid using it and to contribute fewer new data samples, which can lead to *disparity amplification over time* [186] – meaning that the performance on the test data set for these groups will deteriorate over time.

In an in-distribution setting with the i.i.d assumption, Empirical Risk Minimisation is the most commonly used method in supervised learning. Recall that we can define the *risk* for a predictor $f(\cdot;\boldsymbol{\Theta}) : \mathbb{R}^d \to \mathbb{R}$ as a function R that depends only on f:

$$R(f) = \mathbb{E}_{\boldsymbol{x},y}[\ell(f(\boldsymbol{x};\boldsymbol{\Theta}), y)] = \int_{\boldsymbol{x},y} \ell(f(\boldsymbol{x};\boldsymbol{\Theta}), y)p(\boldsymbol{x}, y)d(\boldsymbol{x}, y) \tag{3.1}$$

where ℓ is a *loss function* that measures the 'deviation' between the ground truth label and the predicted label and p is the joint probability density function. In practice, we have a finite sample of inputs and outputs. Hence we define *empirical risk* R_{emp} as a quantity that depends on a predictor f and observed data $\boldsymbol{X}, \boldsymbol{y}$ (as defined above). Empirical risk is then simply defined as:

$$R_{\text{emp}}(f, \boldsymbol{X}, \boldsymbol{y}) = \frac{1}{N}\sum_{n=1}^{N} \ell(\hat{y}_n, y_n) \tag{3.2}$$

where $\hat{y}_n = f(\boldsymbol{x}_n, \boldsymbol{\Theta})$.

A crucial limitation of this approach is that machine learning systems use large datasets consisting of data samples coming from *multiple generative processes*. Data samples are obtained under varying conditions, such as time, location or preprocessing techniques, which affect data distribution. Shuffling those samples destroys information on which data features stem from data collection details (for instance, artifacts from a medical device such as a frame or a foreground timestamp), and which features describe the phenomenon of interest (for instance, the presence of an illness).

Paraphrasing Léon Bottou, *Nature does not shuffle data* [7]. Since the i.i.d assumption is artificially introduced in the training data, we cannot expect that future examples will be i.i.d. Frequently, we cannot afford to only use data coming from one generative process due to data scarcity. Instead, consider using multiple training *environments* or *subpopulations* e representing different sources of data, with the main parallel aims of 1)

achieving a low error on realistic future examples and 2) discovering features that describe the pheonomenon of interest (*invariant features*) rather than data collection idiosyncrasies that are unlikely to be stable in the future (*spurious features*).

Formally, the environments e are drawn from a large set of environments $\mathcal{E}_{\text{all}}$, out of which we only observe the set of training environments $\mathcal{E}_{\text{tr}} \subset \mathcal{E}_{\text{all}}$. For each environment e we associate random variables $\boldsymbol{x}^e$ and y^e and a generative process defining them (hence they may have different distributions across environments). The task is to learn a predictor f that performs well on all environments. Namely, we wish to minimise:

$$R^{\text{OOD}}(f) = \max_{e \in \mathcal{E}_{\text{all}}} R^e(f) \tag{3.3}$$

where $R^e(f) = \mathbb{E}_{\boldsymbol{x}^e, y^e}[\ell(f(\boldsymbol{x}^e), y^e)]$ is the risk for a predictor f associated with environment e. In other words, the out-of-distribution risk $R^{\text{OOD}}(f)$ corresponds to the *worst* possible performance of f on any environment.

Following this theoretical framework, we define a more practical *Distributionally Robust Optimisation* statement, which only considers the environments seen during training:

$$R^{\text{DRO}}(f) = \max_{e \in \mathcal{E}_{\text{tr}}} R^e(f) \tag{3.4}$$

The assumption is that the observed environments are representative of $\mathcal{E}_{\text{all}}$. In practice we use the empirical risk R^e_{emp} as an estimate of R^e. For minority subpopulations we might have additional problems in the form of data scarcity or low data quality, making a subpopulation inherently more difficult to classify. Since achieving a high performance on such 'hard' environments might come at the expense of significantly lowering performance in other environments, we introduce *calibration coefficients* r_e that adjust the subpopulation risks in line with this inherent difficulty of an environment. DRO is thus sometimes (such as in Sagawa et al. [240], where calibration coefficients are called *generalisation gaps*) seen in the following form, which in this thesis we refer to as *Calibrated DRO*:

$$R^{\text{DRO}}(f) = \max_{e \in \mathcal{E}_{\text{tr}}} (R^e(f) - r_e) \tag{3.5}$$

There are several ways of choosing these coefficients. We discuss the implications of this choice in Section 3.4.

How do invariant and spurious features relate to out-of-distribution generalisation? If $\mathcal{E}_{\text{all}}$ is the set of all hypothetical and observable conditions under which data describing our problem can be collected, $\mathcal{E}_{\text{tr}} \subset \mathcal{E}_{\text{all}}$ is the set of observed environments used in training. Such a set can be processed by a standard supervised learning algorithm if the environments are pooled and shufffled. Shuffling removes the information about how

the data distribution changes depending on different conditions (data sources and data generation specifics). This information is useful for predicting whether a feature will hold in the unseen future examples (invariant feature), or if it is an idiosyncrasy of no relevance to the problem at hand (spurious feature). Each environment represents an arbitrary training distribution, and features that are stable across these collected environments are expected to hold in the future.

3.3 Linear unit tests

This section presents a new standardised suite of synthetic linear unit tests. These problems aim to cover a wide range of challenging discrepancies between training and test distributions. I will start by describing the unit tests and specific challenges they pose, continuing with presentation of the results using the most recent algorithms from the fields of out-of-distribution generalisation and causality, concluding by ablation studies and a summary of the results.

The name 'unit tests' is inspired by the related concept in software engineering: they are small examples used to determine whether the basic functionality is correct. A unit test usually probes the smallest piece of code that can be logically isolated in a system. Similarly, *linear unit tests* probe machine learning models on simple inputs with a clear expected output. Such tests were missing in prior work on out-of-distribution generalisation, and in machine learning research in general. I hope this contribution will be used to foster test-driven development practices in machine learning research.

Generating linear unit tests In every problem, we consider a set of environments $\mathcal{E} = \{E_j\}_{j=1}^{n_{\text{env}}}$, where n_{env} is the number of environments. For each of these environments $e \in \mathcal{E}$, we generate a dataset $D_e = \{(\boldsymbol{x}_i^e, y_i^e)\}_{i=1}^{n_e}$, where n_e is the number of samples and each $\boldsymbol{x}_i^e \in \mathbb{R}^d$ for a fixed dimension d. This dimension d represents the total number of features, both spurious (of which there are d_{spu}) and invariant (of which there are d_{inv}). Any input vector $\boldsymbol{x}$ in a dataset D_e is a concatenation of features $\boldsymbol{x}_{\text{inv}} \in \mathbb{R}^{d_{\text{inv}}}$ and $\boldsymbol{x}_{\text{spu}} \in \mathbb{R}^{d_{\text{spu}}}$. The goal is to learn an invariant predictor that estimates the target variable y by relying on $\boldsymbol{x}_{\text{inv}}$, and ignoring $\boldsymbol{x}_{\text{spu}}$.

Train, validation and test splits are sampled per problem and per environment. *In order to render the association between spurious features and labels irrelevant (and to ensure that only the algorithms ignoring spurious features achieve a low test error), values of spurious features in the test set are randomly permuted across examples.* Specifically, the $n_{\text{test}}^e \times d_{\text{spu}}$ matrix of values of spurious features in test examples has its rows (each row representing spurious features of one example) shuffled per each environment. This is

not the standard practice in machine learning, where test error is usually measured on a sample coming from the same distribution as the examples used in training, which tends to yield higher results.

The following discussion is divided into six sections. In Section 3.3.1, I discuss existing work on evaluating OOD generalisation – these are various toy problems that show how various models fail to develop distributionally robust solutions. They provide inspiration for the three sets of linear unit tests – `Example1`, `Example2`, `Example3` – which are respectively described in Sections 3.3.2, 3.3.3 and 3.3.4. I then describe the scrambled variations of these problems in Section 3.3.5. Finally, Section 3.3.6 describes the results of evaluating existing algorithms on the proposed 'unit tests'.

3.3.1 Related work in evaluating OOD generalisation

Existing datasets for studying out-of-distribution generalisation are limited to the *domain adaptation* setting [82] with an access to prior knowledge on the structure of the test distribution in the form of unlabelled examples or a small number of labelled examples. Those are typically image datasets where high dimensional, non-linear features reduce interpretability of the results. [111] use the standard computer vision datasets (*MNIST* [167], *CIFAR* [276]) split into realistic and perturbed examples to measure out-of-distribution generalisation. By contrast, the aim of the linear unit tests is to evaluate the methods in the ability to learn a representation that remains stable across changing environments. If an algorithm can learn an invariant representation, it is robust to spurious, environment-dependent features with no assumptions on the particular distribution of the spurious features. Moreover, all of the examples are linear and explicitly tested in their ability to ignore spurious features, which makes it harder to exploit shortcuts that are common in more complex and noisy data, such as images or language.

Arjovsky et al. [7] introduced *Colored MNIST*, an image binary classification task derived from MNIST, where each digit image is coloured based on the class label. In the test dataset, the correspondence between the label and the colour disappears, so algorithms based on minimising average training error fail due to exploiting the unstable colour-label correlation at the expense of learning the meaningful digit-label correlation. *Regression from causes and effects* (Section 3.3.2) explores a similar problem in the setting of linear regression.

Another related problem was proposed by Parascandolo et al. [210] and is illustrated in Figure 3.2. In order to extract invariant features in this task, an algorithm must learn a highly non-linear decision boundary (*spiral*). Spurious features offer a linear decision boundary: a vertical one in the environment A and a horizontal one in the environment B.

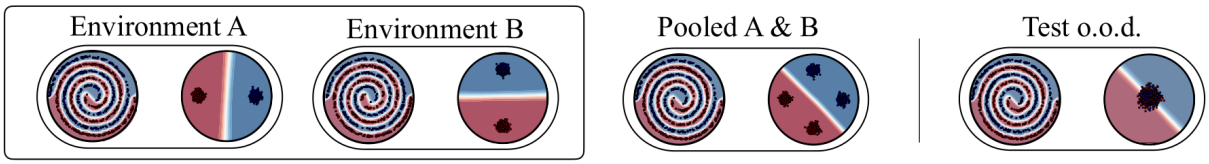


Figure 3.2: A binary classification problem with four features – two spurious and two invariant.
When data points are projected onto the invariant features, the decision boundary follows a 'spiral'-like pattern. When the points are projected onto the spurious features, the decision boundaries are linear but differ across environments. The spurious features might be used to find the decision boundary even for the entire training set (pooled environments A and B), in which case the decision boundary is still linear. However, this connection disappears when probing an algorithm for out-of-distribution generalisation in the test case, where the spurious features (that lead to the linear 'shortcut' solutions) are identically distributed. Figure by Parascandolo et al. [210].

If the training environments are pooled and shuffled, data is still linearly separable due to a diagonal decision boundary along the dimensions represented by spurious features. This solution can be obtained by minimising training error, however, in the test dataset spurious features are replaced with random samples. *Small invariant margin* (Section 3.3.4) shows that this problem poses a difficulty to existing algorithms even when the invariant decision boundary is linear.

3.3.2 Regression from causes and effects

This section describes a *linear least-squares regression* problem where features contain causes and effects of the target variable (based on Arjovsky's example [7], but with slight modifications). First, I describe an example of such a problem to show why the distribution of the invariant features changes across environments, while an invariant solution using these features can be obtained. Next, I present a general definition which can be used to generate a family of specific instances of regression from causes and effects. Finally, I discuss challenges this family of problems poses to existing out-of-distribution generalisation algorithms.

Consider the following toy problem. Suppose we have three environments E_1, E_2, E_3 in which the features and the outputs are generated using the same process: a normally-distributed random variable X_{inv} centred at 0 is generated, after which it gets multiplied by 2 and a Gaussian noise of environment-dependent variance is added to produce the output Y. The spurious feature X_{spu} which acts as the effect of Y is produced by multiplying Y by an environment-dependent constant and adding Gaussian noise with environment-agnostic noise. The spurious features X^e_{spu} and invariant features X^e_{inv} are thus generated

$$X_1^e \xrightarrow{2} Y^e \xrightarrow{w(e)} X_2^e$$

Figure 3.3: Regression from causes and effects.
The numbers on the arrows represent multiplication, whereas noise terms are not shown. The label $w(e)$ represents the fact that the factor by which the output Y^e is multiplied to get the spurious feature X_2^e is a function of the environment e. For example, in the toy problem, $w(E_1) = 1$, $w(E_2) = 3$, $w(E_3) = -1$.

as follows:

$$\begin{array}{lllllllll}
X_{\text{inv}}^{E_1} & \sim & \mathcal{N}(0,10), & X_{\text{inv}}^{E_2} & \sim & \mathcal{N}(0,50), & X_{\text{inv}}^{E_3} & \sim & \mathcal{N}(0,20), \\
Y^{E_1} & \sim & 2X_{\text{inv}}^{E_1} + \mathcal{N}(0,10), & Y^{E_2} & \sim & 2X_{\text{inv}}^{E_2} + \mathcal{N}(0,50), & Y^{E_3} & \sim & 2X_{\text{inv}}^{E_3} + \mathcal{N}(0,20), \\
X_{\text{spu}}^{E_1} & \sim & Y^{E_1} + \mathcal{N}(0,1), & X_{\text{spu}}^{E_2} & \sim & 3Y^{E_2} + \mathcal{N}(0,1), & X_{\text{spu}}^{E_3} & \sim & -Y^{E_3} + \mathcal{N}(0,1).
\end{array}$$

The input data for each environment e is the vector $[X_{\text{inv}}^e, X_{\text{spu}}^e]^\mathsf{T}$ and the solution to the linear regression problem $Y^e \approx \alpha_1 X_{\text{inv}}^e + \alpha_2 X_{\text{spu}}^e$ that we want to find is $\hat{\alpha}_1 = 2$ and $\hat{\alpha}_2 = 0$. However, a predictor trained on an individual environment attaches a high significance to the spurious feature and might prefer an erroneous solution because of the high noise variance in the process generating Y. The predictor $(\hat{\alpha}_1', \hat{\alpha}_2') = (0, 1)$ is preferred in the first environment, but this clearly does not generalise to the other environments. The causal relations between the variables are visualised in Figure 3.3.

We extend this example to a more general setting. We generate a dataset in which the target variable y depends on the result $\tilde{\boldsymbol{y}} \in \mathbb{R}^{d_{\text{inv}}}$ of a linear mapping from the set of invariant features, and the spurious features are the result of applying a linear mapping on $\tilde{\boldsymbol{y}}$ that depends on the environment. In other words, the causes of the target variable are the invariant features and the effects of the target variable are the spurious features. Each dataset D_e for every $e \in \mathcal{E}$ is constructed by sampling for each $i = 1, \ldots, n_e$:

$$\begin{array}{rclrcl}
\boldsymbol{x}_{\text{inv},i}^e & \sim & \mathcal{N}_{d_{\text{inv}}}(0, (\sigma^e)^2), & & & \\
\tilde{\boldsymbol{y}}_i^e & \sim & \mathcal{N}_{d_{\text{inv}}}(\boldsymbol{W}_{yx}\boldsymbol{x}_{\text{inv},i}^e, (\sigma^e)^2), & y_i^e & \leftarrow & \frac{2}{d}\mathbf{1}_{d_{\text{inv}}}^\mathsf{T}\tilde{\boldsymbol{y}}_i^e, \\
\boldsymbol{x}_{\text{spu},i}^e & \sim & \mathcal{N}_{d_{\text{spu}}}(\boldsymbol{W}_{xy}^e\tilde{\boldsymbol{y}}_i^e, 1), & \boldsymbol{x}_i^e & \leftarrow & (\boldsymbol{x}_{\text{inv},i}^e, \boldsymbol{x}_{\text{spu},i}^e);
\end{array}$$

where matrices $\boldsymbol{W}_{yx} \in \mathbb{R}^{d_{\text{inv}} \times d_{\text{inv}}}$ and $\boldsymbol{W}_{xy}^e \in \mathbb{R}^{d_{\text{spu}} \times d_{\text{inv}}}$ have all their entries i.i.d from the standard normal distribution (mean 0, variance 1), and they represent the linear mappings (causes to target, target to effects). The superscript in $\boldsymbol{W}_{xy}^e$ denotes the fact that the mapping from $\tilde{\boldsymbol{y}}$ to $\boldsymbol{x}_{\text{spu}}$ varies across environments.

The invariant solution $y \approx \frac{2}{d_{\text{inv}}}\mathbf{1}_{d_{\text{inv}}}^\mathsf{T}(\boldsymbol{W}_{yx}\boldsymbol{x}_{\text{inv}})$ leads to the only linear regression model whose coefficients are independent of the environment e.

Invariant features do not need to come from the same distribution across environments.

The word 'invariant' refers to the fact that the link between $\boldsymbol{x}^e_{\text{inv}}$ and $\boldsymbol{y}^e$ is the same regardless of the environment. In practice, we are interested not only in an invariant model, but also in one that has a strong predictive power (for instance, regression from an empty set of features to $\boldsymbol{y}^e$ is invariant, yet it does not capture the relation between the input data and the target variable).

This problem poses several challenges to out-of-distribution generalisation algorithms:

1. The distribution of the invariant features $\boldsymbol{x}^e_{\text{inv}}$ differs depending on the environment (note that the standard deviation is σ^e). This disallows the use of methods which seek features with matching distributions across environments, such as domain-adversarial methods [82].

2. The distribution of the residuals $y^e_i - \boldsymbol{W}\boldsymbol{x}^e_i$ varies across environments for any appropriately-sized matrix W due to the inclusion of spurious features. This proves to be important because standard invariance discovery methods such as Invariant Causal Prediction [213] find a solution only when residuals are sufficiently similar across environments.

3. Most of the existing out-of-distribution generalisation algorithms focus on the classification setting. Here, the target variable is continuous, which disallows the use of techniques such as domain classification networks [82]. An existing classification variant of prediction from causes and effects is seen in Colored MNIST [7].

3.3.3 Cows and camels

Existing computer vision algorithms show excellent performance when trained and evaluated on images with a consistent background, yet the performance drastically decreases when the background (a spurious feature[2]) changes. For instance, in a binary classification of images of cows and images of camels, state-of-the-art models can successfully minimise the training error by exploiting a shortcut: 'a green background constitutes a cow, a beige background constitutes a camel'. This example is seen in the 'Recognition in Terra Incognita' paper by Beery et al. [20].

This section presents a linear instance of this problem under the assumption that a background does not, in fact, determine the identity of an object in the foreground. Invariant features in a certain representation constitute a causal explanation of the object (*What is a cow?*), similarly as in Section 3.3.2. Figure 3.4 illustrates this problem in the presence of three environments.

[2]Here, we assume that the goal is to correctly label the foreground object regardless of the image background, following Beery at al [20].

Let:

$$\boldsymbol{\mu}_{\text{cow}} = \mathbf{1}_{d_{\text{inv}}}, \qquad \boldsymbol{\mu}_{\text{camel}} = -\boldsymbol{\mu}_{\text{cow}}, \qquad \nu_{\text{animal}} = 10^{-2},$$
$$\boldsymbol{\mu}_{\text{grass}} = \mathbf{1}_{d_{\text{spu}}}, \qquad \boldsymbol{\mu}_{\text{sand}} = -\boldsymbol{\mu}_{\text{grass}}, \qquad \nu_{\text{background}} = 1.$$

Here $\mathbf{1}_k$ is a k-dimensional vector whose components all equal 1, and ν_{animal} and $\nu_{\text{background}}$ are fixed constants representing how strongly the 'animal' and 'background' features affect the class label in the training set. The choice of ν_{animal} and $\nu_{\text{background}}$ values aims to represent the problem of background features giving a stronger signal in training (for instance, in an image a background often contains a larger number of pixels than the object of interest).

For each environment e, we define p^e as the proportion of 'images' with 'grass' as the background, and s^e as the proportion of 'images' with cows as the target object. The datasets D_e are sampled for every environment $e \in \mathcal{E}$ and each data point $i = 1, \ldots, n_e$:

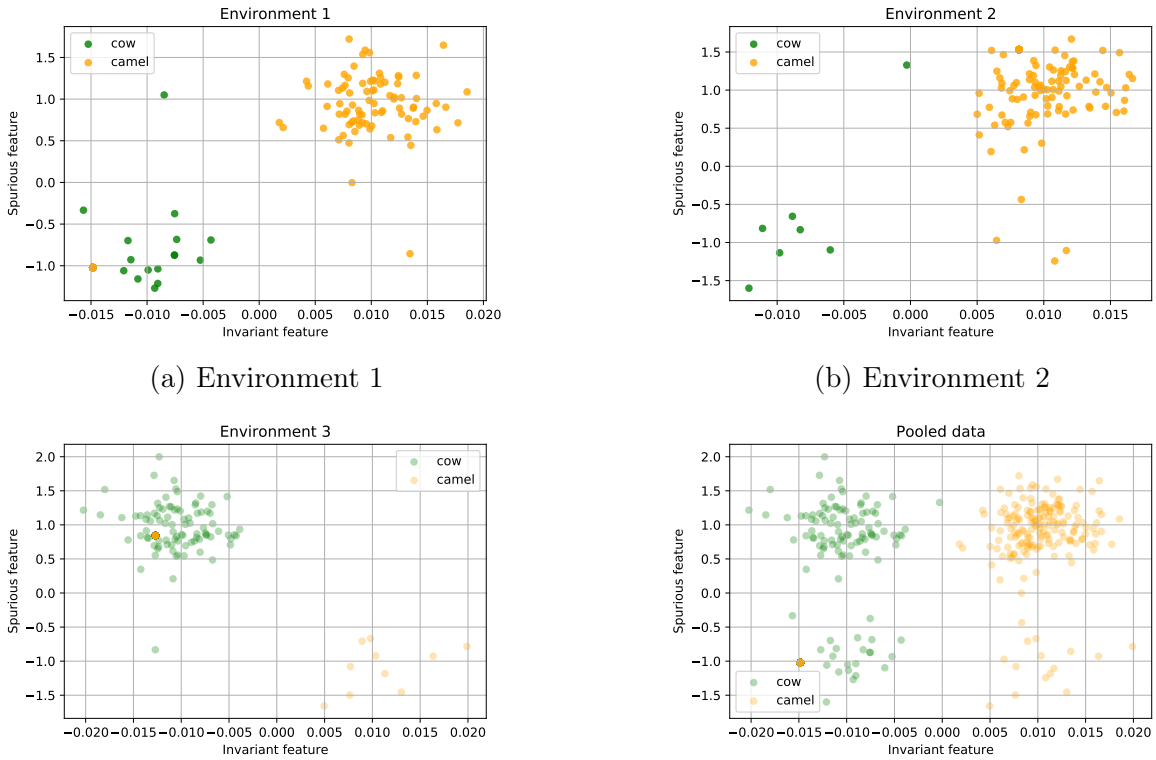
$$j_i^e \sim \mathcal{C}(p^e s^e, (1-p^e)s^e, p^e(1-s^e), (1-p^e)(1-s^e));$$

$$\boldsymbol{x}_{\text{inv},i}^e \sim \begin{cases} (\mathcal{N}_{d_{\text{inv}}}(0, 10^{-1}) + \boldsymbol{\mu}_{\text{cow}}) \cdot \nu_{\text{animal}} & \text{if } j_i^e \in \{1,2\}, \\ (\mathcal{N}_{d_{\text{inv}}}(0, 10^{-1}) + \boldsymbol{\mu}_{\text{camel}}) \cdot \nu_{\text{animal}} & \text{if } j_i^e \in \{3,4\}, \end{cases}$$
$$\boldsymbol{x}_{\text{spu},i}^e \sim \begin{cases} (\mathcal{N}_{d_{\text{spu}}}(0, 10^{-1}) + \boldsymbol{\mu}_{\text{grass}}) \cdot \nu_{\text{background}} & \text{if } j_i^e \in \{1,4\}, \\ (\mathcal{N}_{d_{\text{spu}}}(0, 10^{-1}) + \boldsymbol{\mu}_{\text{sand}}) \cdot \nu_{\text{background}} & \text{if } j_i^e \in \{2,3\}, \end{cases}$$
$$\boldsymbol{x}_i^e \leftarrow (\boldsymbol{x}_{\text{inv},i}^e, \boldsymbol{x}_{\text{spu},i}^e); \qquad y_i^e \leftarrow \begin{cases} 1, & \text{if } \mathbf{1}_{d_{\text{inv}}}^\top \boldsymbol{x}_{\text{inv},i}^e > 0, \\ 0, & \text{else;} \end{cases}$$

In this definition, background features are fully spurious, and animal features are fully invariant. Learning invariant features is more difficult than absorbing spurious features (see note below), yet it is more desirable, as an invariant predictor will correctly classify cows and camels both in the pooled grass and sand examples, and in each of the environments respectively.

This problem is highly relevant to out-of-distribution generalisation algorithms due to the following facts:

1. Achieving zero training error while using only $\boldsymbol{x}_{\text{inv}}^e$ is difficult because of small numbers and small distance between the means of cow and camel distributions (due to the scaling factor ν_{animal}). This can lead to learning large weights in neural networks. Gradient descent with most forms of regularisation penalises large weights: for example, L2



(a) Environment 1

(b) Environment 2

(c) Environment 3

(d) Pooled data

Figure 3.4: Illustration of the cows and camels challenge in two dimensions.
In this instance of cows and camels, there is one spurious feature (that represents background) and one invariant feature (that represents foreground; the actual animal). The scale of the spurious feature is of an order of magnitude larger than the scale of the invariant feature, which makes it easier to fit a spurious classifier.

regularisation for a loss function ℓ optimises an objective function F defined as follows:

$$F(\boldsymbol{w}) = \frac{1}{N}\sum_{i=1}^{N} \ell(h(\boldsymbol{x}_i; \boldsymbol{w}), y_i) + \lambda\|\boldsymbol{w}\|^2,$$

meaning that large weights amplify the objective function.

2. The probability of achieving zero training error using only $\boldsymbol{x}^e_{\text{spu}}$ increases rapidly with the increasing number of spurious features. Invariance penalties based on training error will learn spurious features.

3.3.4 Small invariant margin

In binary classification, a model learns a decision boundary, that is a hypersurface partitioning the data space into two sets. The smallest perpendicular distance between any of the data points and the decision boundary is referred to as the *margin* [26]. Support Vector Machines are based on maximising the margin.

This section presents the challenging problem of learning an invariant small-margin decision boundary in the presence of a spurious large-margin decision boundary.

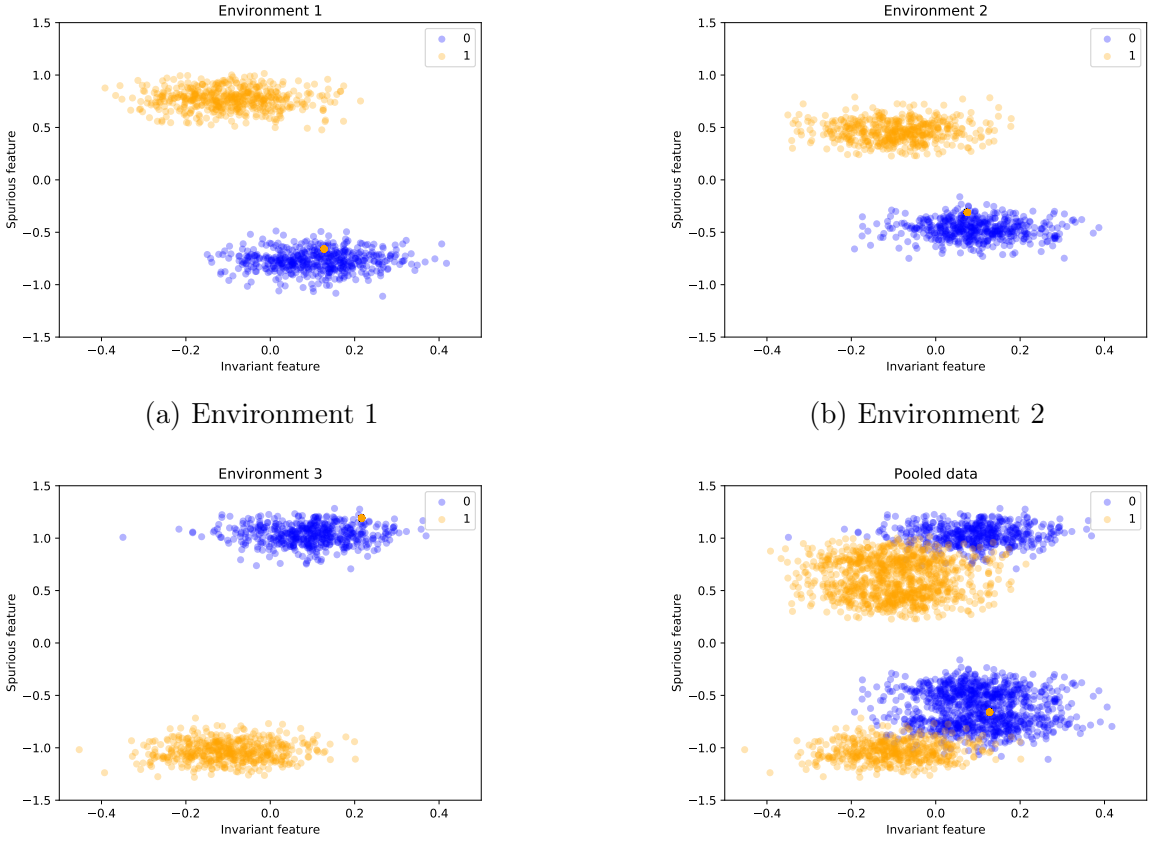
Let $\boldsymbol{\gamma} = 0.1 \cdot \mathbf{1}_{d_{\text{inv}}}$ and $\boldsymbol{\mu}^e \sim \mathcal{N}_{d_{\text{spu}}}(0, 1)$ for all environments. From the property of normal random variables (sum of squares), we expect that $\|\boldsymbol{\mu}^e\| > \|\boldsymbol{\gamma}\|$ in each environment e – therefore, the spurious solution has a larger margin than the invariant solution in each environment.

The datasets D_e are sampled for every environment $e \in \mathcal{E}$ and each data point $i = 1, \ldots, n_e$:

$$\begin{aligned}
y^e_i &\sim \text{Bernoulli}\left(\frac{1}{2}\right), \\
\boldsymbol{x}^e_{\text{inv},i} &\sim \begin{cases} \mathcal{N}_{d_{\text{inv}}}(+\boldsymbol{\gamma}, 10^{-1}) & \text{if } y^e_i = 0, \\ \mathcal{N}_{d_{\text{inv}}}(-\boldsymbol{\gamma}, 10^{-1}) & \text{if } y^e_i = 1; \end{cases} \qquad \boldsymbol{x}^e_i \leftarrow (\boldsymbol{x}^e_{\text{inv},i}, \boldsymbol{x}^e_{\text{spu},i}). \\
\boldsymbol{x}^e_{\text{spu},i} &\sim \begin{cases} \mathcal{N}_{d_{\text{spu}}}(+\boldsymbol{\mu}^e, 10^{-1}) & \text{if } y^e_i = 0, \\ \mathcal{N}_{d_{\text{spu}}}(-\boldsymbol{\mu}^e, 10^{-1}) & \text{if } y^e_i = 1; \end{cases}
\end{aligned}$$

An example of such datasets, using three environments, is given in Figure 3.5.

This is also challenging for existing OOD generalisation algorithms. We can solve this problem to zero training error with high probability using spurious features alone. Things are additionally complicated because solving this task using only the invariant features necessarily results in a small amount of training error. Hence, learning algorithms should learn to sacrifice training error to realise that invariant features lead to the same maximum margin classifier across environments (even though the environment-dependent margin



(a) Environment 1

(b) Environment 2

(c) Environment 3

(d) Pooled data

Figure 3.5: Illustration of the small invariant margin challenge in two dimensions. A vertical decision boundary at $x = 0$ is invariant across all the environments. However, it is not possible to achieve training error equal to zero using this decision boundary, which makes the spurious horizontal decision boundaries more compelling for standard ERM-based methods.

based on the spurious features is larger). While the predictor based on invariant features is optimal in terms of worst-case out-of-distribution generalisation, it is not a causal predictor of the target y since it comes with an error.

3.3.5 Scrambled variations

In the three families of linear unit tests described in the previous sections, features are explicitly given. In practice, an algorithm observes a transformation of features, and it must infer whether the latent feature is spurious or invariant (for instance, in image recognition, an observed feature is a pixel, and a latent feature is a specific object).

In order to evaluate if an out-of-distribution generalisation algorithm can learn a data representation on top of discovering the invariance, I use a random *rotation matrix* $\boldsymbol{S} \in \mathbb{R}^{d \times d}$ to transform the features, such that $D^e = \{(\boldsymbol{S}^\top \boldsymbol{x}_i^e, y_i^e)\}_{i=1}^{n_e}$. Recall that a rotation matrix is an orthogonal matrix whose determinant is 1, and orthogonal matrices are used for mapping orthonormal bases to orthonormal bases.

Scrambled variations are created for each of the linear unit tests independently.

3.3.6 Experiments

This section presents the results of evaluating existing out-of-distribution generalisation algorithms on the proposed linear unit tests. We refer to the three main examples as `Example1`, `Example2`, `Example3` and their scrambled variants are `Example1s`, `Example2s`, `Example3s`, using the linear unit definitions from the previous sections. The implemented algorithms, and their short descriptions, are as follows:

- *Empirical Risk Minimization* (ERM) [279] minimises the error on the union of all the training splits.
- *Invariant Risk Minimization* (IRM) [7] finds a representation of the features such that the optimal classifier, on top of that representation, is the same function for all environments. A practical formulation of IRM, used in this evaluation, is IRMv1 (from Arjovsky's paper cited above).
- *Inter-environmental Gradient Alignment* (IGA) [144] minimises the error on the training splits while reducing the variance of the gradient of the loss per environment.
- *AND-mask* [210] minimises the error on the training splits by updating the model on those directions where the sign of the gradient of the loss is the same for most environments.

Ex.Env	Algorithm						
	ANDMask	ERM	IGA	IRMv1	SD	CLRG	Oracle
Example1.E1	0.11 ± 0.04	1.62 ± 0.60	4.47 ± 1.16	0.20 ± 0.04	0.14 ± 0.00	2.49 ± 0.06	**0.05 ± 0.00**
Example1.E2	**11.39 ± 0.18**	14.25 ± 1.52	18.46 ± 2.14	**11.98 ± 0.75**	23.28 ± 0.11	29.27 ± 0.66	**11.27 ± 0.17**
Example1.E3	**20.28 ± 0.30**	24.22 ± 2.34	29.48 ± 3.19	**21.27 ± 1.34**	31.75 ± 1.11	40.23 ± 0.87	**19.93 ± 0.31**
Example1s.E1	0.07 ± 0.01	1.61 ± 0.59	4.55 ± 1.79	0.19 ± 0.04	0.17 ± 0.00	2.51 ± 0.03	**0.05 ± 0.00**
Example1s.E2	**12.13 ± 0.80**	14.23 ± 1.49	18.68 ± 3.37	11.92 ± 0.69	22.47 ± 0.13	29.08 ± 0.49	**11.24 ± 0.19**
Example1s.E3	**21.52 ± 1.42**	24.14 ± 2.39	29.81 ± 4.78	**21.08 ± 1.31**	30.40 ± 0.42	40.80 ± 0.75	**20.06 ± 0.37**

Table 3.1: Out-of-distribution regression error on the *regression from causes and effects* problem (`Example1-Example1s`) with five spurious features and five invariant features. Errors (Mean Squared Errors) are reported for all algorithms and three environments (E1, E2, E3). Average errors and standard deviations are computed using 50 independent runs. The lowest errors are written in bold.

Ex.Env	Algorithm						
	ANDMask	ERM	IGA	IRMv1	SD	CLRG	Oracle
Example2.E1	0.42 ± 0.02	0.40 ± 0.01	0.43 ± 0.00	0.43 ± 0.00	0.43 ± 0.00	0.17 ± 0.16	**0.00 ± 0.00**
Example2.E2	0.49 ± 0.03	0.47 ± 0.01	0.50 ± 0.00	0.50 ± 0.00	0.50 ± 0.01	0.20 ± 0.18	**0.00 ± 0.00**
Example2.E3	0.42 ± 0.02	0.40 ± 0.01	0.42 ± 0.01	0.42 ± 0.01	0.42 ± 0.01	0.17 ± 0.16	**0.00 ± 0.00**
Example2s.E1	0.43 ± 0.01	0.43 ± 0.01	0.43 ± 0.01	0.43 ± 0.01	0.47 ± 0.08	0.35 ± 0.08	**0.00 ± 0.00**
Example2s.E2	0.50 ± 0.00	0.50 ± 0.00	0.50 ± 0.00	0.50 ± 0.00	0.50 ± 0.01	0.41 ± 0.08	**0.00 ± 0.00**
Example2s.E3	0.42 ± 0.01	0.42 ± 0.01	0.42 ± 0.01	0.42 ± 0.01	0.47 ± 0.09	0.38 ± 0.04	**0.00 ± 0.00**
Example3.E1	0.35 ± 0.22	0.48 ± 0.09	0.47 ± 0.10	0.49 ± 0.07	0.50 ± 0.01	0.50 ± 0.01	**0.00 ± 0.00**
Example3.E2	0.36 ± 0.22	0.48 ± 0.07	0.48 ± 0.08	0.49 ± 0.06	0.48 ± 0.01	0.50 ± 0.01	**0.00 ± 0.00**
Example3.E3	0.32 ± 0.22	0.47 ± 0.12	0.46 ± 0.12	0.48 ± 0.07	0.50 ± 0.00	0.50 ± 0.00	**0.00 ± 0.00**
Example3s.E1	0.45 ± 0.13	0.48 ± 0.08	0.48 ± 0.09	0.49 ± 0.07	0.50 ± 0.01	0.52 ± 0.04	**0.00 ± 0.00**
Example3s.E2	0.49 ± 0.05	0.49 ± 0.05	0.48 ± 0.07	0.49 ± 0.06	0.50 ± 0.00	0.50 ± 0.00	**0.00 ± 0.00**
Example3s.E3	0.46 ± 0.12	0.47 ± 0.09	0.47 ± 0.11	0.48 ± 0.07	0.50 ± 0.00	0.50 ± 0.00	**0.00 ± 0.00**

Table 3.2: Out-of-distribution classification error on the *cows and camels* (`Example2-Example2s`) and *small invariant margin* (`Example3-Example3s`) problems.
Errors (accuracy) are reported for all algorithms and three environments (E1, E2, E3). Average errors and standard deviations are computed using 50 independent runs. The lowest errors are written in bold.

- *Spectral Decoupling* (SD) [214] is a regularisation method introduced to combat *gradient starvation* in neural networks (the phenomenon of only learning a few of the *easiest-to-learn* features [67]).
- *Constrained Linear Regression Game* (CLRG) [3] is a state-of-the-art OOD generalisation algorithm for linear regression games.
- *Oracle* is a version of ERM where all data splits contain randomised spurious features, which are therefore trivial to ignore. The purpose of this method is to understand the achievable *upper bound performance* in our problems.

Averaged results in Figure 3.6 and Figure 3.7 show that no method is able to match the Oracle's performance on all of the proposed problems. Table 3.1 and Table 3.2 show detailed results per environment.

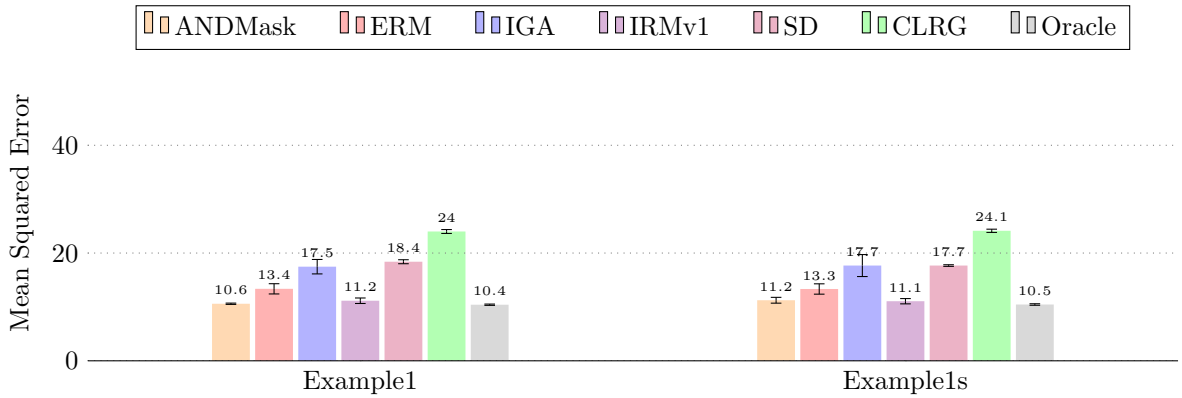


Figure 3.6: Regression errors (Table 3.1) averaged across environments.
The error bars (standard deviation for the estimate of the average MSE) are computed with the assumption that the environments are independent.
ANDMask and IRM are on a par with the Oracle's performance, which means that they are able to ignore the spurious features in *regression from causes and effects*. IRM is more robust to scrambling than ANDMask, most likely because this method learns an invariance-friendly representation of the features before it finds the model weights, while ANDMask relies on the original data representation.

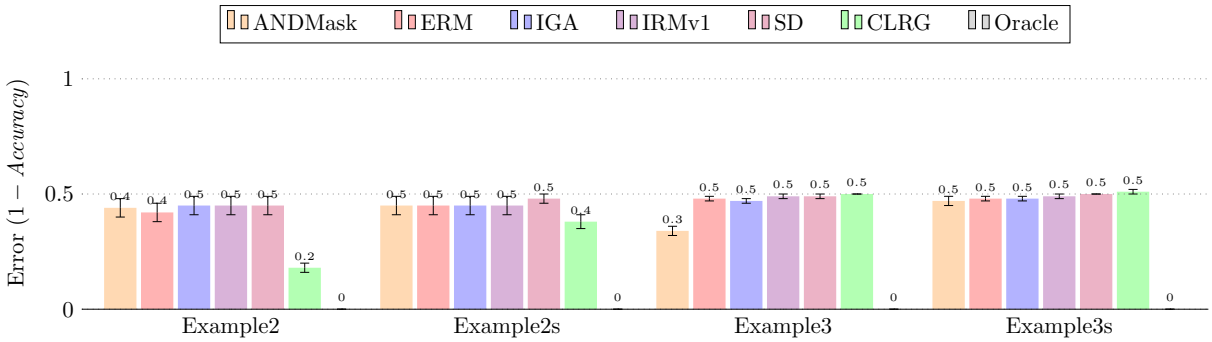


Figure 3.7: Classification errors (Table 3.2) averaged across environments.
The error bars (standard deviation for the estimate of the average classification error) are computed with the assumption that the environments are independent.
Performance of ANDMask is the closest to Oracle in `Example2` and `Example3`. However, the error increases on scrambled variations. It is expected given that ANDMask relies on the consensus in sign between gradients in different environments. In the scrambled variations (`Example2s`, `Example3s`) the link between basis vectors and the notion of invariance is broken, which renders them difficult for ANDMask.

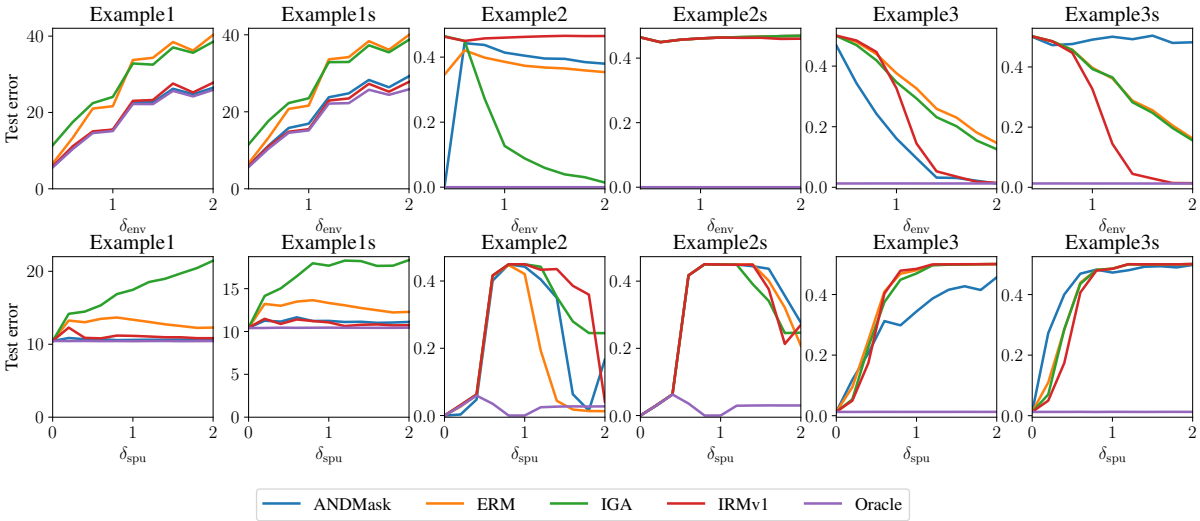


Figure 3.8: Test error averaged across environments for ANDMask, ERM, IGA, IRMv1 and Oracle on the unit tests as a function of the ratio $\delta_{\text{env}} = \frac{n_{\text{env}}}{d_{\text{spu}}}$ at fixed dimensions $(d_{\text{inv}}, d_{\text{spu}}) = (5, 5)$ (**top**) and as a function of $\delta_{\text{spu}} = \frac{d_{\text{spu}}}{d_{\text{inv}}}$ for $(d_{\text{inv}}, n_{\text{env}}) = (5, 3)$ (**bottom**). Figure produced by Benjamin Aubin.
All methods are sensitive to the number of environments and to the number of features, which makes Linear unit tests additionally challenging. The most promising methods based on the previous experiments (ANDMask and IRM) struggle when the number of environments increases. In the IRM solution, the risks become larger for the spurious features when n_{env} increases. In the ANDMask solution, the gradient consensus based on the spurious solution is less likely to be achieved when $n_{\text{env}} > d_{\text{spu}}$.

Discussion In *Regression from causes and effects* (`Example1`, `Example1s`), IRM and ANDMask achieve the lowest error up to the noise. IRM has been shown to exceed ERM on the classification variant of prediction from causes and effects (Colored MNIST [7]). A low error on the regression variant confirms that IRM can learn invariant features in the presence of spurious features that are caused by the target.

Classification problems (`Example2`, `Example2s`, `Example3` and `Example3s`) turn out to be more challenging in practice than the regression problem. Note that the random choice model obtains the accuracy of 50% in binary classification. In the *Cows and camels* problem (`Example2`, `Example2s`) the most promising algorithm turns out to be the recently proposed CLRG method. In the *Small invariant margin* problem, the only method which improves over random choice performance is ANDMask (Figure 3.7).

Linear unit tests are meant to be an initial stepping stone for evaluating and comparing the algorithms that aim to learn invariant features. The results shown here indicate that state-of-the-art out-of-distribution generalisation algorithms are unable to consistently learn robust features, even in low-dimensional linear problems.

Ablation studies Linear unit tests can be used to study the role of the number of environments n_{env} for fixed numbers of invariant and spurious dimensions. We define the ratio $\delta_{\text{env}} = \frac{n_{\text{env}}}{d_{\text{spu}}}$ to illustrate the relation between the number of environments and the number of spurious dimensions. The experiments are conducted for a varying number of environments $n_{\text{env}} \in [2 : 10]$ and a fixed number of spurious dimensions $d_{\text{spu}} = d_{\text{inv}} = 5$. The averaged test errors are shown in Figure 3.8 (top) for the algorithms ANDMask, ERM, IGA, IRMv1 and Oracle. On `Example1`-`Example1s`, both ANDMask and IRMv1 approach the perfect results obtained by Oracle, while on `Example2`-`Example2s` simple ERM outperforms them. On the contrary, ANDMask and IRMv1 achieve good performances on `Example3`. As expected, these algorithms approach an optimal solution for $n_{\text{env}} \approx d_{\text{spu}} + 1$. Moreover, while IRMv1 performances do not suffer due to scrambling, ANDMask collapses on `Example3s`.

The next experiment (Figure 3.8, bottom) uses a fixed number of environments $n_{\text{env}} = 3$ and a fixed number of invariant dimensions $d_{\text{inv}} = 5$ for a varying number of spurious dimensions $\delta_{\text{spu}} = \frac{d_{\text{spu}}}{d_{\text{inv}}}$. We observe that for `Example1`-`Example1s`, ANDMask and IRMv1 do not suffer because of the additional spurious dimensions, while IGA crumbles as soon a spurious feature is added. On `Example3`-`Example3s`, we observe that increasing the number of spurious dimensions with a fixed number of environments decreases the performances of all algorithms. `Example2`-`Example2s` show the same phenomenon for $\delta_{\text{spu}} \leq 1$.

Hyperparameters and implementation details In the results presented in this section, each algorithm is optimised using a random hyperparameter search of 20 trials with the optimisation procedure based on the popular 10^4 full-batch *Adam* [138] updates. The hyperparameters for each algorithm are chosen to minimise the error on the validation splits of all environments. Finally, we report the error of these selected models on the test splits, where the spurious links are destroyed (as defined in the *Generating linear units* paragraph Section 3.3). To provide error bars, this entire process, including data sampling, is repeated 50 times – 50 independently sampled problems $\times$ 20 evaluated hyperparameter sets. In all experiments, the number of samples per environment is equal to $n_e = 10^4$.

Summary of the results The proposed three problems prove challenging for existing algorithms for OOD generalisation. We notice that the classification tasks (`Example2`, `Example2s`, `Example3`, `Example3s`) are particularly difficult and no method consistently achieves a better performance than a fair coin flip on all problem variants and environments. While admittedly synthetic, this collection of problems covers a range of challenging distributional discrepancies that may arise across training and testing conditions. This work is a part of the modularity story (Chapter 1) through assuming multiple training distributions (analogous to incorporating the concept of a *module* at the data level) instead of a single monolithic training dataset.

Impact of Linear unit tests Linear unit tests have been widely used to evaluate and compare new methods that aim to improve out-of-distribution generalisation [248, 50, 136, 72, 171, 184, 49, 80, 283, 51, 64, 207]. Recent work by De Bartolomeis et al. [17] extends *Cows and camels* and *Small invariant margin* problems: among other contributions, the authors propose to rank the algorithms by how much weight is placed on the spurious features rather than by the OOD test error. The authors also show that the *Cows and camels* problem can be solved by using a high learning rate. Finally, the authors provide a theoretical analysis of why ANDMask was shown to fail at solving the linear *Small invariant margin* problem (Figure 3.7), and argue that this problem can be solved by ANDMask with a particular weight initialisation and a sufficiently large number of environments. Future work should explore this claim further.

3.4 Learning from multiple distributions and fairness

This section focuses on Distributionally Robust Optimisation (DRO), as defined in Equation (3.5), repeated here for clarity:

$$R^{\text{DRO}}(f) = \max_{e \in \mathcal{E}_{\text{tr}}} (R^e(f) - r_e)$$

Rather than focusing on the aspect of discovering invariant features as in the Linear unit tests section, this section is focused on another side of learning from multiple distributions: ensuring an acceptable performance across all training distributions (Figure 3.1). Given that in practice, these distributions might represent distinct gender and ethnic groups, this work is related to the contentious debate on the *data bias vs algorithmic bias* ingredients in the instances of machine learning bias observed in practice.

I first discuss DRO and its relation to data curation in Section 3.4.1. Afterwards, I state and prove two important theorems on the relation between the DRO solution and the ERM solution for a mixture of distribution in Section 3.4.2. Finally, I discuss the practical implications of the results of these theorems, especially how to avoid mistakes when using DRO in out-of-distribution generalisation problems, in Section 3.4.3.

3.4.1 DRO versus data curation

Recall that, traditionally, training a model in machine learning seeks parameters, such as the weights $\boldsymbol{w}$ of a neural network, that minimise a risk defined as the expectation of a loss function with respect to a *single distribution of training examples* (Equation (3.2)).

Alas, *even when the training distribution is representative of the actual testing conditions, the trained system might perform very poorly on selected subsets of examples.* For instance, Figure 3.1 describes a training problem where a majority population and a minority population have different classification boundaries. Minimising the expected loss over the full dataset (bottom left plot) yields a system whose performance is skewed towards the majority population at the expense of a random choice performance (48%) in the minority population.

Distributionally Robust Optimisation (DRO) *seemingly addresses this problem* by considering instead a collection of 'training distributions' and minimising the expected risk observed on the *most adverse* distribution (Equation (3.5)). For instance, using DRO with a set of two distributions representing the majority and minority populations leads to the classifier illustrated in Figure 3.1 (bottom right plot).

This viewpoint can be simplistic. For instance, minority groups pose a bigger challenge due to limited data accessibility (*representation disparity*) and bias amplification over time (*disparity amplification*) [192]. However, DRO remains an interesting building block because it provides a bridge between two common approaches to this problem, namely, 1) ensuring that the trained system has consistent performance across subpopulations, and 2) curating the training set by remixing the populations until a more acceptable result is obtained. I elaborate on these points later in the chapter.

DRO is commonly defined using the following notation [222]:

Definition 1. For a set of probability distributions $\mathcal{Q}$ and a corresponding family of cost functions $C_P(\boldsymbol{\Theta})$ over model parameters $\boldsymbol{\Theta}$, the *distributionally robust optimisation problem* is the problem of finding parameters $\widehat{\boldsymbol{\Theta}}$ that minimise the maximum risk across all distributions in $\mathcal{Q}$:

$$\widehat{\boldsymbol{\Theta}} = \arg\min_{\boldsymbol{\Theta}} \max_{P \in \mathcal{Q}} C_P(\boldsymbol{\Theta}) \tag{3.6}$$

In the basic version of DRO, we use the definition $C_P(\boldsymbol{\Theta}) = \mathbb{E}_{\boldsymbol{z}\sim P}[\ell(\boldsymbol{z};\boldsymbol{\Theta})]$ for any distribution P, where $\boldsymbol{z}$ is the data (both input and output) drawn from distribution P and $\ell(\boldsymbol{z};\boldsymbol{\Theta})$ is the loss for $\boldsymbol{z}$ given parameters $\boldsymbol{\Theta}$ (elsewhere we would write this $\ell(h(\boldsymbol{x};\boldsymbol{\Theta}), y)$ for $\boldsymbol{z} = (\boldsymbol{x}, y)$). Calibrated DRO instead uses the definition $C_P(\boldsymbol{\Theta}) = \mathbb{E}_{\boldsymbol{z}\sim P}[\ell(\boldsymbol{z};\boldsymbol{\Theta}) - r_P]$ for a set of *calibration coefficients* r_P.

For remainder of this chapter we assume that the family $\mathcal{Q}$ is finite, meaning $\mathcal{Q} = \{P_1, \ldots, P_K\}$ for some distributions $P_1, \ldots, P_K$.

3.4.2 Results

I present theoretical results that clarify the relation between finding a local minimum of the DRO problem and minimising the usual expected risk with respect to a single training distribution.

DRO has been suggested as a method for combatting bias and achieving out-of-distribution generalisation. The intuition is that we want to find a solution that works well across all environments. However, it turns out that any DRO solution is also a stationary point of an ERM solution for a probability distribution P_{mix} that is a *mixture* of the distributions in $\mathcal{Q}$: $P_{\text{mix}} = \sum_{k=1}^{K} \lambda_k P_k$, where $\sum_{k=1}^{K} \lambda_k = 1$ and all $\lambda_k \geq 0$. Formally, this correspondence is stated by the following theorem:

Theorem 1. *Let $\mathcal{Q} = \{P_1, \ldots, P_K\}$ be a finite set of probability distributions on $\mathbb{R}^n$ and let $\boldsymbol{\Theta}^*$ be a local minimum of the DRO problem $\min_{\boldsymbol{\Theta}} \{\max_{P\in\mathcal{Q}} \{C_P(\boldsymbol{\Theta})\}\}$. Let the costs $C_P(\boldsymbol{\Theta})$ be differentiable at $\boldsymbol{\Theta}^*$ for all $P \in \mathcal{Q}$. Then there exists a mixture distribution $P_{\text{mix}} = \sum_{k=1}^{K} \lambda_k P_k$ such that $\nabla C_{P_{\text{mix}}}(\boldsymbol{\Theta}^*) = 0$.*

Theorem 1 shows that when the collection of distributions $\mathcal{Q}$ is finite, under weak regularity assumptions, a DRO local minimum is always a stationary point of the expectation of the loss function with respect to a suitable mixture of the DRO training distributions. This result generalises that of Arjovsky [7]: previously KKT differentiability was assumed.

The proof of Theorem 1 relies on a lemma which generalises to n dimensions the following trivial fact on a line: for a closed interval $A \subseteq \mathbb{R}$, A either constains 0 or there is a number

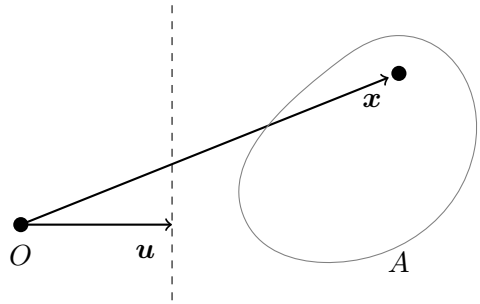


Figure 3.9: A graphical illustration of Lemma 1, in the case of $\mathbb{R}^2$.
If A does not contain the origin, there exist a vector $\boldsymbol{u}$ and a scalar c such that, for any point $\boldsymbol{x} \in A$, the inequality $\boldsymbol{u}^\mathsf{T}\boldsymbol{x} \geq c$. This means that there exists a hyperplane normal to $\boldsymbol{u}$ such that the origin is on the opposite side of the divided space as all points in A. In the $\mathbb{R}^2$ space, this separating hyperplane is a line.

$c > 0$ such that all $x \in A$ satisfy $c < |x|$. This hyperplane separation lemma is closely related to Farkas' lemma [31, Sec.2.5 and Ex.2.20].

Lemma 1. *A nonempty closed convex subset A of $\mathbb{R}^n$ either contains the origin or is strictly separated from the origin by a certain hyperplane, that is, there exists a vector $\boldsymbol{u} \in \mathbb{R}^n$ and a scalar $c > 0$ such that, for all $\boldsymbol{x} \in A$, $\boldsymbol{u}^\mathsf{T}\boldsymbol{x} \geq c$.*

Proof. Assume $\mathbf{0} \notin A$. Let $\boldsymbol{u} \in A$ be the projection of the origin onto the closed convex set A. For all $\boldsymbol{x} \in A$ and all t, $0 \leq t \leq 1$, the point $\boldsymbol{r}(t) = \boldsymbol{u} + t(\boldsymbol{x} - \boldsymbol{u})$ also belongs to the convex set A. Since $\boldsymbol{u}$ is the point of A closest to the origin, we have $\|\boldsymbol{r}(t)\| \geq \|\boldsymbol{u}\|$ for all $t \in [0, 1]$. In other words:

$$\forall t \in [0,1] \quad \|\boldsymbol{r}(t)\|^2 = \|\boldsymbol{u} + t(\boldsymbol{x} - \boldsymbol{u})\|^2 = \boldsymbol{u}^2 + 2t\boldsymbol{u}^\mathsf{T}(\boldsymbol{x} - \boldsymbol{u}) + t^2\|\boldsymbol{x} - \boldsymbol{u}\|^2 \geq \|\boldsymbol{u}\|^2.$$

The derivative of $\|\boldsymbol{r}\|^2$ with respect to t, when evaluated at 0, is $2\boldsymbol{u}^\mathsf{T}(\boldsymbol{x} - \boldsymbol{u})$. Therefore $\boldsymbol{u}^\mathsf{T}\boldsymbol{x} \geq \boldsymbol{u}^\mathsf{T}\boldsymbol{u} = \|\boldsymbol{u}\|^2 > 0$ for any $\boldsymbol{x} \in A$. □

Proof of Theorem 1. Consider the convex hull of the $g_k = \nabla C_{P_k}(\boldsymbol{\Theta}^*)$ for $k = 1 \ldots K$ and call it A. By definition of convex hull, A is closed and convex. If A does not contain the origin $\mathbf{0}$, according to the lemma, there exist u and c such that $\forall \boldsymbol{x} \in A$, $\boldsymbol{u}^\mathsf{T}\boldsymbol{x} \geq c > 0$. From Taylor expansion, for all $t > 0$, moving from $\boldsymbol{\Theta}^*$ to $(\boldsymbol{\Theta}^* - t\boldsymbol{u})$ reduces all costs C_{P_k} by at least $tc + o(t)$. As a consequence, $\max_k C_{P_k}$ is also reduced by at least $tc + o(t)$, contradicting the assumption that $\boldsymbol{\Theta}^*$ is a local minimum. Hence A contains the origin, and by closure this means that there are non-negative mixture coefficients λ_k summing to one such that $\sum_{k=1}^K \lambda_k \nabla C_{P_k}(\boldsymbol{\Theta}^*) = \nabla_{\boldsymbol{\Theta}} C_{P_{\text{mix}}}(\boldsymbol{\Theta}^*) = 0$. □

A simple question to consider is whether the converse holds: do minima of weighted loss mixtures correspond to minima of the Calibrated DRO problem? The answer is *yes* – in

fact, the folowing theorem states that a local minimum $\boldsymbol{\Theta}^*$ of $C_{P_{\text{mix}}}$ is also a local minimum of Calibrated DRO (3.5) with calibration constants r_P equal to the costs $C_P(\boldsymbol{\Theta}^*)$.

Theorem 2 (Converse)**.** *Let $P_{\text{mix}} = \sum_k \lambda_k P_k$ be an arbitrary mixture of distributions $P_k \in \mathcal{Q}$. If $\boldsymbol{\Theta}^*$ is a local minimum of $C_{P_{\text{mix}}}$, then $\boldsymbol{\Theta}^*$ is a local minimum of the Calibrated DRO problem (3.5) with calibration coefficients $r_P = C_P(\boldsymbol{\Theta}^*)$.*

Proof. By contradiction, assume that $\boldsymbol{\Theta}^*$ is not a local minimum of (3.5), that is, for all $\epsilon > 0$ there exists $\boldsymbol{u}$ such that $\|\boldsymbol{u}\| < \epsilon$ and $\max_{P\in\mathcal{Q}} \{C_P(\boldsymbol{\Theta}^* + \boldsymbol{u}) - r_P\} < \max_{P\in\mathcal{Q}} \{C_P(\boldsymbol{\Theta}^*) - r_P\}$. Recalling our choice of r_P yields:

$$\max_{P\in\mathcal{Q}} \{C_P(\boldsymbol{\Theta}^* + \boldsymbol{u}) - C_P(\boldsymbol{\Theta}^*)\} < 0.$$

Since $C_P(\boldsymbol{\Theta}^* + \boldsymbol{u}) < C_P(\boldsymbol{\Theta}^*)$ for all $P \in \mathcal{Q}$, we conclude $C_{P_{\text{mix}}}(\boldsymbol{\Theta}^* + \boldsymbol{u}) < C_{P_{\text{mix}}}(\boldsymbol{\Theta}^*)$. Thus $\boldsymbol{\Theta}^*$ cannot be a local minimum of $C_{P_{\text{mix}}}$. □

Theorems 1 and 2 establish a practical duality between finding a solution to the Calibrated DRO problem and the ERM solution for a given set of mixture coefficients. I argue calibration coefficients r_P are a more useful way of describing the difficulties of accurately classifying a subpopulation than the mixture coefficients, and the following section gives a practical use and interpretation of calibration coefficients. However, there is a discrepancy between the statements of these two theorems: Theorem 1 only provides a stationary point, whereas Theorem 2 requires a local minimum of the expected loss mixture. This distinction disappears if we assume the loss function $\ell(\boldsymbol{z}; \boldsymbol{\Theta})$ is convex in $\boldsymbol{\Theta}$ – all stationary points become global minima – so the two theorems then provide an exact equivalence between the two problems.

3.4.3 Practical recommendations

How can the result above influence the way DRO is calibrated in practice? From a mathematical perspective, one interesting approach is to choose, for each distribution P, a calibration constant r_P^* that represents the best performance we can reach with our machine learning model on this distribution in isolation:

$$r_P^* = \min_{\boldsymbol{\Theta}} C_P(\boldsymbol{\Theta}). \tag{3.7}$$

In practice, in order to estimate r_P^*, we might have to use regularisation terms in order to counter the effects of finite training data. Such regularisation terms could even make use of the training data for other populations or distributions. Let $\boldsymbol{\Theta}^*_{\text{DRO}}$ be a solution of the Calibrated DRO problem using these particular calibration constants. Because

Algorithm 1: Obtaining a DRO solution in practice.

Data: Training sets D_k for $k = 1 \dots K$.

Data: Minimum cost obtained for each D_k alone, i.e. r_k.

Data: Stopping parameter ϵ.

`// ϵ is the 'acceptable' performance gap.`

Result: A parameter vector $\mathbf{\Theta}^*$.

$\lambda_1, \dots, \lambda_K \leftarrow 1/K$; $d \leftarrow 0$ `// d is the current iteration count.`

`// Loop invariant:` $\sum_{k=1}^{K} \lambda_k = 1$.

repeat

  $\mathbf{\Theta}^* \leftarrow \texttt{Optimize}_{\mathbf{\Theta}}(\sum_{i=1}^{K} \lambda_i (C_{P_i}(\mathbf{\Theta}) - r_{P_i}))$

  $worst\_risk \leftarrow \max_i \{C_{P_i}(\mathbf{\Theta}^*) - r_{P_i}\}$

  $best\_risk \leftarrow \min_i \{C_{P_i}(\mathbf{\Theta}^*) - r_{P_i}\}$

  $worst\_risk\_idx \leftarrow \arg\max_i \{C_{P_i}(\mathbf{\Theta}^*) - r_{P_i}\}$

  `// Identify the most vulnerable subpopulation.`

  if $worst\_risk - best\_risk > \epsilon$ **then**

    $\forall k \in \{1, \dots, K\}. \qquad \lambda_k \leftarrow \begin{cases} \frac{d+K}{d+K+1}\lambda_k & k \neq worst\_risk\_idx \\ 1 - \sum_{i \neq k} \frac{d+K}{d+K+1}\lambda_i & k = worst\_risk\_idx \end{cases}$

  end

  $d \leftarrow d + 1$

until $worst\_risk - best\_risk \leq \epsilon$

return $\mathbf{\Theta}^*$

of the definition of r_P^*, we know that $C_P(\mathbf{\Theta}^*_{\text{DRO}}) \geq r_P^*$. In other words, this particular formulation of DRO tries to construct a single machine learning system that performs almost as well on each distribution P as a dedicated machine learning system specifically trained for distribution P only. Whether the outcome is acceptable for the real-world problem depends on whether the best distribution-specific performances r_P^* are themselves acceptable.

Based on this observation and the underlying mathematics (Section 3.4.2), we formulate practical recommendations for machine learning engineers facing real-world bias issues.

Model-agnostic recommendations for solving DRO based on the new theorems

1. Define subpopulations in the available data.

2. Compute the best performance for each of these subpopulations in isolation.
3. Decide whether any subpopulation is at risk and possibly repeat steps 1–2 with improved models and investigate possible data issues.
4. Use the obtained minimum risks as calibration coefficients.
5. Use an iterative algorithm for obtaining a DRO solution, for example, Algorithm 1.
6. Perform an *a posteriori* analysis of the DRO solution to see if any vulnerable subpopulation was initially undiscovered.

We now elaborate on what these steps mean and entail:

Elaboration of steps

1. Metadata and domain knowledge can be used to partition the dataset into subpopulations. For instance, in a face recognition system, subpopulations might contain images of people representing distinct ethnicities.
2. Data available for minority subpopulations might be limited. In such cases, remaining subpopulations can be used as an auxiliary task or as a regulariser to improve performance on an individual subpopulation.
3. We now have K subpopulations, parameters for the trained subpopulation-specific models $\mathbf{\Theta}_1, \ldots, \mathbf{\Theta}_K$ and the lowest possible risk for each subpopulation $C_{P_1}(\mathbf{\Theta}_1)$, $\ldots, C_{P_K}(\mathbf{\Theta}_K)$. At this stage, the problem of deciding whether the lowest achievable risk is acceptable depends on the specific problem at hand, and it is up to the practitioners as it is not a mathematical problem.
4. If minimum cost that can be achieved using available data is acceptable, we can use it to equalize performance according to the Calibrated DRO framework. Calibration coefficients r_P per each subpopulation are going to be equal to the optimum expected risk for that subpopulation alone, $r_P = \min_{\mathbf{\Theta}} C_P(\mathbf{\Theta})$. We can also adjust the calibration coefficients to prevent overfitting to individual subpopulations. For n_P examples in a certain subpopulation P, the expected risk $C_P(\mathbf{\Theta})$ can be replaced by its empirical estimate $C_{P_n}(\mathbf{\Theta})$ augmented with a calibration constant $\frac{1}{\sqrt{n_P}}$ that decreases when the number n of training examples increases [240][a].
5. If the performance on the entire population is unsatisfactory, it might be because of data issues such as inconsistent classes across subpopulations, or due to an insufficient capacity of the model when applied to the entire population. One might need to increase the model capacity before applying it to the entire population. If

the performance is satisfactory and practical considerations are carefully understood, an iterative algorithm for obtaining a revised DRO solution can be used.

6. The examples where the model performs the worst should be examined for any consistent patterns, which might suggest they belong to a vulnerable subpopulation that was not discovered in the initial step. In this case, such subpopulation should be added to the set of subpopulations, and the whole process can be repeated.

[a]Note that Theorem 1 and Algorithm 1 do not assume an equal number of samples per population. A small number of samples n_P can lead to overfitting to the population P, which is partially mitigated by augmenting calibration coefficients with the $\frac{1}{\sqrt{n_P}}$ penalty.

Let us briefly turn to Algorithm 1. It is based on iteration until a sufficiently good solution is found. We introduce coefficients $\lambda_1, \ldots, \lambda_K$ that weigh the significance of individual subpopulations. These coefficients are initially all set to $1/K$. The objective to be optimised is a mixture of the risks for each subpopulation weighed by the corresponding λ_k. If the performance on some subpopulation significantly deviates from that of the best subpopulation, we adjust the coefficients so that the worst subpopulation's coefficient increases. This is done by incrementing the common denominator of all coefficients (initially K). This is repeated until the performance is sufficiently equalised. A variant of this algorithm was implemented to produce the DRO solution in Figure 3.1.

3.5 Summary

This chapter is focused on the framework of learning from multiple training distributions, regardless of the model choice. I motivate and define this framework in Section 3.1 and Section 3.2, respectively. I explore two facets of this framework: achieving out-of-distribution generalisation through discovering stable features (Linear units, Section 3.3) and ensuring consistent performance across training distributions (DRO recommendations, Section 3.4). As mentioned in the previous section, even if training and test distributions are close enough for us not to worry about out-of-distribution performance, there are significant risks associated with models of inconsistent performance across known distributions.

Specifically, 1) I present a standardised suite of linear unit tests for measuring out-of-distribution generalisation. I include results of the evaluation using these tests and state-of-the-art out-of-distribution generalisation algorithms. This work shows that the model behaviour which causes problems in computer vision or natural language processing occurs even in a linear case. 2) I present a clarification of the conditions under which a robust solution to multiple training distributions corresponds to optimising a mixture of distributions. Guided by the theoretical results, I propose a new set of recommendations for ensuring fairness in multiple distributions.

To summarise, the existing approaches to out-of-distribution generalisation in the framework of multiple training distributions have nuances which require careful consideration. In regression problems, IRM and ANDMask are promising even in the challenging problem of the regression from causes and effects. In classification, ANDMask and CLRG obtain the best performance, which is however still far from the perfectly invariant model that achieves 0 out-of-distribution error. DRO is a useful framework both for out-of-distribution generalisation and for ensuring consistent performance across available distributions. However, the best performance obtained using an algorithm for DRO can only be as high as the best performances for individual training distributions. In other words, in certain scenarios the only solution to improving fairness and OOD generalisation appears to be improving data quality and quantity.

Regarding the connection with the research hypothesis and the research questions posed in Chapter 1, the contributions in this chapter investigate *modularity* at the data level, by considering several disjoint training datasets with no i.i.d assumption. Section 3.4 provides an answer to the first research question stated in Section 1.2, namely the *What is the relation between the existing data-centric and algorithmic approaches to improving OOD generalisation and robustness?* question. The set of new theorems presented in Section 3.4 clarifies that under the typical assumption of a differentiable cost function, the most popular algorithmic approach (DRO) is equivalent to curating training data by setting appropriate weights, under the assumption of convex costs.

The next chapter approaches the thesis goal by introducing a new neural architecture and presenting new empirical results in image classification with neural networks, which sets the stage for the final contribution chapter focused on multi-agent communication using neural networks.

Chapter 4

Out-of-distribution generalisation in image classification

Humans are efficient learners. Once someone has seen a single example of a wampimuk, they know how to recognize a small wampimuk.
–Laura Ruis (2022)

Modern Convolutional Neural Networks [165] (Section A.1.2) are largely responsible for the renaissance of neural networks due to the vast amount of image data available online and the online competitions such as ImageNet [65] (among others). However, even the architectures that achieve in-distribution test accuracy of over 90% notoriously underperform or fail if there is a distribution shift between the training and test data [266, 74, 11, 7, 137, 20]. This has been studied in the context of generalisation across spatial transformations such as two-dimensional rotation and translation [266, 74, 11], generalisation to different viewpoints (for example, 'Can a network shown only the Ford Thunderbird from the front and the Mitsubishi Lancer from the side generalise to classify the category and viewpoint for a Thunderbird seen from the side?' [182]), reasoning about new object pairs in the context of visual question answering [15], and generalisation with respect to new shape-colour combinations using toy datasets [7, 137]. In Chapter 3 (**Learning from multiple distributions**) I mention the 'cows and camels' problem [20], which further illustrates how CNNs fail at out-of-distribution generalisation by using shortcuts such as image background, even when it is irrelevant to the classification task.

In Chapter 3, I approached the problem of out-of-distribution generalisation by exploiting

the fact that separate training datasets are formed by sampling from multiple distributions. In contrast, this chapter assumes a single training dataset, which is the most common assumption in image classification. Under this assumption, it proposes modifications to the existing architectures with the goal of improving out-of-distribution generalisation. I investigate several types of distribution shift, including the one studied in Chapter 5 in the context of multi-agent games – *compositional generalisation*, where a model is evaluated on new combinations of familiar features (for example, a red circle might be present in the test examples but not seen during training, but red squares and blue circles are present in the training set).

The goal of this thesis is to improve the understanding and the results of out-of-distribution generalisation across several domains where machine learning is used. This chapter seeks the same goal, from the perspective of image classification.

The main research hypothesis of this chapter is:

> Multi-level feature aggregation improves out-of-distribution generalisation in image classification.

This hypothesis is explored using two different approaches to multi-level feature aggregation: dilated convolutions [298] (Section 4.1.1) and **Neural Function Modules** [159] (Section 4.1.2).

The main research objectives of this chapter are:

- to introduce Neural Function Modules and to provide evidence on their impact on Convolutional Neural Networks in out-of-distribution generalisation tasks;
- to provide evidence on the impact of using dilated convolutions in out-of-distribution generalisation tasks.

The results show that one of the proposed designs (Neural Function Modules) is beneficial in a variety of image classification tasks: compositional generalisation in visual question answering (including relational and non-relational questions; Section 4.2.1), the previously unsolved same/different OOD problem (Section 4.2.2), generalisation with respect to the changes in fonts, scale, rotation, translation, and compositional generalisation with respect to scale-rotation combinations in a single-object classification task (Section 4.2.3), and finally generalisation to a new number of objects in a multi-object classification task (Section 4.2.4).

Chapter structure

Section 4.1 provides a detailed description of two design choices, dilated convolutions and Neural Function Modules (NFMs), which are promising from the perspective of improving out-of-distribution generalisation in image classification. Dilated convolution is an existing concept that was previously used in multi-scale context aggregation [298]. This section revisits it from the perspective of the new hypothesis:

> Dilated convolutions can improve out-of-distribution generalisation in Convolutional Neural Networks.

Neural Function Modules are a new method, which is also investigated here from the angle of out-of-distribution generalisation in Convolutional Neural Networks.

Section 4.2 shows the results of extensive experiments using dilated convolutions and Neural Function Modules on four sets of OOD problems in image classification. The design choices proposed in this chapter are incorporated into various convolutional architectures (a vanilla CNN written from scratch as well as existing, commonly used architectures such as ResNet [106]). The datasets used in this section cover a variety of image classification applications: visual question answering, single-object classification and multi-object classification. This work extends my previous work on NFMs and dilated convolutions by providing new evidence in the setting of out-of-distribution generalisation. In several datasets, the results of the NFM-augmented architectures improve over the best results published on these datasets to date (Section 4.2.2 and Section 4.2.3).

Related publications The main work for this chapter – designing new experiments and implementing them, producing all of the results included in Section 4.2, revisiting the proposed methods in the context of out-of-distribution generalisation – I conducted independently for the purpose of this thesis.

The first architecture (Section 4.1.1) is based on a relational reasoning architecture from my Master thesis (**Relational reasoning with neural networks** at the University of Edinburgh) which was based on a common DenseNet [119] architecture – this chapter shows new results of my individual work in the context of out-of-distribution generalisation. I co-authored the second method (Section 4.1.2) in a collaboration led by Alex Lamb [159]. In this chapter, I show the results of my individual work based on my contribution to the NFM paper.

4.1 Architectures

This section presents the architectural choices investigated here as potential improvements over existing Convolutional Neural Networks in terms of out-of-distribution generalisation in image classification. Both of these design choices – dilated convolution and Neural Function Modules – can be incorporated into plain CNNs as well as into the existing complex and specialised architectures with a CNN component (for example, Relation Networks [241]).

In Section 4.1.1, I describe Dense Dilated Convolutional Neural Networks, or Dilated DenseNets. The second architecture, described in Section 4.1.2, is Neural Function Modules.

4.1.1 Dilated convolutional networks

Out-of-distribution generalisation in the context of image processing requires understanding both local features (such as pixel intensities) and global features (such as semantic interpretation of the image content). If a model can understand a scene (for instance, the spatial relations between semantically different segments of an image), it is more likely to generalise with respect to the properties of individual objects, such as colour, shape and size. Generalisation with respect to human-interpretable properties of an image can make the models more robust to real-life changes in the data, such as changing product trends that currently affect image-based recommender systems [107].

Convolutional neural networks are commonly seen as the go-to architecture when it comes to interpreting local and global features in an image. *Dilated convolutional neural networks* generalise them by utilising a generalisation of the convolution operation. The remainder of this section first described dilated convolutions, and then proceeds by describing Dilated Dense Convolutional Neural Networks.

4.1.1.1 Dilated convolution

A *dilated convolution* [298] is a more general form of the standard discrete convolution (Equation (A.2)):

$$s[t] = (f * g)(t) = \sum_{x=-\infty}^{\infty} f[x]g[t - lx] \tag{4.1}$$

In a dilated convolution, kernel weights are applied to the input only every l^{th} value along both axes, where l is the dilation factor ($l = 1$ for a standard convolution). We use the term l-convolution to refer to the variant seen in Equation (4.1). The extension to multidimensional data is straightforward.

Traditional convolutions have a *receptive field* (number of inputs that influence a particular value in the output) of the same size as the convolution kernel, meaning that only the immediate neighbourhood of an input influences the value of the output. A dilated convolution has the same receptive field, except the inputs are more 'spread out'. Dilated convolutions enter practical use if an input signal f_0 is initially 1-convolved with a given kernel k to obtain f_1, after which f_1 is 2-convolved with k to obtain f_2, then f_2 is 4-convolved with k to obtain f_3 – with the dilation factor doubling at each step. It is easily seen that for a 2d kernel k of size 3×3, the receptive field of the output f_i consists of $(2^{i+1} - 1) \times (2^{i+1} - 1)$ values in f_0.[1] This enables a balance between processing pixel-level information and integrating the wider context (such as object arrangement, image background, etc.), without increasing the number of model parameters. An illustration of this procedure is seen in Figure 4.1.

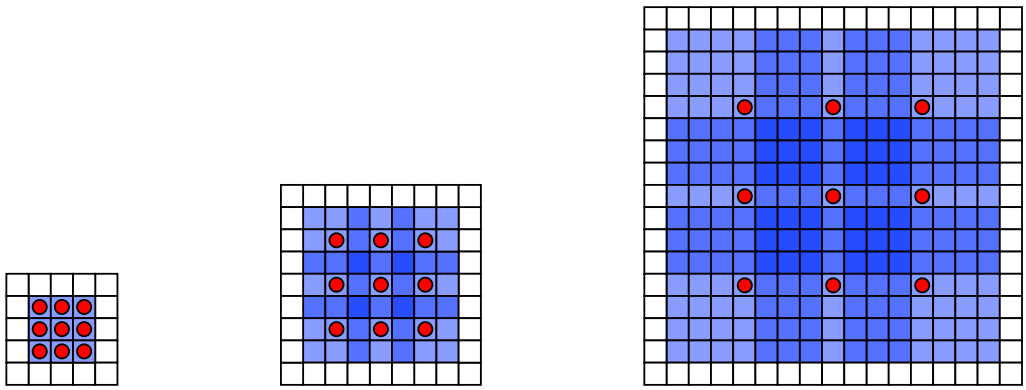
In the context of multi-scale context aggregation in image segmentation tasks [298], dilated convolutions are successfully used in the form of an *exponential schedule of dilated convolutions*. In such case, the dilation factor l increases exponentially with the number of layers ($l = 1, 2, 4, ...$). This means that the first convolutional layer in the network performs a standard convolution which detects local features. Consecutive layers employ increasingly sparse filters that integrate the image context at an increasingly larger scale.

Dilated convolutions were experimentally shown to improve various instances of semantic segmentation tasks including modeling of long-distance dependencies [99] [268], the extraction of very small objects [101], and integration of local and global semantics [6]. This modification to the standard convolutional layer does not introduce additional trainable parameters, and it should allow for a better understanding of the scene due to the combination of extracting the local and global features. Here, the questions is: *do dilated convolutions improve out-of-distribution generalisation, compared to regular convolutions?*

4.1.1.2 Dilated DenseNet

The Dense Convolutional Network (DenseNet) [119] is one of the network topologies commonly used in CNNs. It is a powerful family of architectures that encourage feature reuse, which might be beneficial for out-of-distribution generalisation. A DenseNet contains structural units called *blocks*, where each layer is connected to every other layer in place of the standard one-to-one connections in adjacent layers in a neural network. The input to a single layer consists of a concatenation of the outputs of all the preceding layers, which means that the model preserves the features extracted by individual layers. A single dense block is shown in Figure 4.2.

[1] Contrast this with stacking i 1-convolutions with a 3×3 kernel, which results in a receptive field of size $(2i + 1) \times (2i + 1)$.



(a) $l = 1$, $i = 1$ (b) $l = 2$, $i = 2$ (c) $l = 4$, $i = 3$

Figure 4.1: Illustration of three dilated convolutions for a base 3×3 kernel and an exponential schedule. Source: https://www.inference.vc/dilated-convolutions-and-kronecker-factorisation/.
The red circles represent the pixels in f_{i-1} that the central pixel of f_i depends on. The shading refers to the significance of each pixel in f_0 (original image) for determining the value of the central pixel in f_i.
This shows that, with the stacking approach, the immediate neighbourhood of a pixel is still more significant in determining the final output than more distant pixels.

Dilated convolutions can in principle replace standard convolutions in any architecture. DenseNet is an example of a strong baseline of a modular structure (based on blocks) which is convenient in an implementation of the exponential schedule of dilated convolutions. In a Dilated DenseNet, the exponential schedule of dilated convolutions is used independently in each of the dense blocks. As a result, each block integrates both local and global information. The blocks themselves are connected in a standard feed-forward way with one-to-one connections.

Here, a DenseNet containing three blocks is used as a fixed baseline compared with a Dilated DenseNet, having the same architecture but with the convolutional layers replaced by the exponential schedules of dilated convolutions. In my previous work (Master thesis and a workshop paper [6]), the Dilated DenseNet was proposed in the context of relational reasoning as an alternative to Relation Networks. In this chapter, I use a more general variant of DenseNets/Dilated DenseNets that can learn any image classification task, and compare these two models in the context of out-of-distribution generalisation. The premise is that a standard convolutional architecture (for example, a DenseNet) might overfit to the local information in an image, and adding dilational convolutions (for example, a Dilated DenseNet) should help in learning a wider, more abstract context that might help in generalising to out-of-distribution examples.

Apart from relational reasoning [6], variants of Dilated DenseNets have been previously used

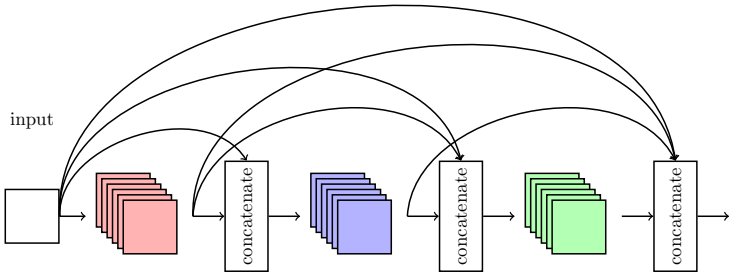


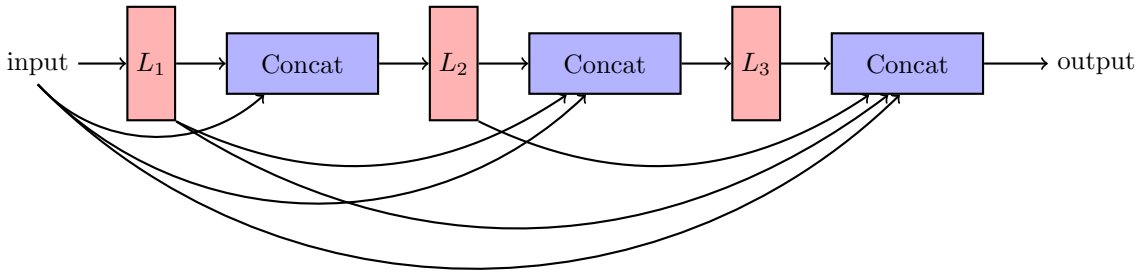
Figure 4.2: An illustration of the operation of a single 3-layer block in a Dense Convolutional Neural Network.
The stacked squares represent convolutional layers. The boxes labeled 'concatenate' construct concatenated vectors out of all the input vectors. Each concatenation in this sequence considers the input image and the output of all of the previous convolutional layers. As a result, the architecture supports multi-level feature aggregation.

in domains such as audio and speech processing [274] and macromolecule classification [83]. Here, the method is seen as on of the ways to endow Convolutional Neural Networks with the ability to aggregate local and global features. This method is then investigated in the context of out-of-distribution generalisation in image classification (Section 4.2).

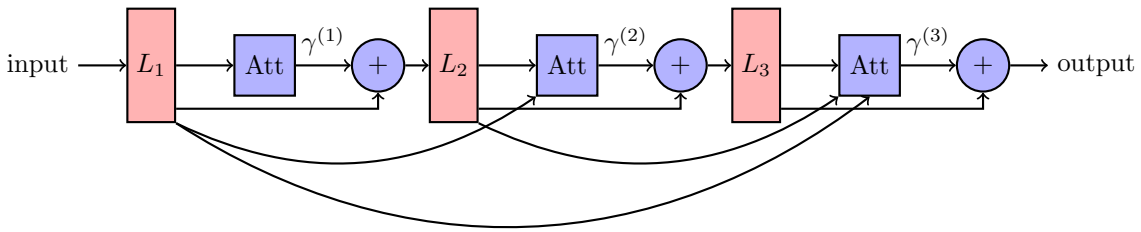
4.1.2 Neural Function Modules

Similarly to dilated convolutions, Neural Function Modules (NFMs) [159] are a design choice that can be incorporated into existing architectures. While dilated convolutions aim to integrate the local and global context, NFMs aim to increase the specialisation of layers in a neural network, by mimicking the concept of *subroutines* used in programming languages: rather than applying an operation to the entire hidden state of a network, the operation can be performed on its subsets. This is in contrast to the standard practice of designing feed-forward neural networks as a sequence of layers, in which each layer processes the entire output of the antecedent layer (Figure A.2).

For example, let a program state consist of the variables a, b, c, d, e, f, g. According to the logic of programming languages, a subroutine may be applied to appropriate actual parameters to introduce a new variable or otherwise modify the program state – for instance, for a two-parameter function z, the instruction $x = z(a, f)$ leads to a new program state with variables x, a, b, c, d, e, f, g; we may also mutate e with $e = z(b, c)$. Advantages of this logic are: (1) it allows the programmer to avoid overwriting variables unrelated to the currently used function parameters, (2) it is easier to track down mistakes to changes in specific variables, (3) the programmer can use either the recently computed



(a) 3-layer dense block in a DenseNet



(b) 3-layer NFM

Figure 4.3: Illustration of how Neural Function Modules are directly inspired by DenseNets. In a dense block of a DenseNet, the input and the layer output are passed to all subsequent layers. This means that the dimension of the input to L_{k+1} is not the same as the dimension of the output of L_k. Neural Function Modules avoid this issue by using attention blocks that attend over all previously computed representations. Concatenation is then replaced by a weighted sum of the output of L_k and the computed attention.
NFMs extend the idea of multi-level feature aggregation by allowing a dynamic choice of relevant information using attention mechanism.

variables, or the variables that were computed much earlier and remained in the scope of the program state.

In contrast, according to the logic of standard feed-forward neural networks (Figure A.2), a layer is seen as a function that gets applied to all of the arguments and overwrites the entire state. NFMs aim to incorporate the logic of subroutines from programming languages into existing neural networks by allowing each layer to decide which parts of the hidden state it processes.

In the DenseNet architecture described in the previous section, each layer takes as input all previous layers within the same block, which reduces the need to store redundant information in multiple layers, as a layer can directly use information from any prior layer within the block, and not just from the antecedent layer as in Figure A.2. NFMs expand on this idea and allow the layers to dynamically choose which past information to take. An NFM allows the layers to *attend* over the previously computed outputs to construct their input (Algorithm 2 and Figure 4.4), which is done using *attention mechanisms* (Section 2.3.2). In particular, NFMs use multi-headed attention [280] (Section 2.3.2.2).

The NFMs are inspired by multi-headed attention from the widely-used Transformer architecture [53]. The key differences between NFMs and Transformers are: (1) NFM is a design choice that can be used on top of any existing neural architecture, rather than a standalone architecture as is the case for Transformers, (2) The Transformer architecture uses attention over positions of the output of the antecedent layer only. In contrast, NFM attends over outputs of all the previous layers. The methods share the high-level motivation of allowing parts of a neural network to dynamically select their inputs using attention.

Apart from the specialisation/attention component, NFMs aim to implement the idea of using *top-down* feedback in the context of neural networks. Existing neural networks rely on the *bottom-up* feedback. For instance, in a feed-forward neural network in Figure A.2, the first layer processes the input, and each of the subsequent layers processes the output of its antecedent layer. In the bottom-up processing, the network does not have access to a previously computed representation of the input. An analogy can be drawn between the bottom-up feedback in a standard neural network and the biological ability to process a new visual signal for the first time.

Top-down feedback is inspired by cognitive psychology, where it refers to the brain's ability to use the contextual information of things that are already known in combination with our senses to perceive new information [92]. NFMs aim to implement top-down feedback using multiple *passes*. Each pass corresponds to running the original forward pass of the network

Algorithm 2: The forward pass of a Neural Function Module (NFM)

Data: Input $\boldsymbol{x}$.

Data: Number of passes K.

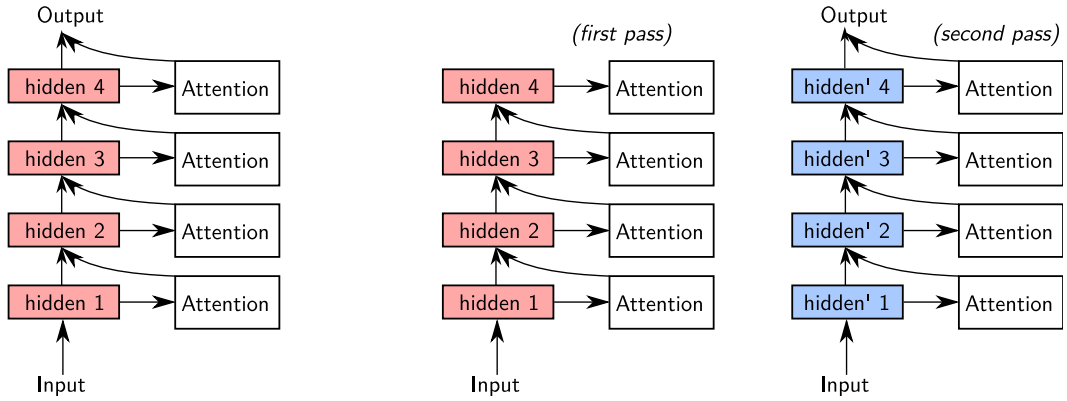
Data: A neural network with N layers for each $k \in \{1, \ldots, K\}$: $f^{(1)}_{\boldsymbol{\theta}_k}, f^{(2)}_{\boldsymbol{\theta}_k}, f^{(3)}_{\boldsymbol{\theta}_k}, \ldots, f^{(N)}_{\boldsymbol{\theta}_k}$

Data: Attention modules $\mathsf{Attention}^{(i)}_{\boldsymbol{\omega}_k}$ for each layer i and each pass k, where parameters $\boldsymbol{\omega}_k$ represent key, value and query embeddings (all together). These embeddings are followed by an attention function (in the implementation used in this thesis, this is multi-headed scaled dot-product attention described in Section 2.3.2.2).

Data: Trainable factors $\gamma^{(i)}_k$. They increase the significance of the output of the attention module $\boldsymbol{a}$ compared to the output of the previous layer $\boldsymbol{y}$.

Result: Final hidden state $\boldsymbol{h}^{(N)}$.

$\mathcal{M} \leftarrow \mathit{List.Empty}$ `// List of outputs of all previous layers`
for $k \leftarrow 1$ **to** K **do**
  $\boldsymbol{h}^{(0)} \leftarrow \boldsymbol{x}$
  for $i \leftarrow 1$ **to** N **do**
    $\boldsymbol{y} \leftarrow f^{(i)}_{\boldsymbol{\theta}_k}(\boldsymbol{h}^{(i-1)})$ `// Output of the` i`-th layer of the network`
    $\mathcal{M}.\mathit{append}(\boldsymbol{y})$
    $\widetilde{\mathcal{M}} \leftarrow \mathit{List.Empty}$
      `// List of outputs of previous layers, all scaled to have the same shape as` y `(so that keys and the query have compatible sizes)`
    foreach $\boldsymbol{m} \in \mathcal{M}$ **do**
      $\widetilde{\mathcal{M}}.\mathit{append}(\mathit{rescale}(\boldsymbol{m}, \text{shape} = \mathit{shapeOf}(\boldsymbol{y})))$
    end
    $\boldsymbol{a} \leftarrow \mathsf{Attention}^{(i)}_{\boldsymbol{\omega}_k}(\text{Keys} = \widetilde{\mathcal{M}}, \text{Values} = \widetilde{\mathcal{M}}, \text{Query} = \boldsymbol{y})$
      `// Attention function preceded by embeddings for keys, values and the query`
    $\boldsymbol{h}^{(i)} \leftarrow \boldsymbol{y} + \gamma^{(i)}_k \boldsymbol{a}$
  end
end
return $h^{(N)}$



(a) $K = 1$ (b) $K = 2$

Figure 4.4: A high-level illustration of the NFM principle.
Given a neural network with a feed-forward structure – here with four hidden layers – the 1-pass NFM (left) appends an attention module after each hidden layer. This makes the attention module act as a 'subroutine': it accepts different subsets of 'working memory' of the neural network, and thus aims to specialise the roles of layers and neurons.
Not shown in the diagram is the fact that the keys and values in each attention block attend over all of the previously seen hidden layer outputs.
The 2-pass NFM (right) has twice as many layers as the 1-pass: after the final hidden layer of the original network of four hidden layers, the output is discarded and the procedure is repeated. The copy of the original network in the second pass shares the weights with the original network from the first pass – however, in the second pass the keys and values fed into attention modules are based on the keys and values computed in the first pass. Note that NFM can take as input any feed-forward neural network, either written from scratch or one of the commonly used architectures as ResNet [106] (as demonstrated in Section 4.2). The official source code (`https://github.com/Slowika/NeuralFunctionModules`) contains an implementation of the attention block and an example of applying an NFM in a neural network.
Algorithm 2 includes a detailed description of each step.

that is augmented with an NFM, computing the attention weights, and then repeating this process while allowing the NFM to attend over all of the previous passes.

In Figure 4.4, the first pass corresponds to the 'bottom-up' feedback directly from the input, which is what most neural networks exclusively rely on. We can think of the output of each hidden layer as representing successive compressed representations of the input stimulus – starting from unstructured raw data, each layer of a neural network is a step towards getting to the final result, for example one of two classes in a binary classification problem. The second pass corresponds to 'top-down' feedback, because the information comes from the layers that already have a compressed representation of the entire input thanks to the hidden layer outputs from the previous pass forming the basis of the keys and values of the attention modules (Figure 4.4). The 2-pass variant of NFM allows attention modules to have a richer set of keys and values. Since in an NFM each layer has a choice

of outputs from not only all of the previous layers but also the previous passes, NFMs allow for a higher diversity of specialised layers in comparison to the existing architectures that promote layer specialisation, such as DenseNet [119] and Neural ODEs [47].

Algorithm 2 presents a detailed description of each step in a forward pass of an NFM-augmented neural network. The attention module implements multi-headed scaled dot-product attention described in Section 2.3.2.2. Trainable factors $\gamma_k^{(i)}$ were previously used in a similar way in Self-Attention Generative Adversarial Networks [304] to increase the importance of the output of the attention module. The *rescale* function uses either down-sampling or up-sampling on each previously seen output m so that all previously computed outputs are given in the same format. The internal structure of NFM (Algorithm 2) can remain the same regardless of the specifics of a neural network that is augmented with NFM.

The NFM is a promising idea from the perspective of handling distribution shifts due to the increased specialisation of the layers and the ability to dynamically choose which previous information to use. This is particularly interesting in the context of Convolutional Neural Networks and image classification. For instance, a convolutional layer which learns fine-grained visual features may be difficult to use early in the network, but its output might come in handy in deeper layers of the network, allowing the model to better understand an image by combining low-level and high-level features. In this way, NFMs are complementary to dilated convolutions (Section 4.1.1) as another design choice which allows multi-level aggregation of features. The fact that NFM can be used to augment any feed-forward neural network rather than constituting a standalone architecture makes it a great choice for the experiments that aim to answer research questions based on varying a single 'parameter' at a time (instead of comparing the results of using very different architectures, as described in the methodology section Section 2.2). Moreover, in the paper that introduces NFMs [159] we focus mostly on standard supervised learning, generative modelling and reinforcement learning, with little evidence in the context of distribution shifts. The next section extends my previous work on NFMs by providing new evidence in the context of out-of-distribution generalisation using four sets of OOD problems in visual question answering, single-object classification and multi-object classification.

4.2 Experiments

This section presents new results on the relation between design choices in convolutional architectures and the ability to generalise out-of-distribution. Specifically, it shows the results of using dilated convolutions and Neural Function Modules as two different ways of enhancing multi-level feature aggregation in various instances of image classification tasks:

visual question answering (including relational and non-relational questions; Section 4.2.1 and Section 4.2.2), single-object classification (letter recognition; Section 4.2.3) and multi-object classification (digit recognition; Section 4.2.4). In each task, the appropriate baselines are augmented with either NFM or dilated convolutions, in order to investigate the influence of these two architectural choices in isolation on out-of-distribution generalisation. In-distribution performance is also included for reference.

Research questions The main hypothesis of this chapter:

> Multi-level feature aggregation improves out-of-distribution generalisation in image classification

is explored using two approaches to multi-level feature aggregation: Dilated DenseNets (Section 4.1.1) and Neural Function Modules (Section 4.1.2).

The experiments in this section aim to explore this hypothesis in several instances of image classification by answering the following sets of research questions:

- **Visual question answering:** What is the effect of augmenting a baseline model with NFMs? What is the effect of augmenting a baseline model with dilated convolutions? Is there a systematic difference in the performance on the relational and the non-relational questions in a visual question answering task? (Section 4.2.1) Can NFMs or dilated convolutions improve the accuracy in the instance of a visual question answering task where only the random chance performance has been reported previously? (Section 4.2.2)
- **Single-object classification:** Can NFMs strengthen the existing baselines, in comparison with the published results and the reproduced results? Is there a systematic difference in the performance on the *compositional* OOD partitions (based on two features) and *stratified* OOD partitions (based on one feature)? (Section 4.2.3)
- **Multi-object classification:** Can NFMs improve the performance of a baseline model in generalisation from a smaller to larger number of objects in an image and vice versa? (Section 4.2.4)

Architectures In this section, experiments compare the performances of the same underlying architectures, with only different input lengths affecting the sizes of individual layers in these neural networks. All of the baselines are chosen based on the existing papers that use the same (Section 4.2.3) or related (Section 4.2.1, Section 4.2.2 and Section 4.2.4) datasets in order to have a point of comparison with the related work. 2-pass NFM is used to test the combination of bottom-up (the first pass) and top-down (the second pass) feedback in out-of-distribution scenarios. The following classifiers were used:

- CNN (or Conv_4): a generic 4-layer convolutional neural network, with ReLU and batch normalisation [125] layers following each convolutional layer (each of which has a stride of 2 and doubles the number of channels), with a final layer of average pooling, and a dropout rate of 0.1. This is the same architecture as used in the paper that introduces the in-distribution version of the Sort-of-CLEVR dataset [241] investigated in Section 4.2.1. As the most lightweight and generic convolutional neural network, this method and its NFM-augmented counterpart are used in each set of experiments in this chapter;

- CNN+NFM: just like CNN, but extended with a two-pass ($\mathcal{K} = 2$) NFM;

- ResNet_12: a residual network [106] of width 1, with five building blocks of three convolutional layers each (with batch normalisation) and 'shortcut' weights, with dropout rate 0.1. This method is used in the paper [151] that introduces the Synbols dataset investigated in Section 4.2.3;

- ResNet_12+NFM: ResNet12 with a two-pass NFM;

- WRN_28: a 28-layer wide residual network [299] with three groups of four residual blocks with two convolutions per block, with a dropout rate of 0.1 after the first convolution of each block (following Lacoste et al. [151], the paper that introduces the Synbols dataset used throughout Section 4.2.3);

- WRN_28+NFM: WRN_28 with an added two-pass NFM;

- Relational Networks: a 4-layer CNN followed by four fully-connected layers (with output sizes 256) that process all pairwise combinations of the CNN embeddings [241]. This is the same architecture as used in the papers that introduce the in-distribution version of the Sort-of-CLEVR dataset [241] and the Not-so-CLEVR dataset [231]. These two datasets are used in Section 4.2.1 and Section 4.2.2, respectively;

- Relational Networks+NFM: a relational network with an added two-pass NFM;

- DenseNet: a DenseNet [119] with growth rate 8, depth 16 (3 blocks with 4 layers each and 4 non-convolutional layers), reduction rate 0.5 and dropout 0.2. This method is used in my previous research on the effect of dilated convolutions in in-distribution relational reasoning [6];

- DilatedDenseNet: a Dilated DenseNet with the same parameters as DenseNet, with dilation parameters within each block being 2^i for the i-th convolutional layer (for $i \in \{0, 1, 2, 3\}$). This is an implementation of the exponential schedule of dilated convolutions presented in Figure 4.1.

The Neural Function Modules are all 4-headed, with key size and value size 16.[2] In order to answer the research questions posed in this section (Paragraph 4.2), the parameters described above are fixed and the design choices (NFM, dilated convolution) are the most meaningful axes of variation. The baseline architectures to be augmented with either NFM or dilated convolutions are based on their relevance to the datasets used in this chapter – that is, there are existing results using the same baseline with the same parameters on the same or related datasets. The baselines are also chosen such that they work well on the in-distribution variants of the datasets used in the chapter, in order to focus on the difficulty of moving to out-of-distribution evaluation.

In the following subsections, I describe the OOD generalisation experiments that were run in a variety of image reasoning contexts: visual question answering on the Sort-of-CLEVR dataset (Section 4.2.1); learning the 'same/different' concept in image reasoning, using the Not-so-CLEVR dataset and its variants (Section 4.2.2); letter and font recognition using the Synbols dataset (Section 4.2.3); multi-object classification in digit recognition, using a dataset derived from MNIST (Section 4.2.4). While describing the experiments, I first define the dataset, and then provide experimental results.

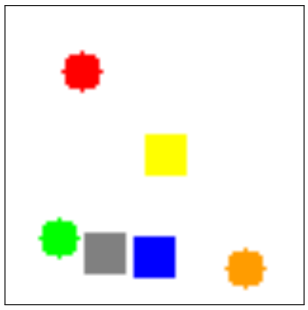
4.2.1 OOD generalisation in visual question answering

4.2.1.1 Data

The Sort-of-CLEVR dataset [241] was first introduced to probe Convolutional Neural Networks in their ability to answer relational questions such as 'What is the shape of the object closest to the red object?' based on an image (Figure 4.5). Each data sample consists of an image and a question regarding the image content. Each image has a total of 6 'objects', where each object has a randomly chosen shape (square or circle) and a randomly chosen colour (red, blue, green, orange, yellow, gray). The colour unambigously identifies the object.

The dataset contains 9800 distinct images in the training set and 200 distinct images in the test set. The training and test samples are generated by pairing each of these images with 10 relational and 10 non-relational questions. Similarly as in Santoro et al. [241], the questions are encoded as binary vectors of length 11. The first 6 values identify the colour of the object addressed in the question, the next 2 values encode the question type (relational or non-relational), and the last 3 values indicate the question subtype (1, 2 or 3). The object referred to in each question is chosen uniformly at random, so that each of the objects (identifiable by their colours) is mentioned an approximately equal number of

[2]Furthermore, the NFM implementation does not use the 'textbook' variant of scaled dot-product attention explained in this thesis – it uses a variant (Sparse Attention) based on the idea that the attention matrix is sparse. This improves overall NFM performance.



Relational questions:

1. What is the shape of the object closest to the red object? ⇒ square
2. What is the shape of the object furthest to the orange object? ⇒ circle
3. How many objects have same shape with the blue object? ⇒ 3

Non-relational questions:

1. What is the shape of the red object? ⇒ Circle
2. Is green object placed on the left side of the image? ⇒ yes
3. Is orange object placed on the upside of the image? ⇒ no

Figure 4.5: In Sort-of-CLEVR, a data sample consists of an image and a question to be answered based on a particular image. The questions belong to 2 categories (relational and non-relational), with the relational questions being more challenging for non-specialised CNNs, even in the in-distribution case – in the results reported in the original paper, a CNN plateaus at 63% test accuracy on the relational questions [241]. When answering non-relational questions, it is sufficient for a model to learn the properties of individual objects without referring to the whole scene, while the relational questions require learning the idea of spacial distance between the objects ('closest/furthest'), the concept of a same/different relation ('same shape with the blue object') and counting ('how many objects').

times. The question subtype is also chosen at random. Each image is of size 75×75 pixels with each object of a fixed diameter (circles)/side length (squares) of 10 pixels (Figure 4.6 and Figure 4.7 contain random image samples).

In the original Sort-of-CLEVR [241], all the possible 12 shape and colour combinations are included both in the training set and in the test set. The authors of this dataset and the subsequent paper [117] focus only on the in-distribution variant. For the purpose of the experiments in this section, I re-generate the in-distribution variant (Figure 4.6), in which all shapes and colours appear in the training set and in the test set. I additionally generate new out-of-distribution variants of Sort-of-CLEVR with an increasing number of shape-colour combinations that are omitted in the training set but nevertheless appear in the test set (Figure 4.7). In these *out-of-distribution* variants of Sort-of-CLEVR, the architectures are tested on the ability to answer questions based on images containing objects of previously unencountered shape-colour combinations. This is an example of an out-of-distribution generalisation task that is often referred to as *compositional generalisation* [154] (described in Section 2.4.2). In compositional generalisation tasks, a model is evaluated on the examples containing new combinations of familiar features, which is also investigated in Chapter 5 in the context of multi-agent systems.

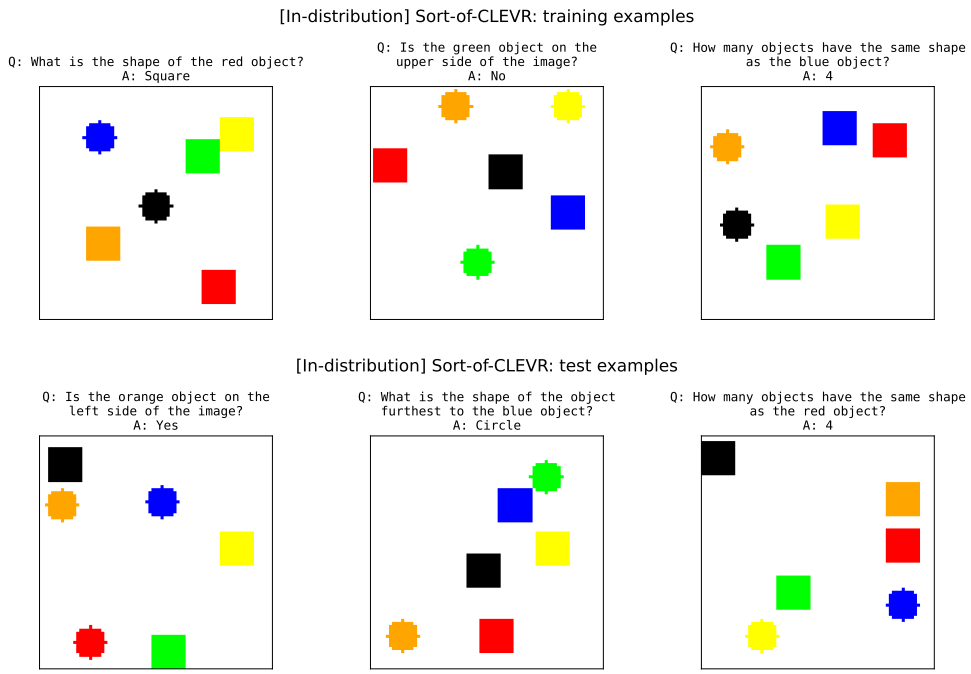


Figure 4.6: In-distribution Sort-of-CLEVR: random samples from the generated training and test data (both the relational and non-relational questions).

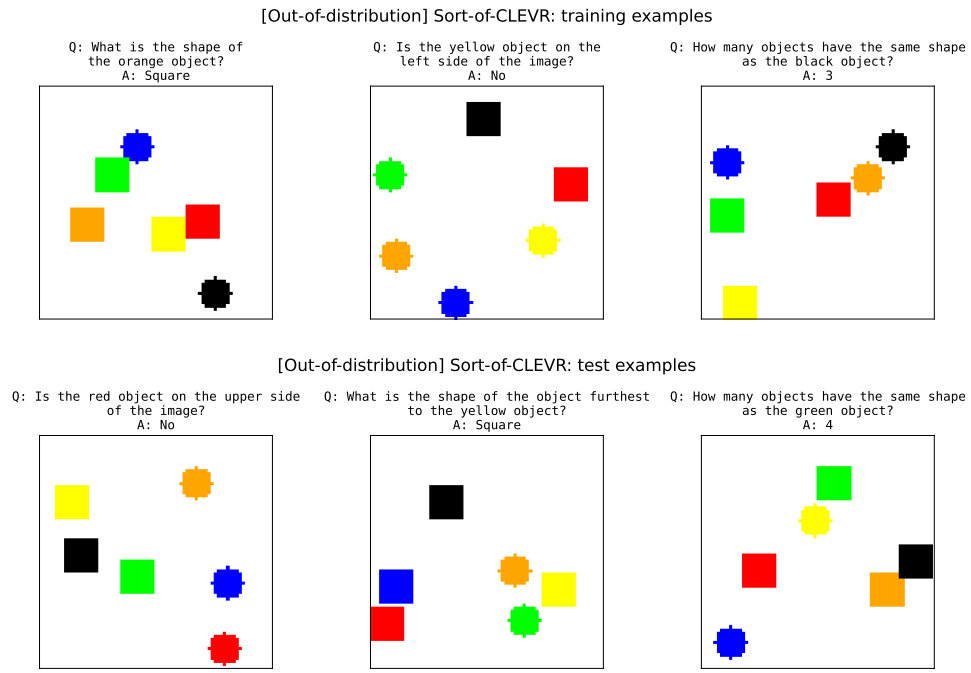


Figure 4.7: Out-of-distribution Sort-of-CLEVR: random samples from the generated training and test data (both the relational and non-relational questions). In this example, there is one held-out combination that appears in the test set and not in the training set: *red circle*. The combinations to be held-out are randomly chosen.

4.2.1.2 Results

The experiments on Sort-of-CLEVR are meant to answer the following research questions:

- **Q1:** What is the effect of augmenting a baseline model with NFM in terms of out-of-distribution generalisation?
- **Q2:** What is the effect of augmenting a baseline model with dilated convolutions in terms of out-of-distribution generalisation?
- **Q3:** In the out-of-distribution variants of Sort-of-CLEVR, is there a systematic difference in the performance on the relational and the non-relational questions?

Q1: Table 4.1 shows the results per number of held-out combinations N (where $N = 0$ is the standard in-distribution Sort-of-CLEVR and $N = 6$ is the maximum number of combinations that can be held-out without changing the number of objects in the scene). NFM improves the performance of a standard convolutional baseline both in terms of the in-distribution and out-of-distribution performance, with overlapping error margins for some of the combinations (for example, non-relational questions in the $N = 3$ case). On average (Figure 4.8) augmenting the convolutional baseline with NFM leads to a higher accuracy in the out-of-distribution scenarios, both in the relational and non-relational tasks.

Q2: In both the relational and non-relational task, the accuracies of Dilated DenseNet and DenseNet overlap on average (Figure 4.8). There is no evidence that Dilated DenseNet improves over DenseNet in this task. However, in the case of relational questions, dilated convolutions appear to help for $N = 0, 1, 2$ (Table 4.1). This might be because a dilated convolution allows for integrating a bigger context (as opposed to local features learnt by the standard convolutional layers), which helps in answering questions about the relations between objects in a scene. This means that the effect previously observed in the in-distribution relational setting [6] extends to simple out-of-distribution relational settings where one or two new colour-shape combinations are introduced.

Q3: The relational and non-relational accuracies are analysed separately, which uncovers two different trends. For all models, the performance on the relational questions is linearly decreasing with the number of held-out combinations. However, in the case of CNN and CNN+NFM, the performance on the non-relational questions seems to oscillate between approx. 76% and approx. 87% regardless of the number of the held-out examples. This is possibly due to (1) non-relational questions being easier, both in terms of the conceptual difficulty and in terms of the random baseline performance: 50% for the non-relational questions and 38.89% for the relational questions (the counting question has the random

Model	Number of held-out combinations (N)						
	$N = 0$ *(In-dist.)*	$N = 1$	$N = 2$	$N = 3$	$N = 4$	$N = 5$	$N = 6$
Relational questions:							
CNN	71.98 ± 0.37	59.66 ± 0.71	52.99 ± 0.73	50.5 ± 0.29	46.57 ± 0.71	42.09 ± 0.43	37.43 ± 0.24
CNN+NFM	78.22 ± 0.68	62.74 ± 0.42	55.8 ± 0.6	54.15 ± 0.29	47.83 ± 0.1	44.66 ± 0.42	39.03 ± 0.19
DenseNet	80.81 ± 0.58	63.58 ± 0.55	55.23 ± 1.39	52 ± 1.01	47.16 ± 0.35	43.72 ± 0.18	38.34 ± 0.3
DilatedDenseNet	82.93 ± 0.83	67.02 ± 1.21	58.25 ± 1.06	51.66 ± 0.28	45.6 ± 0.47	43.5 ± 0.19	38.37 ± 0.47
Non-relational questions:							
CNN	76.04 ± 2.47	82.5 ± 0.35	86.61 ± 1.28	86.59 ± 0.36	84.42 ± 0.18	80.39 ± 0.16	75.64 ± 1.89
CNN+NFM	80.63 ± 2.03	89.95 ± 0.4	90.36 ± 0.61	87.05 ± 1.4	84.53 ± 0.44	81.72 ± 0.98	76.78 ± 1.21
DenseNet	97.48 ± 0.49	95.26 ± 0.31	92.56 ± 0.24	89.85 ± 0.21	86.36 ± 0.46	84.12 ± 0.15	80.7 ± 0.18
DilatedDenseNet	98.44 ± 0.64	94.94 ± 0.7	91.63 ± 0.85	89.11 ± 0.46	86.74 ± 0.23	83.79 ± 0.19	80.59 ± 0.13

Table 4.1: Out-of-distribution generalisation in visual question answering (Sort-of-CLEVR dataset) in terms of test accuracy. The error margins are reported based on three runs. 'CNN' refers to the model implemented based on the description of a CNN baseline in the Sort-of-CLEVR paper [241]. 'CNN+NFM' refers to the same baseline augmented with two passes of NFM. 'DenseNet' is another common baseline (here, the growth rate of 8 and the total number of 16 layers are used). All models are adapted for the visual question answering task: the output of the image processing component (CNN, CNN+NFM, DenseNet or DilatedDenseNet) is concatenated with the question, and the result is passed through an 'MLP' stage consisting of three fully connected layers and a softmax function that transforms the output into a probability distribution over the possible answers. Since the MLP part is the same for all the models, it is omitted in the models' names. All models are trained using Adam [138] with the learning rate equal to 0.001, regularised by the weight decay of 0.00001 [179].

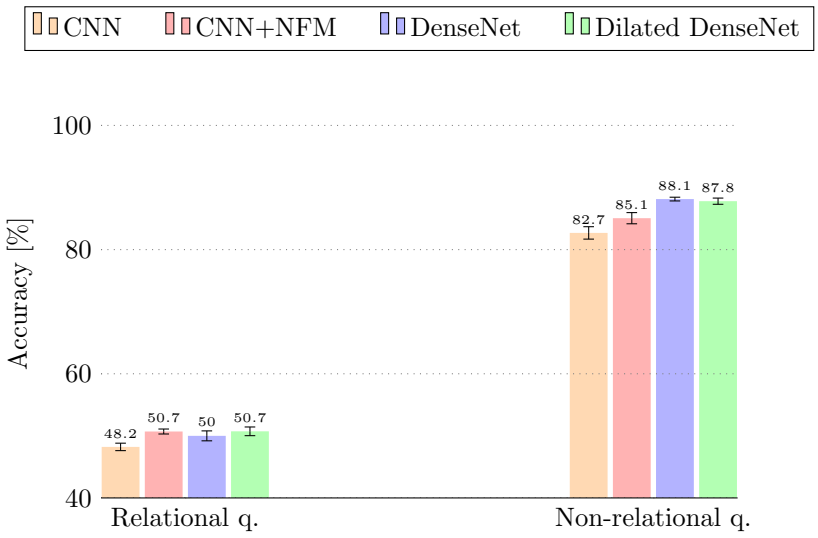


Figure 4.8: Average out-of-distribution generalisation accuracy in Sort-of-CLEVR. The error margins are computed based on the results in Table 4.1 and the rule for computing a standard deviation of a mean of independent variables: $\text{StDev}\left(\frac{X+Y}{2}\right) = \frac{1}{2}\sqrt{(\text{StDev}\,X)^2 + (\text{StDev}\,Y)^2}$. Each average value is computed for 3 random seeds and 6 OOD Sort-of-CLEVR variants (18 independent runs).

baseline performance of 16.67%); (2) each relational question requires detection and analysis of each of the 6 objects in the scene, which means that the model 'pays attention to' the unseen colour-shape combinations more frequently than when answering the non-relational questions centered on a single object in isolation. Each model performs better on the non-relational than relational questions in the in-distribution case (Table 4.1), and this discrepancy increases substantially in the out-of-distribution setting (Figure 4.8).

4.2.2 OOD generalisation in learning the 'same/different' concept

4.2.2.1 Data

Kim et al. [137], inspired by Sort-of-CLEVR, proposed a variant of this dataset – Not-so-CLEVR – where (1) the number of objects is reduced from six to two per image; (2) the set of possible questions is reduced to two opposite questions: 'Are they the same?' and 'Are they different?' (that is, whether two objects in an image have the same shape and the same colour); (3) one colour-shape combination is excluded in the training set in order to test the algorithms in their ability to transfer the skill of recognising same-different relations to unseen objects. Similarly as in my out-of-distribution variant of Sort-of-CLEVR (Figure 4.7), both the colour and the shape from the held-out combination appear in different combinations in the training set.

This dataset is inherently out-of-distribution. The authors found that a specialised

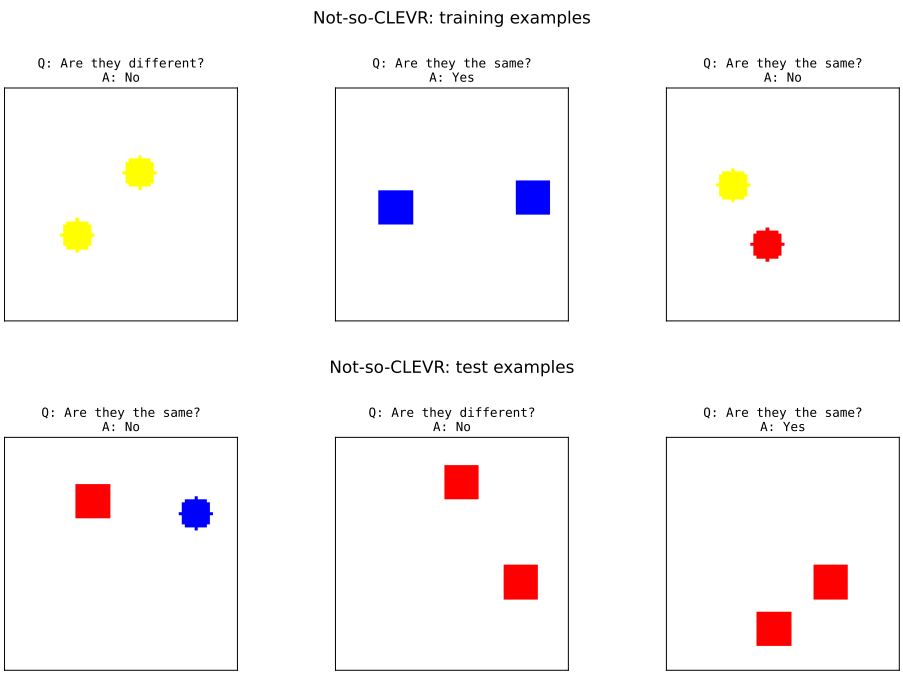


Figure 4.9: Samples from the Not-so-CLEVR task, held out configuration: *red square*.

relational architecture (Relation Networks [241]) that achieves 94% test accuracy on the original in-distribution Sort-of-CLEVR [241] achieves only the random baseline performance on the Not-so-CLEVR task.

This seemingly straightforward task highlights a significant, unresolved challenge in image classification. To the best of my knowledge, nobody has proposed a model that would exceed the accuracy of approximately 50% (random baseline performance) on the original, out-of-distribution Not-so-CLEVR task. In the main follow-up work, Liu et al. [175] use a single-object, greyscale version of Not-so-CLEVR for coordinate classification (where each pixel is a separate object) in an in-distribution context.

Here, I generate the original Not-so-CLEVR dataset following the description by Kim et al. [137]. Figure 4.9 shows an example where the held-out combination is a red square. Similarly as in the original paper, half of the examples in my test set contain images of the same pair of held-out objects (for example, two red squares) and the other half contains images where the held-out configuration (for example, a red square) is paired with a randomly chosen familiar configuration (for example, a red square paired with a dark blue circle). Hence, the held-out configuration appears in each of the test examples, and the ability to correctly classify a pair of equal objects has the same weight as the ability to correctly classify a pair of different objects. Additionally, the samples are generated by pairing each image with one of the two questions: 'Are they the same?' or 'Are they

different?' (assigned with the probability of 50% and encoded as binary vectors, similarly to Sort-of-CLEVR). This makes it harder for the models to learn the mapping between the set of images and the set of the $\{Yes, No\}$ answers – the model is forced to use the information encoded in the question in order to give a correct answer.

Similarly as in the original Not-so-CLEVR paper by Kim et al. [137], I generate 12 versions of the Not-so-CLEVR dataset, each one missing one of the 12 possible colour-shape combinations. For each variant, the number of training and test examples is equal to 9800 and 200, respectively (each image is randomly paired with either 'Are they the same?' or 'Are they different?').

4.2.2.2 Results

The starting point for the experiments in this section is the research question:

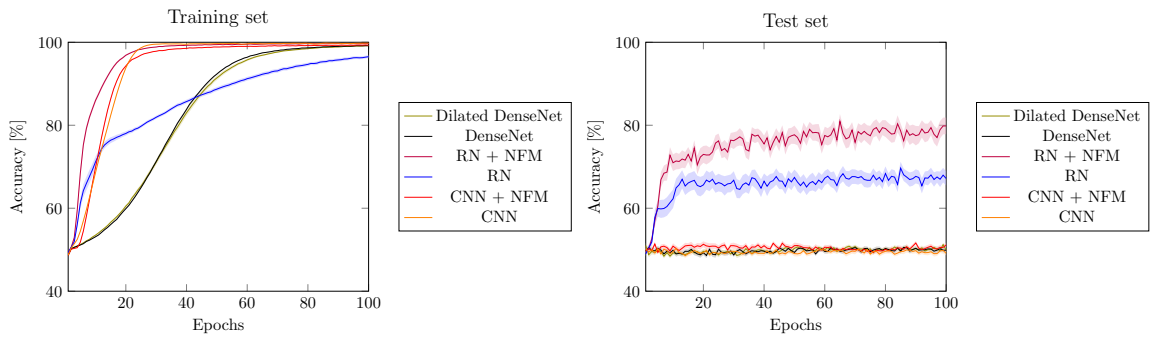
> Can NFMs or dilated convolutions improve the accuracy in the out-of-distribution generalisation task of Not-so-CLEVR?

Here, CNN, CNN+NFM, DenseNet and Dilated DenseNet models are the same as in the previous section, with the only change coming from the difference in the question length in Not-so-CLEVR. They are trained and tested independently on each of the 12 variants of Not-so-CLEVR (one per colour-shape combination that is omitted in the training set). Each variant should pose a problem that is conceptually the same, as they only differ in the held-out colour-shape combinations used in the OOD test set.

Based on the average results (Table 4.2 and Figure 4.10), the only method proposed in this dissertation that slightly improves over the random baseline accuracy of 50% is CNN+NFM. However, I also tried to reproduce the results from the original Not-so-CLEVR paper, and I found that Relation Networks (RN), that were reported to oscillate around the test accuracy of 50%, reach accuracy of around 67% in 100 epochs. The experimental setup in this section differs from the original paper (1) possibly in the number of epochs (not included in the paper); (2) in the size of the image: here, it is 75×75 pixels (the same as the Sort-of-CLEVR images in the previous section), and in the paper it is reported as 128×128 pixels. The authors do mention that the small size of the Sort-of-CLEVR images might be responsible for the high performance of Relation Networks, and they aim to increase the difficulty by increasing the image dimensions. However, changing the image size from 75×75 to 128×128 should not influence the task of recognising same/different objects to this extent: as seen in this section, the size of 75×75 is sufficient to break several convolutional architectures, while Relational Networks perform significantly better (Table 4.2 and Figure 4.10).

Model	*OOD Test Accuracy*
CNN	49.16 ± 0.55
CNN+NFM	51.14 ± 0.60
DenseNet	49.91 ± 0.47
DilatedDenseNet	49.87 ± 0.45
Relation Networks	67.19 ± 1.74
Relation Networks+NFM	79.83 ± 2.04

Table 4.2: Out-of-distribution generalisation in learning the idea of 'sameness' (Not-so-CLEVR dataset). OOD test accuracy is averaged across all 12 possible instances of the Not-so-CLEVR dataset and 3 random seeds (36 independent runs).



(a) Training accuracy

(b) OOD test accuracy

Figure 4.10: Training and test accuracy in Not-so-CLEVR: same/different task.
All methods apart from Relation Networks (RN) converge in 100 epochs in terms of training accuracy. However, only Relation Networks and NFM-augmented Relation Networks improve throughout training in terms of test accuracy, and reach a result above the random baseline performance. NFM-augmented Relations Networks (RN+NFM) learn quickly in the first epochs and achieve the highest final test accuracy of around 80%. Both training and test accuracies are averaged across 12 variants of Not-so-CLEVR and 3 random seeds (36 independent runs), and standard errors are plotted.

Another interesting finding is that augmenting the convolutional part of Relation Networks with NFM leads to a significant gain in OOD test accuracy: the performance increases from 67.19 ± 1.74 to 79.83 ± 2.04 (Table 4.2). It shows that NFM can be integrated into various existing architectures and increase their performance. As seen from the training curves (Figure 4.10), NFM significantly speeds up the optimisation of Relation Networks in terms of the number of required epochs. A plausible explanation is that multi-stage architectures such as Relation Networks are aided by the NFM effect on layer specialisation: if layer roles are inferred earlier during training, the optimiser can achieve better performance in fewer epochs.

4.2.3 OOD generalisation in letter and font recognition

4.2.3.1 Data

Previous experiments evaluate convolutional baselines augmented with dilated convolutions and Neural Function Modules on image classification tasks in the context of visual question answering (using variants of the Sort-of-CLEVR dataset). In this section, I use a data generator for a letter recognition task that lends itself well to creating out-of-distribution splits – the *Synthetic Symbols (Synbols)* generator proposed by Lacoste et al. [151].

Here, I use Synbols[3] to generate the default Synbols dataset that contains 100k images of lower case Latin letters, written using a font that is uniformly selected from a collection of 1120 open-source fonts (Figure 4.11). The second dataset, Less Variation, contains 100k easier to read letters (no bold/italic options, less variation in scale and rotation) and the target classes of 1120 fonts.

These two image datasets (Synbols Default for letter classifications and Less Variation for font classification) are used to generate an array of out-of-distribution tasks.

Motivated by the out-of-distribution experiments ran by Lacoste et al. (Table 2 in the Synbols paper [151]), I reproduce 5 out-of-distribution splits using the default letter classification dataset:

- **Stratified partitions:**
  - *Stratified Font*
  - *Stratified Scale*
  - *Stratified Rotation*
  - *Stratified x-Translation*
- **Compositional partitions:**
  - *Compositional Scale-Rotation*

and 1 out-of-distribution split for the font classification dataset – *Stratified Char*.

For the continuous attributes (Scale, Rotation, x-Translation), the stratified partitions are generated by assigning the first and last 20 percentiles of a continuous latent factor as the validation and test set respectively, and leaving the remaining samples for training. For discrete attributes (Font, Char), the sets of possible values are randomly partitioned

[3]The official implementation: `https://github.com/ElementAI/synbols`.

Figure 4.11: 100 randomly sampled images from the default Synbols dataset. The task is to assign each image to a class corresponding to the appropriate letter/symbol. The images vary in difficulty due to the diversity of fonts, image resolutions, translation, scale and rotation of the letter. There are 48 distinct classes, and consequently the random baseline performance is 2.1%.

Figure 4.12: 100 randomly sampled images from the Less Variation font classification dataset. There are 1120 distinct fonts, and consequently the random baseline performance is only 0.09%.

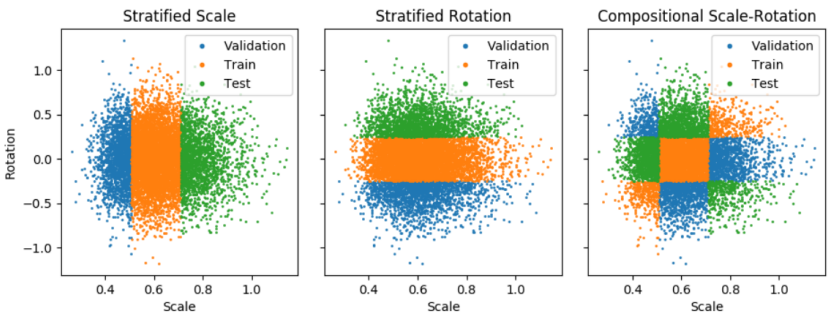


Figure 4.13: Illustration of the method for generating out-of-distribution splits in Synbols proposed by Lacoste et al. [151].
This example uses continous attributes of Scale (size of the letter relative to the image size) and Rotation (unit: radians). In the compositional partitions, the splits are representative with respect to each attribute in isolation (marginal distributions are very similar), however, the models are validated and tested on the distributions that are systematically different than the training distribution. In the stratified partitions, the first and the last 20 percentiles of a continuous latent factor are used as the validation and test set, respectively, while the remaining data points are used for training.
Figure produced by Lacoste et al. [151].

instead of using fixed thresholds (such as the threshold of 20 percentiles). Figure 4.13 shows an example of creating stratified and compositional splits.

4.2.3.2 Results

The experiments on Synbols aim to answer the following research questions:

- **Q1:** How do the results of incorporating NFM into several different convolutional architectures compare to their NFM-less counterparts (both the reproduced baseline results and the official results of these architectures reported in the Synbols paper [151])?
- **Q2:** Is there a systematic difference in the performance on the compositional partition and the stratified partitions?

Table 4.3 includes reproduced results from Table 2 containing the set of OOD experiments performed by the authors of Synbols [151] (Conv_4, ResNet_12, WRN_28), the exact results reported by the authors, and the results of augmenting each of these architectures with NFM (Conv_4+NFM, ResNet_12+NFM, WRN_28+NFM).

Q1: Despite following all the information in the paper and using the official implementation of both the baseline architectures and the Synbols dataset, the results reproduced for the sake of this chapter differ slightly from the results published in the original paper. In Table 4.3, all of the results are included so that NFM variants can be compared both with the individually reproduced results and with the results published by the authors. For

Model	Synbols Default						Less Variation	
	In-dist.	*Stratified Font*	*Stratified Scale*	*Stratified Rotation*	*Compositional Rot-Scale*	*Stratified x-Translation*	*In-dist.*	*Stratified Char*
Conv_4 (reproduced)	71.62 ± 0.05	70.94 ± 0.09	66.86 ± 0.14	53.04 ± 0.17	60.11 ± 0.3	50.52 ± 0.62	3.77 ± 0.09	3.74 ± 0.16
Conv_4 (reported [151])	68.51 ± 0.66	67.17 ± 0.68	71.75 ± 0.17	5.54 ± 0.31	53.87 ± 0.89	44.78 ± 0.72	0.21 ± 0.04	0.24 ± 0.01
Conv_4+NFM	71.94 ± 0.08	71.33 ± 0.05	66.69 ± 0.4	53.01 ± 0.85	60.67 ± 0.38	50.68 ± 0.19	3.85 ± 0.17	3.9 ± 0.11
ResNet_12 (reproduced)	93.86 ± 0.1	92.97 ± 0.07	91.47 ± 0.22	79.28 ± 1.04	87.85 ± 0.4	92.3 ± 0.16	32.22 ± 0.29	22.48 ± 0.36
ResNet_12 (reported [151])	95.43 ± 0.12	94.62 ± 0.09	94.10 ± 0.13	82.37 ± 0.03	90.56 ± 0.20	94.62 ± 0.19	39.41 ± 0.30	25.59 ± 0.22
ResNet_12+NFM	94.19 ± 0.02	93.15 ± 0.07	91.81 ± 0.35	79.6 ± 0.27	88.17 ± 0.06	92.29 ± 0.01	33.2 ± 0.44	23.55 ± 0.73
WRN_28 (reproduced)	94.57 ± 0.08	93.84 ± 0.15	91.29 ± 0.66	79.75 ± 0.6	88.75 ± 0.24	93.73 ± 0.26	15.09 ± 1.28	9.98 ± 1.18
WRN_28 (reported [151])	93.57 ± 0.29	93.27 ± 0.11	92.18 ± 0.16	79.53 ± 0.09	87.68 ± 0.46	92.99 ± 0.07	23.10 ± 0.90	16.85 ± 0.23
WRN_28+NFM	94.58 ± 0.04	94 ± 0.09	92.75 ± 0.15	80.18 ± 0.72	88.4 ± 0.63	93.71 ± 0.22	14.01 ± 0.71	11.21 ± 0.93

Table 4.3: In-distribution and out-of-distribution results on the letter recognition (Synbols Default) and font recognition (Less Variation) tasks. Both the reproduced baseline results and the previously published baseline results are reported. The results are averaged over 3 independent runs. In the implementation shared by the authors of the published results, 3 random seeds are used as well (https://github.com/ElementAI/synbols-benchmarks/blob/master/classification/exp_configs.py).

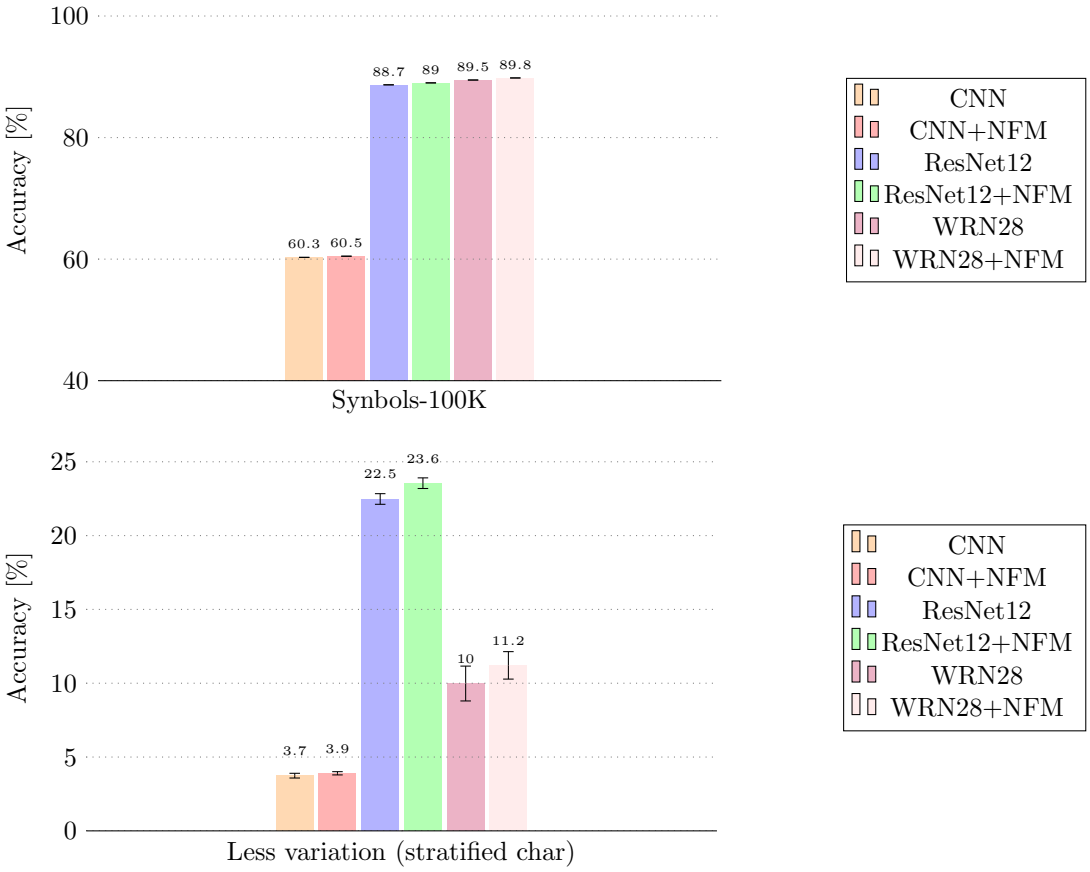


Figure 4.14: OOD test accuracy on the Synbols dataset.
The upper histogram contains average results for 5 OOD splits based on the default Synbols dataset (5 variants and 3 random seeds, that is, 15 independent runs). The histogram below shows OOD performance in the more challenging task of font classification. Following the Synbols paper, the experiments are run on one OOD variant of the Less Variation dataset (3 random seeds).

clarity, histograms in Figure 4.14 contain a comparison of the reproduced baseline results and their NFM-augmented counterparts.

In the new experiments performed for this section, adding NFM improves the average accuracy of each of the convolutional baselines (Figure 4.14) – the averages in the Synbols-100K case (the default dataset) are computed for the 5 OOD tasks and 3 random seeds based on the full set of the results (Table 4.3). In the case of the more challenging Less Variation task, NFM-augmented CNN and NFM-augmented WRN overlap with their respective baselines. However, there is an improvement in the performance of ResNet due to extending it with NFM.

When comparing the new NFM variants to the results published by the authors (Table 4.3), it turns out that CNN+NFM improves over CNN by a larger margin than in the case of an attempt to reproduce the original CNN baseline. CNN+NFM improves over the published CNN results in each in-distribution and out-of-distribution scenario apart from Stratified Scale, where the published result is higher (in the comparison of the reproduced results, the error margins overlap for Stratified Scale). In the case of a vanilla CNN, it is clear that NFM is beneficial regardless of whether the reported or the reproduced results are used.

The reported results of ResNet are higher than I could reproduce (using the code published by the authors). In this comparison, ResNet+NFM is slightly less successful than ResNet. In the new experiments, ResNet+NFM is slightly better than ResNet.

Finally, WRN+NFM improves over the published WRN results for the in-distribution Synbols Default, Stratified Font, Stratified Scale and Stratified x-Translation. In the remaining cases, the error bars overlap.

The conclusion is then that augmentation with NFMs increase the accuracy of the majority of the baselines that were previously used in Synbols-based tasks. Augmentation with NFMs also leads to an improvement over the majority of the published results on Synbols.

Q2: By design, the Compositional Scale-Rotation dataset is more challenging than the Stratified variants. In practice, all architectures are more sensitive to the Stratified Rotation split than to the Compositional Scale-Rotation variant. This is possibly due to the lack of an explicit rotation-robust mechanism (for example, data augmentation). In the Stratified Rotation split, the models see new angles at test time, whereas in the Compositional Scale-Rotation variant all possible rotations are seen in training (in different Scale-Rotation combinations). This is an interesting example of how Convolutional Neural Networks can exploit partial information in a (seemingly more difficult) compositional generalisation task, yet struggle when confronted with a new value in a single feature.

4.2.4 OOD generalisation in digit recognition (multi-object classification)

4.2.4.1 Data

For this last study, I generate out-of-distribution and multi-object variants of the classic MNIST digit classification dataset [66].

Unlike the visual question answering tasks based on Sort-of-CLEVR and Not-so-CLEVR, and the single-object classification tasks based on Synbols, this section covers multi-object classification. In the multi-object MNIST, each image contains from 1 to 5 hand-written digits sampled with replacement from the original MNIST dataset. The goal is to predict whether a certain digit appears in the image. For instance, for an image containing the digits 5 and 4, the model has to detect that these two digits are visible. Figure 4.15 shows several images sampled from the generated multi-object MNIST.

In the OOD variants of this task, the models are tested on their ability to generalise to a new number of digits. This completes the investigation in terms of the most human-interpretable changes to the distribution of image data: the experiments on Sort-of-CLEVR and Not-so-CLEVR presented in the earlier sections are focused on generalisation in terms of the shape and colour. Here, OOD splits are generated by pairing a test set containing a held-out number of digits per image (for example, 2 digits per image) with a training set where the images of the held-out digit cardinality are not present.

4.2.4.2 Results

The experiments in this section aim to answer the question:

> Can NFMs improve the performance of a baseline model in generalisation to a new number of digits in a multi-object classification task?

The CNN baseline and CNN+NFM were trained on four separate datasets and evaluated on five test sets (Table 4.4). For each training dataset, there are two in-distribution test sets (samples of images containing the number of digits present in the test example) and three out-of-distribution test sets (where the model has to generalise to a new number of digits).

As a sanity check, one of the in-distribution test sets contains single-object images only. All methods perform above 90% on this dataset, apart from the vanilla CNN trained on the images containing 1 and 5 digits. However, the same method in the same configuration improves from 87.23 ± 6.51 to 96.06 ± 0.31 after incorporating NFM in the architecture. This might be due to the increased specialisation and modularity introduced by NFM

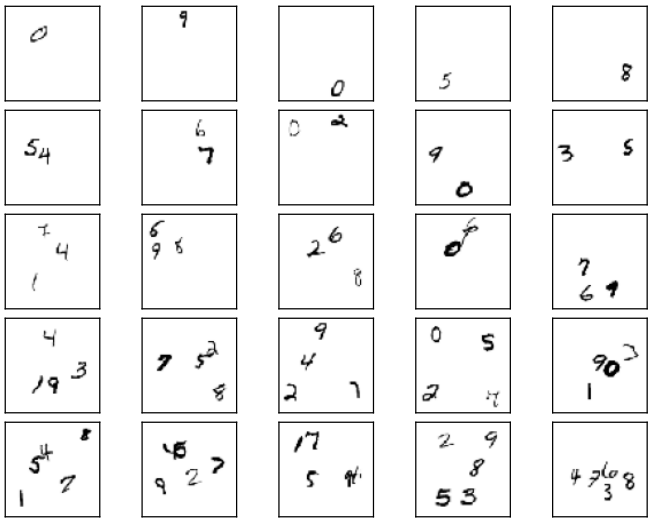


Figure 4.15: Example images sampled from the multi-object MNIST.
All possible numbers of digits per image are shown. Each image has 64×64 pixels. Each digit from the original MNIST is downsampled from the original size of 28×28 pixels to 16×16 pixels and then pasted into a random position within the frame. The positions of digits in an image can overlap. Each training set has 16000 images and each test set has 4000 images.

	Tested on	Trained on			
		1&2	1&3	1&4	1&5
CNN	1	90.77 ± 5.40	95.72 ± 0.14	94.73 ± 1.01	87.23 ± 6.51
	2	73.08 ± 8.16	83.97 ± 0.12	75.45 ± 6.60	69.35 ± 6.94
	3	53.39 ± 1.00	66.55 ± 1.37	61.39 ± 0.93	46.62 ± 3.70
	4	20.46 ± 5.12	44.80 ± 1.44	43.97 ± 6.79	36.71 ± 3.33
	5	9.09 ± 1.07	22.92 ± 1.12	28.53 ± 4.86	26.46 ± 2.75
CNN+NFM	1	$\mathbf{96.37 \pm 0.18}$	$\mathbf{96.29 \pm 0.19}$	95.55 ± 0.37	$\mathbf{96.06 \pm 0.31}$
	2	$\mathbf{82.42 \pm 0.67}$	83.27 ± 0.08	$\mathbf{82.78 \pm 0.33}$	$\mathbf{85.20 \pm 0.92}$
	3	$\mathbf{56.68 \pm 1.21}$	64.43 ± 0.32	63.55 ± 1.94	$\mathbf{64.40 \pm 1.37}$
	4	$\mathbf{27.39 \pm 1.31}$	42.78 ± 0.46	49.94 ± 1.74	$\mathbf{52.10 \pm 1.35}$
	5	$\mathbf{11.66 \pm 0.62}$	21.73 ± 1.74	$\mathbf{36.27 \pm 1.57}$	$\mathbf{41.81 \pm 1.92}$

Table 4.4: In-distribution and out-of-distribution results on the multi-object MNIST.
OOD test sets are highlighted. The instances when CNN or CNN+NFM is better than the alternative method are in bold. In the remaining entries, error margins overlap for CNN and CNN+NFM. 'Trained on 1&3' etc. refers to using a training dataset that contains images of these two digit cardinalities. All models were trained for 500 epochs. Error bars are computed for three independent runs.
CNN+NFM consistently outperforms CNN in extrapolation to a larger number of digits in an image.

(distinct layers might learn to detect distinct digits), which makes the method more robust to the changes in the digit count.

Incorporating NFM improves the OOD performance in eight instances, whereas the vanilla CNN is better only in one instance (for the remaining configurations, error margins overlap). NFM is the most beneficial to the model trained on the training sets containing 1 and 2 digits, and to the model trained on the images of 1 and 5 digits. In these cases, the performance is improved for all of the in-distribution and out-of-distribution test sets. CNN+NFM tends to have smaller standard errors than CNN.

4.3 Summary

This chapter investigates two designs – dilated convolutions and Neural Function Modules – in the context of out-of-distribution generalisation in image classification.

Dilated DenseNets have previously been used in in-distribution relational reasoning, where they can match the performance of specialised Relation Networks. Dilated DenseNets share a similar motivation as Neural Function Modules; in particular, in the context of multi-level feature aggregation that – I hypothesise – improves out-of-distribution generalisation in image classification. In the out-of-distribution experiments conducted in Section 4.2.1, there is a small improvement in the set of relational questions after introducing dilated convolutions to a generic DenseNet architecture. However, this can only be seen for a small number of omitted shape-colour combinations. There is no evidence that Dilated DenseNets help in a larger study where the number of omitted shape-colour combinations varies up to N ($N \in \{1, 2, 3, 4, 5, 6\}$). There is also no evidence in the non-relational set of questions and in the case of learning the same-different relation (Section 4.2.2), where augmentation with NFM is able to improve even a very weak CNN baseline, while Dilated DenseNet does not improve over DenseNet. Since Dilated DenseNet does not look promising in these extensive experiments (18, 18 and 36 independent runs, respectively), the focus of the remaining sections is on the NFM architecture, which appears to be more useful in practice. Based on the DenseNet/Dilated DenseNet results, it seems that dilated convolutions are helpful in addressing small distribution shifts in the context of relational reasoning, but there is no evidence that they help in broader OOD contexts.

Neural Function Modules are consistently beneficial when incorporated into different architectures (a simple CNN, Relation Networks, ResNet, Wide ResNet) and evaluated on different OOD tasks (relational questions, non-relational questions, same-different relation, generalisation to new fonts, new characters and a new number of objects in multi-object classification). These results show the great flexibility of NFM and suggest that there is a relation between using NFM and out-of-distribution generalisation, regardless of the

architecture and the task. They also show that NFM can be successfully incorporated both into CNNs written from scratch and into existing, specialised architectures such as Relation Networks.

This work also connects to the well-known reproducibility issue in machine learning. Despite using the official implementation and the information provided in the published papers, some of the baseline results differ. I include both the individually reproduced and the previously published baseline results.

Finally, since the OOD Sort-of-CLEVR and OOD multi-object MNIST are proposed and implemented for this chapter, they are released to foster the research on out-of-distribution generalisation in image classification: https://github.com/Slowika/ood-dataset-generators. I believe these datasets can be useful in evaluating CNN-based methods in terms of out-of-distribution generalisation because they are simple while posing unresolved problems that strain existing methods. These problems include generalisation to an increasing number of digits (*productivity* [281]) and generalisation in the context of learning spatial relations and counting, which is challenging already in the in-distribution setting.

This chapter approaches the *modularity* theme at the architecture level. Neural Function Modules (Section 4.1.2) can be used to increase modularity of any neural network through the mechanism of dynamic selection of the outputs of the previous layers, and the combination of top-down and bottom-up feedback. Both of these mechanisms are loosely inspired by the global workspace theory, which posits that brain has a modular structure [12]. Attention allows dynamic, sparse connections that are more akin to what might be happening in the cortex than fixed connections in a standard neural network [90]. Top-down feedback is a simplified implementation of the ability to use the contextual information apart from the sensory input. It provides the attention modules with a richer choice of both low-level and high-level information.

This chapter also tackles the second research question posed in Section 1.2: *Do methods that encourage multi-level feature aggregation help in improving out-of-distribution generalisation in image classification?*. To this end, it evalutes two different approaches to integrating low-level and high-level features: Dilated DenseNets and Neural Function Modules across several different instances of image classification. One of the architectures (Neural Function Modules) turns out to be more promising, which suggests that dilated convolutions, while beneficial in the relational reasoning tasks [6], might not be the right direction for increasing out-of-distribution generalisation.

To conclude, this chapter presents a new method, Neural Function Modules, with the motivation of improving out-of-distribution generalisation in image classification, and

provides new evidence on how NFMs improve the OOD accuracy across several types of image classification. This accomplishes the goal of the thesis, that is, to improve out-of-distribution generalisation in machine learning, from the perspective of image classification. Image classification is a single-agent task and one of the main applications of modern machine learning. The next chapter contributes to the research on out-of-distribution generalisation from the perspective of multi-agent communication.

Chapter 5

Out-of-distribution generalisation in multi-agent systems

To understand language is to understand generalisation.
–Eric Jang (2021)

This chapter presents my work on out-of-distribution generalisation in *multi-agent systems* where several *intelligent agents* (each agent corresponding to a separate neural network) interact and communicate with each other. In contrast to previous chapters, in which a single model learns from the examples coming from a predefined and fixed dataset with regression or classification labels, the agents in a multi-agent setup learn through simulations and interactions.

In a realistic multi-agent scenario, such as team cooperation towards a shared goal, the agents must adapt to the outcomes of their own previous actions as well as to the actions of their partners. As a result, a multi-agent scenario inherently requires the agents to be resilient to changes in data distribution [23].

In the previous chapter, dedicated to out-of-distribution generalisation in single-agent classification tasks, the agent is introduced to out-of-distribution data samples in the form of new combinations of the constituents previously seen in a limited number of combinations: for instance, having learnt the meaning of a 'red square' and a 'blue circle', the agent is tested on the ability to correctly classify a 'red circle'. The assumptions are that a future distribution in a fixed task is not completely different from the training distributions, and that the semantic interpretation of the data is important (for example,

in practical image recognition tasks, the semantic, human-like interpretation of images is more important than detection of perturbations to individual pixels).

In this chapter, the agents are tested on the ability to *communicate* out-of-distribution data samples, and act upon the messages describing these samples: after learning to communicate the concepts of 'red square' and 'blue circle', are they able to successfully communicate 'red circle'?

The two main research questions explored in this chapter are:

- How does *data representation* affect the ability to learn to generalise out-of-distribution in a multi-agent setting?
- How does *the number of agents* affect the ability to learn to generalise out-of-distribution in a multi-agent setting?

Chapter structure

The chapter presents the results of research in multi-agent systems with graphs (Section 5.1) and images (Section 5.2) as input data.

Graphs can represent arbitrary relational and hierarchical information; these have not yet been studied in the framework of multi-agent/emergent communication used throughout this chapter. Images have been previously used in multi-agent communication [163, 69], yet developing an interpretable or compositional language based on image data remains a challenge [163]. Using realistic images in this framework is interesting in the context of the research goal of learning succinct, practical image representations, that can be used to solve downstream tasks.

Section 5.1 shows the results of the out-of-distribution experiments with more commonly used *bags-of-words* and *sequential* representations, and goes a step beyond by proposing and analysing multi-agent systems defined over graph input data in the form of *graph referential games*. This multi-agent framework is used to test the influence of data representation and corresponding learning methods on generalisation and language properties. This section contains motivation and the context, description of the games and their main components, research questions posed, and empirical results with analysis and conclusion. This section defines the notion of a referential game, used throughout the chapter. Apart from directly measuring the ability to generalise out-of-distribution by measuring the accuracy on out-of-distribution samples, this section introduces the concept of *language compositionality*, which helps as a proxy measure for the agents' ability to readily recombine the descriptions of the basic concepts in the data.

Section 5.2 presents the results of research on generalisation in *visual* referential games with realistic images as input (where agents learn to communicate about images). This section contains motivation and the context, description of the games and the experimental method, research questions posed, and experimental results with analysis and conclusion. This section approaches the problem of implementing out-of-distribution experiments by modifying a labelled image dataset and adapting it to the framework of multi-agent games. The sampled out-of-distribution categories strike a balance between grounding in the training data and sufficient difference with respect to the standard in-distribution testing examples.

Related publications Section 5.1 shows the results from two first-authored publications: a workshop paper that I presented at the *Workshop on Reinforcement Learning in Games at The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI 2020)* [254] and an article published in the *Proceedings of the 43rd Annual Meeting of the Cognitive Science Society* (and presented at the CogSci 2021 conference) [259]. I also covered the material from this section in a well-attended long talk *Emergent Multi-Agent Communication: The story so far* at the Artificial Intelligence Research Group Talks (Computer Laboratory).

Section 5.2 presents the results of experiments I conducted in a collaboration with the authors of *Interpretable agent communication from scratch* [69] that have not been published yet.

5.1 Graph referential games

This section presents *graph referential games*, a framework for testing the effect of *data representation* on out-of-distribution generalisation in multi-agent games. The graph referential games are fully cooperative: the agents learn to communicate with each other in a series of simulations in order to solve a shared task.

I first motivate the idea of communicating graphs in Section 5.1.1. Afterwards, in Section 5.1.2 I define graph referential games in multi-agent communication. This sets the stage for describing the experiments. The initial experiments, described in Section 5.1.3, explore the performance of graph agents in graph referential games. The main experiments are described in Section 5.1.4. The research questions are given in the experimental sections.

5.1.1 Motivation and related work

This work is loosely motivated by two findings regarding the acquisition of the ability to generalise in biological and artificial systems.

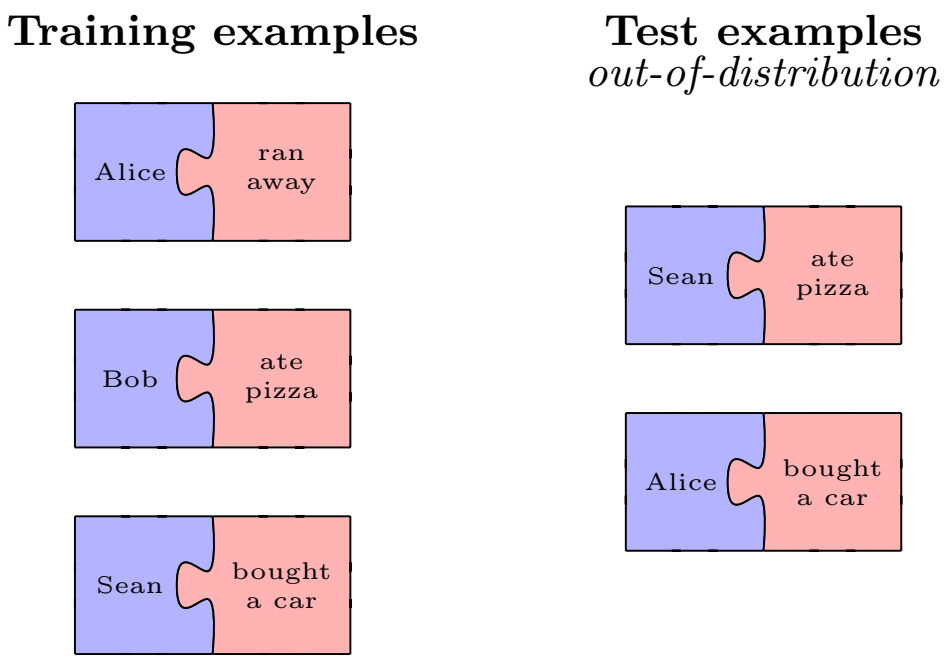


Figure 5.1: An illustration of the main idea behind the OOD evaluation in graph referential games (Section 5.1).
This instance of an out-of-distribution generalisation task follows *the principle of compositionality*: 'The meaning of a complex expression is determined by its structure and the meanings of its constituents' [127], and it has been shown to be difficult for deep neural networks to learn [170, 15]. See Section 2.4.2 for an in-depth review of compositional generalisation in machine learning.

Structured representations lead to a structured language, which might lead to a better generalisation. Lazaridou et al. [163] show that the emergence of a structured, *compositional* language in a multi-agent game is tied to the degree of structure in the input. This in turn is said to corroborate a hypothesis on the influence of *structured semantic representations* [263] on language compositionality from human language evolution studies. Compositionality is believed to be a sufficient (even if not necessary) condition for being able to generalise to new combinations of familiar constituents [43, 201]. For instance, Figure 5.1 shows training examples ('Alice ran away', 'Bob ate pizza', 'Sean bought a car') consisting of two parts connected in a regular way: one of the possible subjects ('Alice', 'Bob', 'Sean') and a possible action ('ran away', 'ate pizza', 'bought a car'). If an agent learns to separate the concepts of the subject and the action, and to communicate them in a compositional way, it will at least generalise to new combinations of familiar words ('Sean ate pizza'), which is often referred to as *compositional generalisation* or *systematic generalisation* in the context of machine learning (for example, in the work of Brenden Lake [153, 235], Dzmitry Bahdanau [14, 24], and their collaborators). The hypothesis and the empirical results on structured representations inducing a structured language lead to the idea of representing input data as trees and graphs in multi-agent communication games.

Another important result from the paper by Lazaridou et al. [163], which helps in the interpretation of the experimental results presented in this chapter, is that more information in the input and a higher complexity of the agents does not translate to a more compositional language or to a higher generalisation to unseen objects (as shown using images and symbolic representations, and agents with and without a neural visual module, respectively). This key prior work shows that structure precedes semantics in emergent communication, as vision models applied to unstructured image data struggled to produce structured language. This work motivates the main experimental results presented in this chapter (Section 5.1.4).

Representing knowledge in terms of relations and hierarchies might help in understanding new information. Humans' ability to solve previously unseen problems by composing familiar skills relies on our ability to represent knowledge in a relational and hierarchical way [206, 89]. David Navon [206] studies extrapolation based on hierarchical reasoning in the context of global and local visual features. He shows that global analysis of the scene and the context precedes local analysis of its constituents, which highlights the importance of the relations between the 'building blocks' in learning to solve various tasks requiring inference and extrapolation. In linguistics, it is generally agreed that various aspects of human language have a natural representation as trees or graphs [55, 25], which are flattened into a linear sequence of words when humans communicate. For example, the

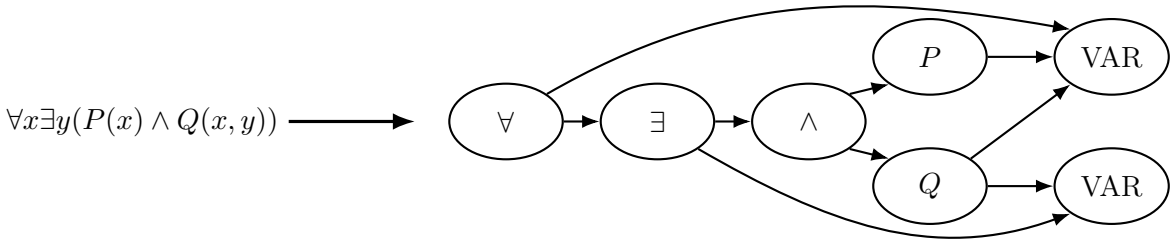


Figure 5.2: An example of a logical statement represented as a graph, recreated from the paper by Mingzhe et al. [285].

relationship between words in a sentence is frequently modelled using formal grammars (for instance, context-free grammars) – and the *parse trees* for sentences matching a desired non-terminal symbol show these relations. Such non-linear representation is more flexible when it comes to understanding new sentences as it captures the relations between words and the rules of a grammar. In natural language processing and machine learning, this hypothesis leads to a number of studies empirically showing that graph and tree representations improve generalisation [251, 1, 239]. Similarly, graphs prove to be a useful and flexible data representation in scene understanding in computer vision [145, 294] and in automated theorem proving [209] due to the potential to represent relations between constituents (for instance, relations between the elements of a scene in computer vision, or syntactic and semantic structures of a mathematical formula such as variable binding). Graph representations allow invariance to variable renaming: for example, the graph representing the formula in Figure 5.2 is the same regardless of how the variables are named in the sequential representation, which allows a greater robustness to systematic changes in the input than a sequential representation.

5.1.2 Problem setting

This section describes the design of graph referential games: the basic components of each game and the way the agents learn (Section 5.1.2.1), input data (Section 5.1.2.2) and the agents' architectures (Section 5.1.2.3).

5.1.2.1 Game

In the work presented in this chapter, each agent takes the role of either a *speaker* (f_θ) or a *listener* (y_ϕ) (Figure 5.3). The speaker has access to input data (*target* d^*) and is tasked with describing the input to the listener, using a message m of predefined maximum length l. The listener receives the message from the speaker as well as a set consisting of the target and *distractors* – new objects sampled without replacement from the target data distribution. The agents are rewarded if and only if the listener recognises the target among distractors. None of the agents has a predefined knowledge of the language the

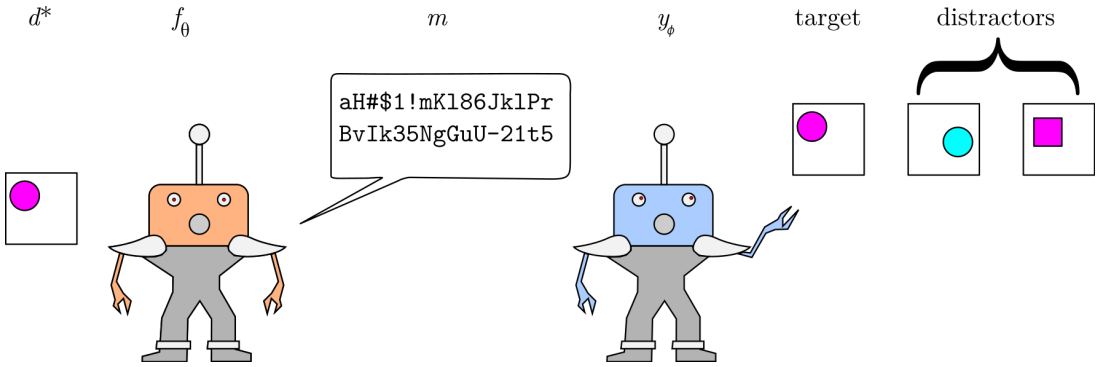


Figure 5.3: An illustration of a referential game used throughout this chapter.
The game is a variant of the *Lewis signalling game* [169] from game theory, which since then has been used in the studies on language evolution [33, 264] and in multi-agent research in machine learning [162, 163, 45].
A referential game involves two agents with fixed roles: the *speaker* and the *listener*. Here, the agents are represented by neural networks f_θ and y_ϕ.
The speaker has access to input data and is tasked with describing the input to the listener. The listener receives the message from the speaker and a set consisting of the target and distractors – new objects sampled without replacement from the target data distribution. **A set that consists of a target and distractors is referred to as a game sample. In standard in-distribution experiments, each sample contains a target and distractors sampled without replacement from the same input distribution.** The agents are rewarded if and only if the listener recognises the target among distractors. In existing work, target and distractors are typically represented as bags-of-words or sequences [41], with more recent studies focusing on images in referential games [69]. The work presented in this chapter contains the first studies on using graph representations and *graph representation learning* methods in referential games.

speaker uses to describe the targets – instead, they learn to communicate with each other by simulations of trial and error based on the speaker's utterances.

Depending on the input representation, both the speaker and the listener must have an appropriate *encoder* to process the target and distractors (for example, a convolutional neural network for image data or a graph neural network to process graphs). The speaker also has a *decoder* capable of generating discrete messages. The decoder is based on the sequence-to-sequence encoder-decoder method proposed by Sutskever et al. [270] and adapted for referential games by Havrylov and Titov Havrylov and Titov [104]. The decoder architecture remains the same regardless of the encoder/representation learning method used by the agent in the game.

One of the key challenges of training a referential game is its *non-differentiability*. This is a consequence of the fact that each of the *symbols* (also referred to as *words*, following the convention in emergent communication literature [161, 203]) in the message is seen as a one-hot vector with $|V|$ components, where V is the vocabulary (set of all available symbols), with the i-th component equal to 1 if and only if the i-th symbol is used. A training algorithm, when it processes sample messages, is actually working over a vector of such one-hot vectors. Non-differentiability is overcome by the agents learning jointly using REINFORCE [290] or using a Gumbel-softmax relaxation (Section 2.3.1.2). Either of these approaches renders the game differentiable, and they are both widely used in the existing literature on referential games (REINFORCE in referential games [163, 162]; Gumbel-softmax in referential games [104, 63, 68, 69]). The latter approach is used in graph referential games and I briefly describe it and motivate it below.

The graph referential games presented in this chapter use the *Straight-through Gumbel-softmax estimator* (STGS) introduced by Bengio et al. [22] – in brief, allowing the game to be learned using backpropagation by using continuous relaxation in the backward pass. Unlike REINFORCE, the method based on Gumbel-softmax trick allows propagation of the gradients from the listener to the speaker. Denamganaï et al. [63] argue that STGS gives a richer signal towards solving the *credit assignment* problem[1] in emergent communication than REINFORCE. In the context of graph referential games, the most important axis of variation is the choice of an input representation and a corresponding representation learning method. The optimisation method is fixed to be STGS in each configuration.

In each simulation, the speaker receives the target graph d^* as input and produces a message m, in which each word is drawn from the vocabulary V, where V refers to the

[1]Credit assingment in machine learning refers to the analysis of which weights in neural networks are responsible for their success or failure, and how changing these weights can improve the performance [196].

finite set of all distinct words that the speaker can generate. The speaker creates m by using an encoder-decoder architecture f_θ, in which the Gumbel-softmax trick is used in the decoding step. The input to the listener consists of the message m sent by the speaker along with the set K of distractors and the target d^*. The listener then produces an output o, which is a vector in $\mathbb{R}^{|K|+1}$. Softmax is then applied to o to produce a categorical distribution over the set $K \cup \{d^*\}$. We formally write this as follows:

$$m \leftarrow f_\theta(d^*)$$

$$o = y_\phi(m, K \cup \{d^*\})$$

The complexity of graph referential games is controlled through the parameters $|V|$ (vocabulary size: the number of categories in the distribution from which each symbol/word is drawn), the number of distractors $|K|$ and maximum length of the message l. The first experimental section (Section 5.1.3) shows the results of an initial exploration using one-word messages ($l = 1$).

A *sample* in this environment consists of a target graph d^* and a set of $|K|$ distractors:

- In *in-distribution samples*, the target and each of the distractors are sampled from a single set of graphs $\mathcal{G}$ without replacement. The resulting collection of samples is split into train, validation and test subsets. Therefore, in the in-distribution case the agents might see familiar graphs in new target-distractors configurations at the testing stage. This is the standard practice for creating train/test or train/validation/test partitions in existing referential games [75, 163].

- In *out-of-distribution samples*, the set of graphs $\mathcal{G}$ is partitioned into $\mathcal{G}_{\text{train}}$ and $\mathcal{G}_{\text{test}}$. The train split contains as the target and the distractors only graphs drawn from $\mathcal{G}_{\text{train}}$, whereas the test split contains only graphs drawn from $\mathcal{G}_{\text{test}}$. Therefore, the agents see not only new samples but also new sample constituents (new graphs) at the testing stage. This is a novel approach in testing the agents in referential games.

5.1.2.2 Data

Graph referential games contain *Game-1* with tree representations including an empty root, and *Game-2* with arbitrary graphs (Figure 5.4).

Game-1: hierarchy of concepts and properties In this game, a tree is constructed from a vector of $[p_1, p_2, \ldots, p_n]$, where n corresponds to the number of nodes (*concepts*) and $p_1, p_2, \ldots, p_n$ denote the numbers of possible node values (*properties*). This is a proposed implementation of the idea of describing an object via communicating fixed concepts and

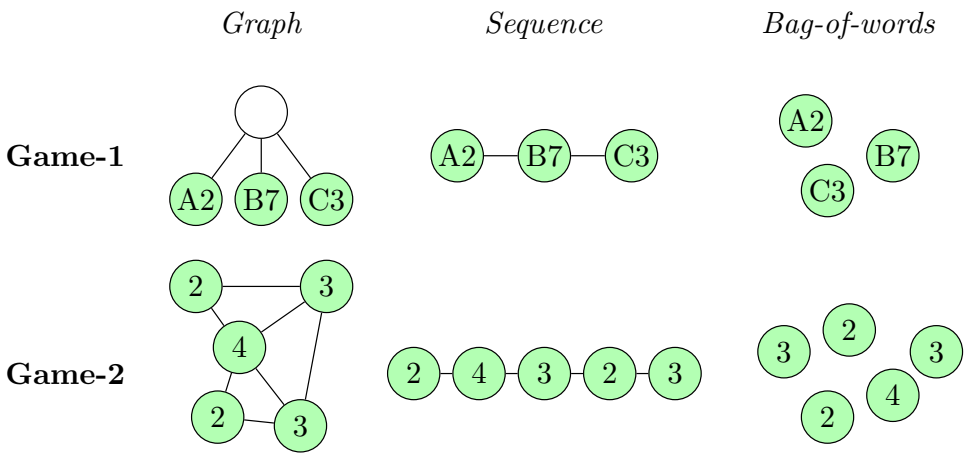


Figure 5.4: Structural biases in the game input.
Baseline representations of sequences and bags-of-words are constructed as in the existing work on emergent communication [142, 163]. Graph inputs are processed using graph encoders, while sequences and bags-of-words use a recurrent neural network and a linear layer, respectively. The sequences preserve the order of properties, while in the bags-of-words representation the properties are shuffled and concatenated before applying a linear layer. In Game-1, letters represent concept vectors, while digits represent property vectors.

the observed properties assigned to the concepts. For example, a visual object can be described using the concepts of 'colour' and 'shape'. For a specific visual object, such as a 'red square', the concepts of 'colour' and 'shape' are assigned the properties of 'red' and 'square', respectively. Each tree has the same number of concepts n and they only differ in the property values. Formally, each tree is an undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the set of all nodes representing unique properties along with a 'central' node, and $\mathcal{E}$ is the set of edges. The central node corresponds to the latent representation of the entire input, such as of a 'red square' as a single entity, which is initially empty and then learned by a graph agent based on the properties of the concepts. The node features consist of a concatenation of the property encoding and the type encoding (represented as one-hot vectors), to disambiguate between the concepts of the equal number of possible properties. The information distinguishing the target from the distractors comes from at least one difference in properties (for example, a property 'red' of the concept 'colour' distinguishes a 'red circle' from a 'blue circle').

Game-2: relational concepts In this game, graph agents learn to communicate about arbitrary undirected graphs with a varying number of edges. Such graphs are meant to represent relations between arbitrary entities; for example, connections between users of a social media platform. In Game-2, each undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is defined over the set of nodes $\mathcal{V}$ and the set of edges $\mathcal{E}$. In a given instance of the game, $|\mathcal{V}|$ is fixed for all targets and distractors. The number of edges varies across the graphs. Each node has a self-loop in order to include its own features in the node representation aggregated through

message passing from the neighbouring nodes. We use node degrees converted to one-hot vectors as the initial node features. The information distinguishing the target from the distractors comes from the graph topology (node degrees). Game-2 extends Game-1 by allowing the agents to communicate arbitrary relations, without using the 'hierarchy' of an empty node.

The baseline data representations of bags-of-words and sequences contain the same features (the same vectors) as the nodes in the graph representation. Sequence representations follow a fixed order of nodes for each graph (based on the matrix representation of nodes in the Deep Graph Library [286] used in the implementation of graph data and graph agents). Bag-of-words representations are the least structured, with no ordering or relations between the values.

5.1.2.3 Agents

Graph referential games come with three groups of generic encoders: bag-of-words encoder, sequence encoder and graph encoders. These encoders use a comparable number and size of the hidden layers so that the main axis of variation between them is the degree of structure in the input. Bag-of-words and sequence encoders are typically used in multi-agent communication [163, 44] whereas graph encoders have not been used before in this setting.

A graph encoder generates node embeddings for each node, and then it uses them to construct an embedding of the entire graph. Node representations are computed for each node v_i through neighborhood aggregation that follows the general formula

$$\boldsymbol{h}_{v_i}^{(l+1)} = \mathsf{ReLU}\left(\sum_{j \in N_i} \boldsymbol{h}_{v_j}^{(l)} \boldsymbol{W}^{(l)}\right),$$

where l corresponds to the layer index, $\boldsymbol{h}_{v_i}$ are the features of node v_i, $\boldsymbol{W}$ refers to the weight matrix, and N_i denotes the neighborhood of node v_i. Graph referential games use encoders parametrised by a Graph Convolutional Network (GraphConv [141]) and GraphSAGE [102], an extension of GraphConv that allows modification of the trainable aggregation function beyond a simple convolution. An embedding of an entire graph is obtained through pooling the node features using average, sum or max functions. A graph embedding vector is used in the same way as the hidden state (*context*) in a sequence-to-sequence model; that is, a graph speaker is a graph-to-sequence encoder-decoder.

Figure 5.5 shows a comparison between the graph encoders considered. The purpose of this experiment is to choose a variant of a graph neural network for the speaker and the listener. GraphConv with sum pooling outperforms its extension GraphSAGE in a graph

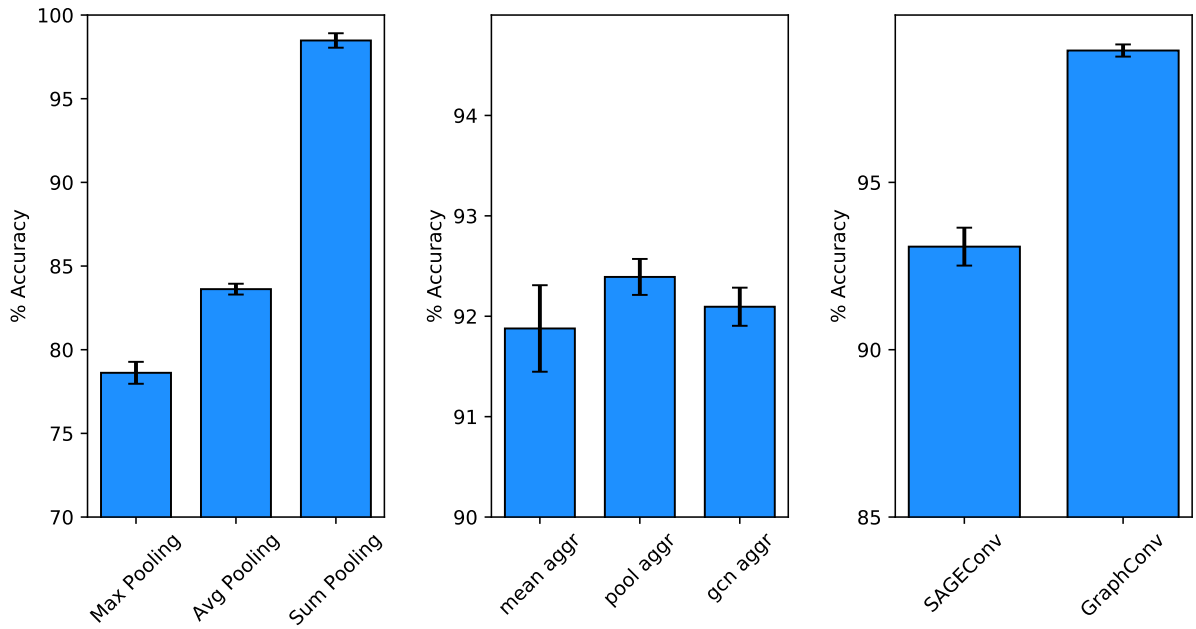


Figure 5.5: Ablation studies on different parameters of graph neural networks in Game-1. GraphConv is the standard graph representation learning method [141] and SAGEConv/GraphSAGE is an extension capable of handling dynamic graphs [102] (right plot). An embedding of an entire graph is computed through aggregation of the node embeddings using functions such as `max`, `mean` or `sum` (left plot: the results of varying the node pooling method in GraphConv). GraphConv uses convolution at the stage of learning node embeddings: SAGEConv allows for replacing this function with a mean aggregator or a pooling aggregator, which uses an additional neural layer to learn the node embedding [102] (middle plot: the effect of varying the aggregator in SAGEConv). The y-axis in all plots represents test accuracy. All the runs are averaged across three different random seeds and standard error bars are shown.
Figure produced by Abhinav Gupta.

referential game, which leads to the choice of this method for the remaining experiments. This choice also aligns well with the goal of keeping the methods simple and generic, with the degree of input structure as the main axis of variation.

5.1.3 Initial experiments with one-word messages

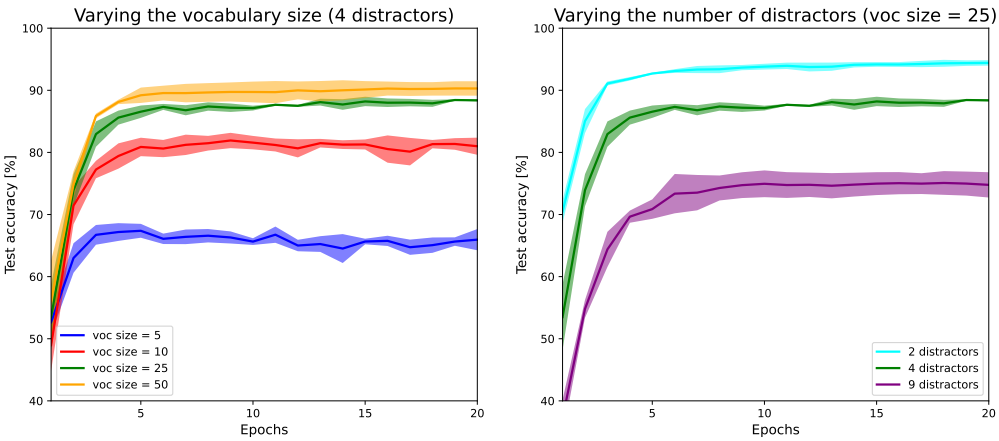


Figure 5.6: Learning curves showing the performance of the agents on the test set in Game-1 ($p = 3$ concepts of $t = 4$ properties each).
Train, validation and test split is 6:2:2. Left: The vocabulary size varies with a fixed number of distractors (4). Right: The number of distractors varies with a fixed vocabulary size (25). The results are averaged across five independent runs and error bars are shown.

This subsection shows initial experiments using graph agents and one-word messages in graph referential games (message length 1). This part is focused on looking into the efficacy of graph agents in referential games before comparing them to the representation learning methods relying on a lower degree of structure in the input (sequences and bags-of-words). The research questions are:

- **Q1:** Does the game difficulty increase with the increase in the number of distractors? (Figure 5.6, right.)
- **Q2:** Does the speaker make use of the entire vocabulary? (Figure 5.6, left and Table 5.1.)
- **Q3:** Is the listener able to solve the task without the signal from the speaker? (Figure 5.7)

Q1: Given a set of $|K|$ distractors, a random baseline in a graph referential game has the accuracy of $\frac{1}{|K|+1}$ (probability of selecting the target uniformly at random in the set of $|K| + 1$ objects). Figure 5.6, right shows a comparison in test accuracy depending on the number of distractors, for $|K| = 2, 4, 9$. The corresponding accuracies of the random baselines for these variants of the game are 33%, 20% and 10%. While the graph agents

$\|V\|$	2 distractors	4 distractors	9 distractors
5	5.0 ± 0.0 (100 ± 0)	5.0 ± 0.0 (100 ± 0)	5.0 ± 0.0 (100 ± 0)
10	9.33 ± 0.47 (93.33 ± 4.71)	9.33 ± 0.47 (93.33 ± 4.71)	9.0 ± 0.81 (90 ± 8.16)
25	12.66 ± 0.47 (50.66 ± 1.88)	14.66 ± 0.47 (58.66 ± 1.88)	13.33 ± 0.47 (53.33 ± 1.88)
50	14.0 ± 1.41 (28.0 ± 2.83)	16.33 ± 0.94 (32.66 ± 1.89)	16.66 ± 0.47 (33.33 ± 0.94)

Table 5.1: The number of symbols the system used per number of distractors in the games that converged in terms of training accuracy. The parentheses show the percentage of symbols used for the given vocabulary size. All the values are averaged across three different random seeds and standard errors are shown. There is little increase in the number of symbols used in the game even if the vocabulary size is doubled.

perform well above the random baseline in each configuration, there is a drop in test accuracy in a game with 9 distractors. The task of recognising the target becomes more difficult as the number of distractors increases.

Q2: A naive way to solve the game is to learn a unique symbol for each graph. The example shown in Figure 5.6 involves $4^3 = 64$ unique trees as described in the Game-1 paragraph (Section 5.1.2.2). The naive approach requires a vocabulary V of at least $|V| = 64$ symbols. However, increasing the vocabulary beyond $|V| = 25$ symbols brings little improvement in test accuracy (Figure 5.6, left). Interestingly, in the games that converged in terms of the training accuracy, the agents only use around 10 symbols (Table 5.1) regardless of the number of distractors. It suggests that the language they learn has little to do with the interpretation of concepts and properties, possibly due to the constraint of describing each graph in a single symbol, which does not allow compositionality. Such language might grant out-of-distribution generalisation to some examples in practice (given that several distinct in-distribution graphs are successfully described using a single symbol, the protocol might be robust enough to handle out-of-distribution examples) but it is not interpretable. One-word signalling is replaced with longer messages in Section 5.1.4, which leads to a more interpretable and compositional protocol.

Q3: The listener itself is an end-to-end differentiable architecture, which makes the gradients of the loss function with respect to its parameters easier to estimate than the parameters of the speaker, who is preceded by a discrete channel (Section 5.1.2.1, [104]). The final experiment from the initial exploration targets the question of whether the listener relies on the messages from the speaker in order to identify the correct input, or whether it learns on its own in an opaque way, without propagating the signal to the speaker. Figure 5.7 shows that for all words/symbols in the vocabulary the symbol sent by the speaker is the one that most frequently leads to the correct choice made by the listener (identification of the target among distractors). This suggests that the listener

actually cooperates with the speaker and the agents arrive at a graph-symbol mapping which allows them to solve the referential game.

Conclusion from the initial experiments using graph referential games Graph agents are able to recognise the target among distractors with accuracy of over 90% on new in-distribution test samples. The accuracy is decreasing with an increase in the number of distractors (**Q1**), the agents only use a fraction of the available symbols for vocabulary size $|V| > 10$ (**Q2**), and the agents converge to a mapping which most frequently allows them to solve the task (**Q3**).

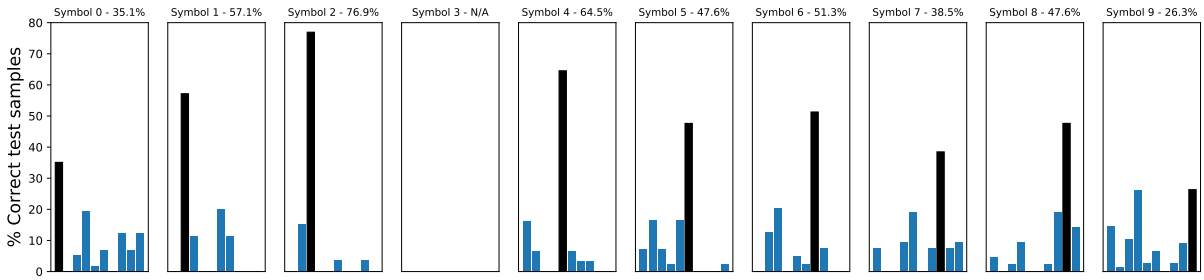


Figure 5.7: Robustness of the communication protocol learnt by the agents.
For each available symbol $i = 0, 1, \ldots, 9$ in the vocabulary of size $|V| = 10$, y-axis corresponds to the number of successful test samples – the samples where the listener correctly identified the target based on the symbol. Each subplot represents a distribution over the vocabulary. In each subplot the black bar corresponds to the symbol which the agents converged to after training (also in the subplot titles). The remaining blue bars correspond to the symbols that are not learnt in training, and only inputted to the trained listener to check whether it can correctly classify a sample based on the remaining symbols from the vocabulary. The above chart is for vocabulary size $|V| = 10$ and $|K| = 4$ distractors. There were no test samples found with the message $m = 3$ – the agents do not use the entire vocabulary (see Table 5.1).

5.1.4 The main experiments and analysis

This section presents experiments that aim to answer three research questions:

- **Q1:** What is the effect of data representation (of the target and the distractors) and the corresponding representation learning model on the compositionality of the emerged language (Section 5.1.4.1 and Section 5.1.4.2)?
- **Q2:** What is the effect of data representation and the corresponding representation learning model on the ability to generalise out-of-distribution (Section 5.1.4.3)?
- **Q3:** Can the listener identify the target if it receives a distorted message? (Section 5.1.4.4).

5.1.4.1 Qualitative analysis

Sample input data	Data representation		
	Bag-of-words	*Sequence*	*Graph*
A2 B4 C6	[1 4 4]	[5 1 3]	[7 1 4]
A2 B4 C5	[1 0 0]	[8 3 2]	[4 7 2]
A2 B2 C6	[6 4 1]	[9 9 1]	[9 4 1]
A5 B4 C6	[8 8 9]	[3 5 2]	[6 6 1]

Table 5.2: Random samples of input-message pairs across all three representations (in this game, there are $10 * 6 * 8 = 480$ distinct inputs in total).
In the example, the vocabulary size is $|V| = 10$ with messages of length $l = 3$ and three concepts of the respective property space being equal to $10, 6, 8$. Similarly as in Figure 5.4, the capital letters correspond to the concepts and the numbers correspond to the properties. We assume the language is order-invariant (for example, a message [7 1 4] is equivalent to a message [4 7 1]). In the games where graph representations and graph agents are used, varying one input property (for example, replacing 'C6' with 'C5' for the concept 'C') changes only one symbol in the message (the speaker replaces the symbol/word '1' with '2' in the utterance). In the games with sequential agents, changing the same symbol in the input leads to a change of two symbols in the transmitted messages ('5' → '8', '1' → '2'), and similarly for bags-of-words ('4' → '0', '4' → '0').

Table 5.2 shows a symbolic representation of the input data and the corresponding messages in Game-1. In this game, the length of the messages is equal to the number of concepts, which might act as an additional supervision and pressure the agents to represent each concept as one symbol (in more recent variants of referential games, for instance, Dessì et al. [69], vocabulary size is larger than the number of distinct concepts to avoid such supervision). The random samples illustrate how graph representations might induce a more compositional language. Due to the large input space, qualitative analysis has to be accompanied by quantitative analysis to draw conclusions.

5.1.4.2 Quantitative analysis: Topographic similarity

A quantitative metric is needed to discuss the entire language and draw conclusions. There is no definitive quantitative measure of language compositionality. However, *topographic similarity* (Figure 5.8) is most commonly used as a proxy for compositionality in language evolution studies [32] and in the existing research on referential games [163, 44]. The metric is applied in the graph referential games as follows (using agents that converged in terms of training accuracy):

1. All target inputs and the corresponding messages from the speaker are enumerated.

2. A vector of cosine similarities s between all pairs of the input objects is computed.

3. A vector of edit distances d between all pairs of the messages is computed.

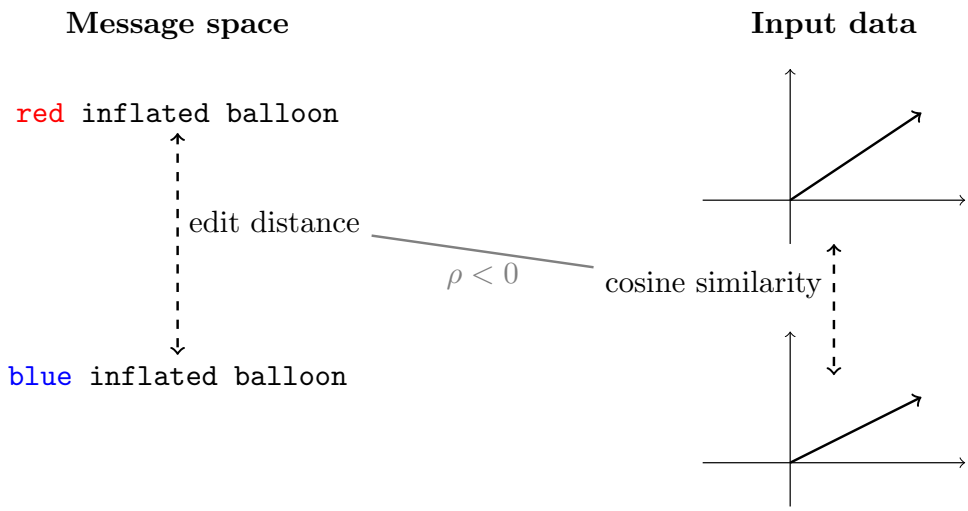


Figure 5.8: *Topographic similarity* ($-\rho$) is computed as the negative Spearman correlation between distances in the target space (inputs to the speaker) and the distances in the message space (outputs of the speaker) of all input-message pairs.
The negative correlation is used because of the negative relation between distance metrics and similarity metrics. In the example in the figure, similar messages (low edit distance) are expected to be inversely correlated ($\rho < 0$) with the similarity between their original representations in the input space (high cosine similarity; assuming the language is compositional). Topographic *similarity* is the negative of such correlation so that the higher the topographic similarity, the more similar the relations between the message and the input space are; that is, similar inputs are described in similar sentences.

4. The topographic similarity is equal to the negative Spearman correlation ρ between s and d.

The steps above are repeated for each data representation separately. In case of graph representations, the nodes are concatenated in the same order as in the corresponding sequences before computing cosine similarities. Across different numbers of distractors ($|K| = 19, 29, 49$) graph representations lead to a more compositional language, followed by sequential representations (Figure 5.9). A compositional language is more desirable as it broadens the scope of the concepts the agents can communicate, bringing them closer to the idea of 'making an infinite use of finite means' [121, 54]. This result further corroborates the hypothesis that a compositional language is more likely to emerge when the agents receive structured prelinguistic representations as input [163, 262].

5.1.4.3 Out-of-distribution generalisation

Does a higher compositionality translate to a better performance on out-of-distribution examples?

In *out-of-distribution* examples, both the target and the distractors are new to the agents, that is, the agents did not see any of them in the training data – however, they do follow

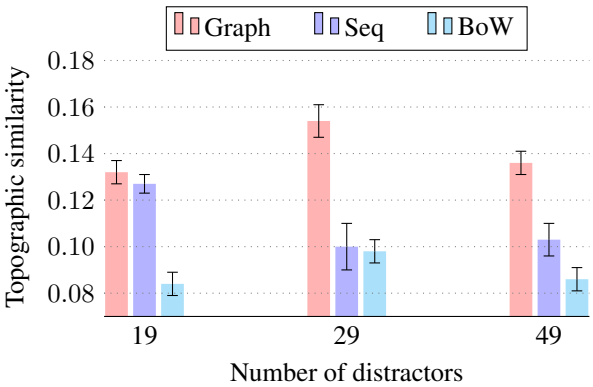


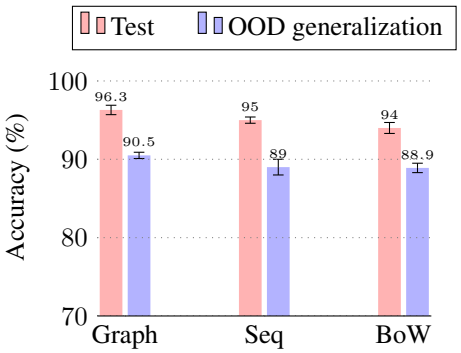
Figure 5.9: Topographic similarity in Game-1 with perceptual dimensions [10, 6, 8, 9, 10], a message length of size 3, and a vocabulary size of 50. The means and standard deviations are computed across five random seeds.

the same general structure as the training examples (the same number of nodes). Because of this implementation, both agents have to generalise out-of-distribution to successfully solve the test cases: the speaker has to learn to describe out-of-distribution graphs, and the listener has to learn to distinguish between the out-of-distribution graphs. This is a new approach to evaluation of the agents in referential games.

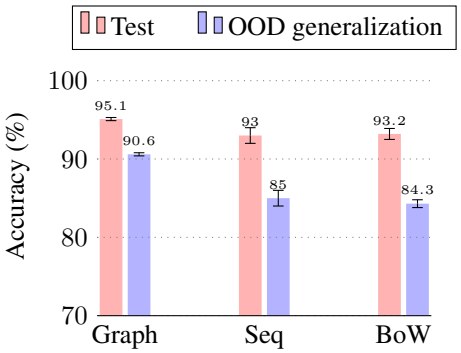
Figure 5.10 shows the results of experiments using $|K| = 9, 19, 49$ distractors. Graph representations outperform the baselines of a lower degree of structure, in particular as the number of distractors increases and the game becomes more difficult ($|K| = 19, 49$). This result suggests that graph representations and graph representation learning methods are able to better exploit transferable information in the training data, and apply it to new examples. This is an interesting result in the context of using referential games in data-scarce regimes, and it adds another argument in favour of the hypothesis that graph representations improve compositional generalisation in machine learning [19].

5.1.4.4 Do agents rely on the communication channel in solving the game?

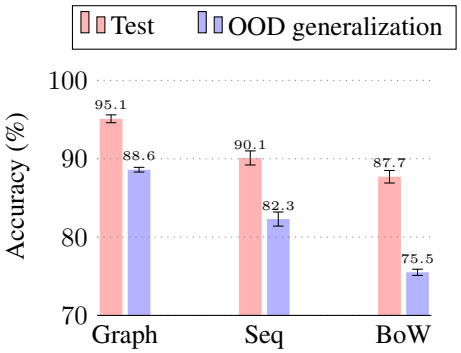
The final experiment is similar to the robustness test in Section 5.1.3, where the entire one-word message is varied to test if the listener relies on the message from the speaker to identify the target. Figure 5.11 shows the results of this analysis for longer messages ($l = 3$) being distorted by replacing the first symbol m_1 by each of the possible remaining symbols from the vocabulary of size $|V| = 10$. As shown in the Figure 5.11, for all emerged messages, distorting the first symbol leads to a decrease in test accuracy. The highest test accuracy attained by the agents in each case is dependent on receiving the original target encoding produced by the trained graph speaker. This suggests that the listener relies on the message in learning to solve the task, and individual symbols carry meaning that is useful for recognising the target.



(a) 9 distractors



(b) 19 distractors



(c) 49 distractors

Figure 5.10: Standard test accuracy (Test) and out-of-distribution (OOD) generalisation in Game-2 for graphs (Graph), sequences (Seq) and bags-of-words (BoW). Number of nodes, message length and vocabulary size are equal to 25. Mean and standard deviation is computed across three runs.

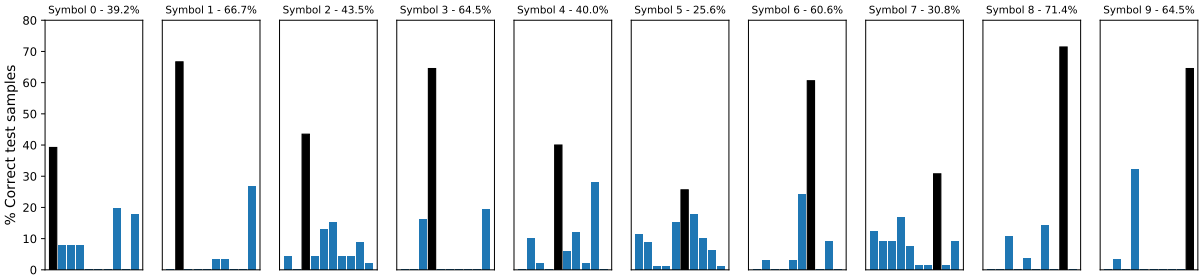


Figure 5.11: Robustness of the communication protocol in graph referential games with messages of the length $l > 1$.
Results shown for trained speakers and emerged messages from Game-1 with perceptual dimensions [10, 6, 8], a message length of size $l = 3$, and a vocabulary size of $|V| = 10$. In each subplot, the title corresponds to the symbol m_1 in the original message developed through playing the game. The dark bar corresponds to the number of correctly classified samples given the original message (also included in the subplot title, for example, Symbol 0: 39.2%). The light blue bars correspond to the results of replacing the first symbol m_1 in the messages with each of the remaining symbols from the vocabulary. The rest of the message (symbols m_2 and m_3) remains fixed.

Conclusion from the main experiments using graph referential games The choice of data representation for the target and distractors in referential games influences the compositionality of the emerged languages, as measured by topographic similarity (Figure 5.9) and qualitative analysis of samples (Table 5.2) (**Q1**), graph representations lead to a higher accuracy on out-of-distribution samples than sequences and bag-of-words, especially as the number of distractors and the difficulty of the game increase (Figure 5.10) (**Q2**), and the agents converge to a language which allows them to solve the task most frequently, with even one word changes leading to a decrease in test accuracy (Figure 5.11) (**Q3**).

The work presented in this section connects two previously disjoint subfields of machine learning: 1) multi-agent systems with a discrete channel (specifically, referential games, or *emergent communication*) and 2) graph neural networks. Introducing graph representations and graph representational learning methods to referential games opens a new line of research on communicating complex structures and relational information in multi-agent systems with a discrete channel. The main and most promising result is in out-of-distribution generalisation – graph representations lead to a higher out-of-distribution generalisation, which suggests that organising data as graphs can allow the agents to make a better use of the training dataset, and potentially reduce the number of training samples needed to successfully train the agents. This result echoes previous results on the generalisation potential of graph representations in machine learning [19], and structured data representations leading to a compositional language and better generalisation to new samples [262, 163].

5.2 Visual referential games

This section presents my work on out-of-distribution generalisation in referential games with images as input (*visual referential games*), using agents parametrised by *convolutional neural networks* [166]. In contrast to the previous section, this work refutes the initial hypothesis, which is still useful from the perspective of increasing the current state of knowledge on out-of-distribution generalisation in multi-agent games.

I first define the type of game considered in this problem – *population games*, in which the number of speakers and listeners is not always 1 – in Section 5.2.1. Afterwards, in Section 5.2.2 I describe the experiments conducted in the setting of visual referential games.

5.2.1 Population games

Population games are an extension to the standard referential game with N speakers and N listeners (where $N \geq 1$). At each optimisation step, one speaker and one listener are uniformly sampled and paired together. The chosen agents proceed as in the classic one-pair referential game and receive weight updates based on a batch of inputs. This process is repeated until all speaker-listener pairs converge in terms of training loss (or training accuracy).

5.2.2 Experiments

I consider the problem of using realistic images (taken from ImageNet) as inputs to referential games. My work is based on prior work by Dessì et al. [69]. The setup of their game is as follows: the sender receives as input a target picture, and it produces as output one symbol; the receiver network receives in input this symbol, as well as a list of n pictures – one of them being the input given to the send, and the others being distractors; receiver produces a categorical probability distribution over the images, representing the probabilities that each of the presented images is the target.

Dessì et al. conclude that the accuracy drops in the OOD set in comparison to the standard in-distribution set, yet it remains well above chance. The language that emerged in a visual referential game is partially interpretable with the use of a qualitative analysis (Figure 5.12) and a quantitative analysis (Table 5.3 and Table 5.4, *Trained from scratch* rows).

In contrast to the previous work focused on a single pair of agents in a referential game, the new work presented in this section increases the number of speakers and listeners.

In this section, I aim to answer the following research questions:

- The authors in the existing work train the agents (including the convolutional encoders) from scratch in a referential game. This warrants the question (**Q1**):

  How does training from scratch compare to using agents pretrained on ImageNet before they engage in a referential game?

  Using pretrained modules is a standard approach in large-scale machine learning applications with realistic image or language data [106, 306, 229] and it speeds up training, allowing the task complexity to increase.

- Dessì et al. [69] use a fixed vision model ResNet50 [106] as a convolutional encoder of the speaker and the listener. As a preliminary experiment before the main population experiment, I look into the effect of increasing the complexity of the vision module on generalisation and language properties to answer the question (**Q2**):

  How does the increase in the model complexity affect generalisation in a visual referential game?

- As in the previous section on graph referential games, inspiration is drawn from the literature on language evolution in biological systems. This time, the focus is on the number of speakers, with the goal of replacing the standard two-agent framework with a community (*population*) of speakers and listeners. Human languages are affected by the size of the linguistic communities, with a small number of language users leading to more complex languages, while larger communities give rise to easier and more transferable languages [191, 181]. This leads to the question of whether a similar pattern occurs in artificial multi-agent communication. Out of various properties of the emerged language that could be studied, such as whether it follows Zipf distribution, I am most interested in the question (**Q3**):

  How does increasing the number of speakers and listeners affect out-of-distribution generalisation in a visual referential game?

  My hypothesis was that increasing the number of agents would improve out-of-distribution generalisation: if there are more speakers and listeners that need to learn a shared language, over time they develop a more robust communication than if the community is smaller.

At the end of this section, I briefly discuss connections of this work to unsupervised learning and self-supervised learning.

Game setup The game presented in this section follows the referential game framework described in Section 5.1.2.1, with a few differences: 1) here, the speakers and the listeners

are parameterised by an example of a convolutional neural network, ResNet50 [106] 2) the input data (target and distractors) are realistic images from ImageNet (1000 image categories; examples of the images are shown in Figure 5.12) 3) the number of distractors $|K| = 128$, which means that the accuracy of a random choice model is only 0.8%.

Data The base work uses realistic images (1.3M natural images from 1K distinct categories from *ImageNet* [65]) as input to referential games. Apart from the standard in-distribution set included in ImageNet (the *ILSVRC-2012* validation set, containing around 50K images from the same categories as the training data), I use an out-of-distribution test set (OOD set) proposed by Dessì et al. [69].

The OOD set contains 80 categories that were neither in the training set nor hypernyms or hyponyms[2] of the classes present in the training set (for example, since images labelled as 'hamster' are present in the training set, the OOD set excludes not only the 'hamster' images but also the images labelled as 'rodent' and 'golden hamster'). However, the OOD categories are not entirely different: they belong to the same high-level domains as those used in training (for example, images of fish, images of furniture). The similarity between these images is controlled using WordNet-derived hierarchy [76] (in ImageNet, each node from WordNet is depicted by hundreds and thousands of images).

5.2.2.1 Pretrained vision modules in the game by Dessì et al. (Q1)

Experimental method The results obtained by Dessì et al. [69] are compared with the results obtained in the same setup by replacing *tabula rasa* agents (trained from scratch) with the agents equipped with pretrained vision modules, as this is a common practice in machine learning tasks that rely on a large, realistic dataset. If the pretrained modules can be safely used in visual referential games, it might speed up training in the costly population experiments where 2+ agents are trained.

All axes of variation and evaluation procedures in Table 5.3 follow the experiments by Dessì et al. [69] apart from the new comparison between using trained from scratch and pretrained convolutional modules. The parameters $\pm$*augmentations* mark whether the game was trained using data augmentation and $\pm$*shared* indicates whether the weights of the vision module are shared between the speaker and the listener. In the 'pretrained' option, the vision modules of both the speaker and the listener are frozen during the game, and so it only makes sense to share the module between the speaker and the listener (+shared option). In case of the agents trained from scratch, sharing visual module was found to make little difference in terms of test and OOD test accuracies (Table 5.3),

[2] A word A is a hyponym of word B, and B is a hypernym of A, when the concept referred to by A is subsumed by the concept referred to by B. For example, 'mammal' is a hyponym of 'animal'.

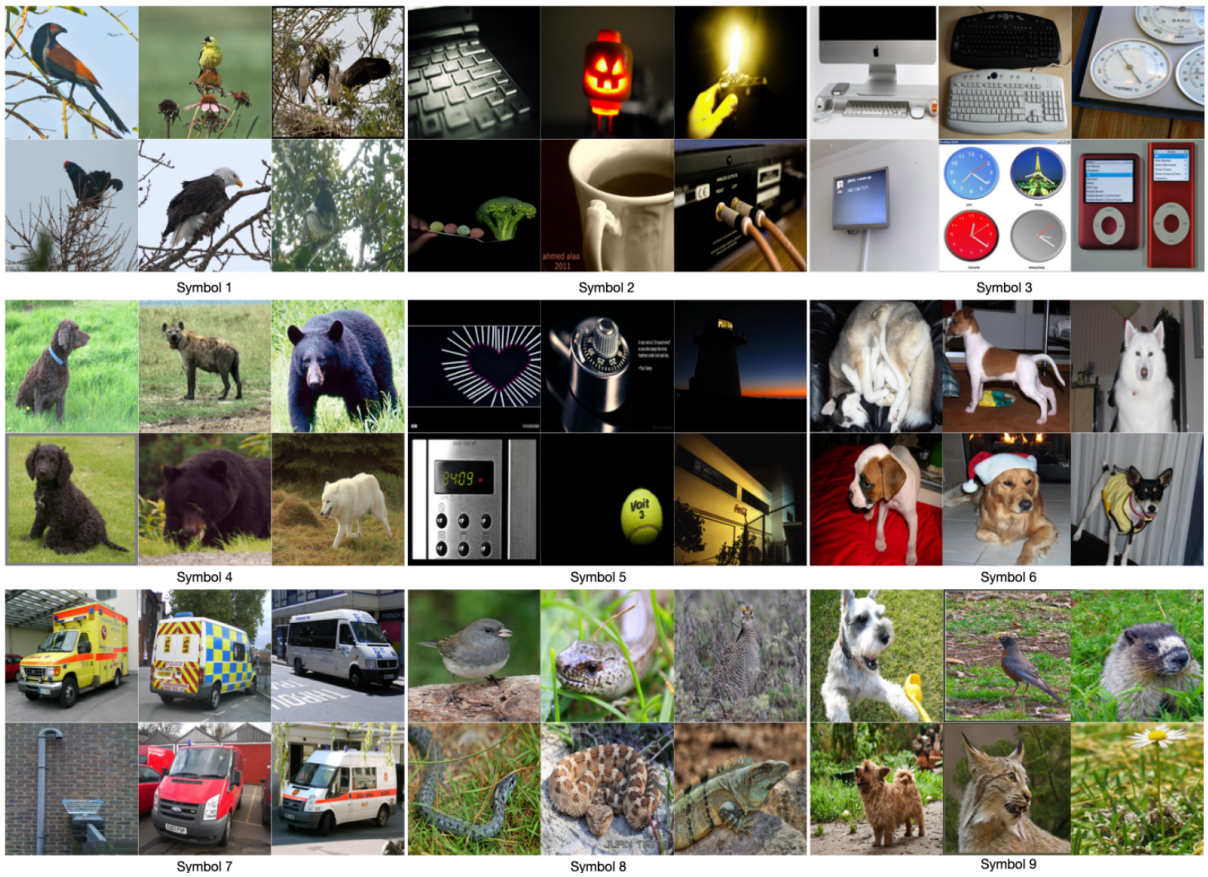


Figure 5.12: Qualitative analysis of a game with single-word messages by Dessì et al. [69]. The agents learn to assign the words (symbols) to the input images in an interpretable way: for instance, Symbol 1 corresponds to images of birds on branches, and Symbol 3 corresponds to human artifacts with flat shapes. There is no direct supervision encouraging the agents to learn such mappings – as in the previous section on graph referential games, the agents are only rewarded for recognising the target among distractors. The capacity to learn this level of abstraction (for example, grouping all dogs under one symbol rather than learning different symbols for each dog breed) might help the agents generalise with respect to fine-grained ImageNet categories – even if the 'poodle' category is excluded from the training dataset, the agents might be able to successfully recognise such images at test time as they learn to recognise high-level dog features (Symbol 6).
Figure produced by Dessì et al. [69].

which is encouraging from the perspective of using diverse architectures in a referential game. This is particularly relevant in the cases when the agents would not be able to share weights. For example, in future applications where models developed by different companies learn to coordinate on a shared task, there may be no access to their internal parameters due to the use of proprietary systems by different companies.

Gaussian Blobs [163, 29] (previously described in general in Section 2.3.1.3) refers to a sanity check of whether the agents can successfully describe and recognize images of pixels drawn from the standard Gaussian distribution. The goal is to make this unlikely: agents succeeding in this task are likely to be directly communicating the values of individual noisy pixels as opposed to devising a general language. The higher the accuracy in the game with blobs of Gaussian noise, the more the agents rely on low-level uninterpretable aspects of images. Instead, the goal is to encourage them to describe semantic information, such as ImageNet categories or the features that are human-interpretable (Figure 5.12). The baseline (chance) performance in this game is 0.8%.

The agents were trained using Layer-wise Adaptive Rate Scaling (LARS) [296], an extension of Stochastic Gradient Descent (SGD) which improves training using a larger batch size (useful when training a model on a large dataset).

Results Using a pretrained vision module instead of training the agents from scratch leads to comparable results in terms of the standard test accuracy and OOD generalisation in ImageNet (Table 5.3: `-augmentations +shared` setup with and without the pretrained module). A pretrained module seems to be significantly more robust to Gaussian noise than the same model trained from scratch, without data augmentations (the pretrained module leads to a close to random choice performance on Gaussian Blobs, which is what is hoped for in successfully trained agents, as in the Gaussian test the agents do not have any semantic information to rely on). This suggests that using a pretrained module prevents a model from adopting degenerate strategies based on low-level pixel information.

In the original game [69], the setup of `-augmentations, +shared` gave the highest results of all the considered configurations, apart from being the most susceptible to developing an opaque protocol. Using a pretrained module alleviates the last problem.[3]

The agents evaluated in Table 5.3 can be used to analyse the mapping between images (their labels) and the corresponding utterences by a trained speaker. In order to better understand the results of using pretrained vision modules, I repeat the quantitative analysis

[3]LARS improves the performance in the visual referential game with ImageNet (the results with a pretrained vision model and a standard SGD without LARS are *ILSVRC-val*: 85.1%, *OOD set*: 84.3%, *Gaussian Blobs*: 0.78%). The main conclusion regarding the use of a pretrained module to avoid a protocol based on pixel-level information holds, regardless of employing LARS.

Model	Test set		
	ILSVRC-val	*OOD set*	*Gaussian Blobs*
Pretrained vision module:			
`-augmentations, +shared`	92.8%	92.6%	1.6%
Trained from scratch:			
`-augmentations, -shared`	91.2%	90.8%	43.4%
`-augmentations, +shared`	92.8%	92.7%	84.7%
`+augmentations, -shared`	81.5%	72.0%	0.8%
`+augmentations, +shared`	82.2%	73.7%	0.8%

Table 5.3: Accuracy in the referential game using the full ImageNet dataset.
ILSVRC-val, *OOD set* and *Gaussian Blobs* refer to the datasets used by Dessì et al. [69]: namely, a validation set containing around 50K images from the same categories as the training data, a custom 'out-of-distribution' validation set containing unseen image categories, and a sanity check using blobs of Gaussian noise as targets and distractors [29]. The first row of values contains the results of experiments I ran, whereas the remaining rows are taken from Dessì et al. [69] and are included for the sake of comparison.

Model	nMI		$\lvert P \rvert$	
	ILSVRC-val	*OOD set*	*ILSVRC-val*	*OOD set*
Pretrained vision module:				
`-augmentations, +shared`	0.52	0.47	1137	1127
Trained from scratch:				
`-augmentations, -shared`	0.5	0.45	2044	1921
`-augmentations, +shared`	NS	NS	2048	2025
`+augmentations, -shared`	0.58	0.53	2042	1752
`+augmentations, +shared`	0.56	0.51	2046	1765

Table 5.4: Protocol analysis (normalised mutual information nMI and the observed protocol size $|P|$).
NS corresponds to the cases where the obtained scores were not significantly different from chance according to a permutation test. In the pretrained experiment (unlike in its trained from scratch counterpart) the obtained scores pass the significance permutation test. The remark in Table 5.3 applies here as well.

from Dessì et al. [69] on the communication protocol that emerged in the game where the speaker and the listener share a pretrained convolutional network.

Normalised mutual information (nMI) is computed by dividing mutual information between the ground-truth image labels and the messages produced for those images by the average entropy of messages and labels:

$$\text{nMI}(X, Y) = \frac{I(X;Y)}{\frac{1}{2}(H(X) + H(Y))}.$$

The value of nMI ranges between 0 and 1 (since $I(X;Y) \leq H(X)$ and $I(X;Y) \leq H(Y)$). The metric denotes whether the agents' language associates symbols with human-intepretable categories from ImageNet. The protocol size $|P|$ is the observed protocol size at test time, that is, the number of distinct symbols the speaker uses on the test/OOD set.

The language analysis in terms of normalised mutual information (Table 5.4) suggests that even though a pretrained module leads to a more interpretable language than when training from scratch (nMI = 0.52 and nMI = 0.47 vs the results not passing the significance test [69] in the trained from scratch counterpart), data augmentations are still beneficial in learning to express semantic content of the image (the highest nMI for the +augmentations, -shared configuration).

The pretrained variant is using fewer symbols than the models trained from scratch ($|P| = 1137$ and $|P| = 1127$ vs $|P| = 2044$ and $|P| = 1921$). Since the vision module is pretrained on ImageNet, the learned visual features might be disentangled in a way that helps to disambiguate the actual number of ImageNet classes (1000). This might also explain its robustness to learning low-level image information.

In conclusion, using a pretrained visual module is a valid choice in visual referential games. It combines the advantages of training without data augmentations (high in-distribution and out-of-distribution test accuracies; Table 5.3) with the advantages of training with data augmentation (robustness to non-intepretable pixel-based information and higher mutual information between the image labels and the symbols; Table 5.3 and Table 5.4).

5.2.2.2 Model complexity in visual referential games (Q2)

Table 5.5 shows the results of increasing the complexity of the visual module in a one-pair visual referential game. There is practically no difference when introducing a larger number of layers, which suggests that ResNet50 is the appropriate choice due to a smaller computational complexity.

ResNet size	Train acc [%]	Test acc [%]	OOD test acc [%]	GB [%]	Conv. ep.	nMI	overlap	$\|P\|$	O-nMI	O-overlap	O-$\|P\|$
50	100	92.67	92.51	0.78	3rd	0.52	100	2048	0.47	100	2018
101	100	92.41	91.76	0.78	3rd	0.52	100	2048	0.47	100	2010
152	100	92.16	91.82	0.78	3rd	0.51	100	2048	0.47	100	1996

Table 5.5: The effect of changing the ResNet size (50, 101, 152) in the one-pair visual referential game. The first column corresponds to the number of layers in a ResNet architecture, with ResNet50 corresponding to the network with 50 layers etc. The metrics with the prefix O correspond to the analysis of the communication protocols using out-of-distribution examples. The column Conv. ep. indicates the epoch, in which the agents reach 100% of accuracy. The column GB corresponds to the accuracy in the Gaussian sanity check, computed in the same way as in Table 5.3. nMI values report normalised mutual information between the image labels and the messages produced for those images, as in Table 5.4. Overlap corresponds to the percentage of symbols that the speakers-listeners pairs share (100% in the game with one speaker and one listener, as in this case there is only one emerged language). Protocol size $|P|$ is computed as in Table 5.4. Vocabulary size in all games is equal, with $|V| = 2048$, following the approach of a weaker supervision in developing the input-symbol mappings ($|V| >> |K|$, where $|K|$ is the number of distinct classes in the dataset and $|K| = 1000$).

5.2.2.3 Population size and out-of-distribution generalisation (Q3)

This section investigates the effect of increasing the number of speakers and listeners on generalisation in a visual referential game.

Table 5.6 shows the results of increasing the number of speakers and listeners in terms of the performance and language analysis. The performance is measured using accuracy and the language properties are measured using normalised mutual information nMI, the overlap between the languages, and the number of distinct symbols used at test time. The metrics other than train accuracy and convergence time correspond to the analysis using test data on three different test sets: in-distribution, out-of-distribution and the Gaussian sanity check. The axis of variation is the number of speaker-listener pairs used in the game.

In terms of the in-distribution performance, there is a small decrease in test accuracy with the increase in the number of agents. There is also a decrease in the normalised mutual information. A similar pattern can be observed in the out-of-distribution case.

The main conclusions from this experiment are:

- It is possible to successfully train a larger community of speakers and listeners to solve a visual referential game with realistic images. Both in-distribution and out-of-distribution test accuracy values are above 90%.
- Despite the high test accuracy, normalised mutual information does not exceed 0.52 in any configuration (for a value ranging from 0 to 1). This means that the agents learn a different mapping than the ground-truth image-label mapping from ImageNet. However, this likely plays to their advantage when evaluated on the OOD set, which contains images from the categories unseen in training, yet belonging to the same high level categories (for example, images of plants, fish and furniture). This communication protocol is intuitively similar to how humans describe new data, for example, a person not versed in marine biology might describe several species of fish as 'fish' instead of distinguishing them with the granularity of their Latin names.
- Unlike in biological systems, where the increase of the community size leads to a more structured and more comprehensible language, there seems to be no such pattern in visual referential games. The results across the games with 1, 9 (3×3) and 25 (5×5) speaker-listener pairs are comparable in terms of accuracy and language interpretability (as measured by nMI and the Gaussian sanity check), with a possible trend of a decline in performance as the number of agents grows. The initial hypothesis inspired by the biological multi-agent systems turned out not to apply in the artificial multi-agent systems.

Nr of agents	Train acc [%]	Test acc [%]	OOD test acc [%]	GB [%]	Conv. ep.	nMI	overlap	$\|P\|$	O-nMI	O-overlap	O-$\|P\|$
1×1	100	92.67	92.51	0.78	3rd	0.52	100	2048	0.47	100	2018
3×3	100	91.6 ± 0.1	91.0 ± 0.1	1.46	3rd	0.51 ± 0.0	74.1	2048	0.45 ± 0.0005	75.1	2012.7
5×5	100	90.89 ± 0.0006	90.2 ± 0.002	1.56	4th	0.51 ± 0.00004	62.96	2048	0.45 ± 0.0005	65.35	2008.8

Table 5.6: The effect of the population size on the in-distribution and out-of-distribution performance in a visual referential game. The first column corresponds to the number of speakers and listeners. The metrics with the prefix O correspond to the analysis of the communication protocols using out-of-distribution examples. The column Conv. ep. indicates the epoch, in which the agents reach 100% of accuracy. The column GB corresponds to the accuracy in the Gaussian sanity check, computed in the same way as in Table 5.3. nMI values report normalised mutual information between the image labels and the messages produced for those images, as in Table 5.4. Overlap corresponds to the percentage of symbols that the speakers-listeners pairs share (100% in the game with one speaker and one listener, as in this case there is only one emerged language). Protocol size $|P|$ is computed as in Table 5.4. Vocabulary size in all games is equal, with $|V| = 2048$, following the approach of a weaker supervision in developing the input-symbol mappings ($|V| >> |K|$, where $|K|$ is the number of distinct classes in the dataset and $|K| = 1000$). The error bars for accuracy and language analysis are computed by averaging the values for all 9 speaker-listener pairs in the 3×3 scenario, and across all 25 speaker-listener pairs in the 5×5 case (hence they are omitted in the 1×1 case).

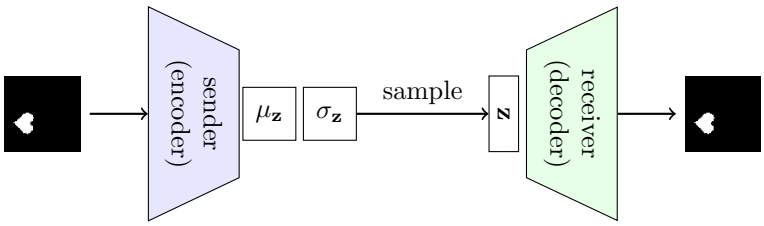


Figure 5.13: Variational Autoencoder (VAE) [139] cast as an image reconstruction game with a message $\mathbf{z}$.

5.2.2.4 Discussion: Links to unsupervised and self-supervised learning

This chapter focuses on referential games, which can be seen as a classification problem over a set of the target input and distractors. Unlike in standard classification, a referential game comes with an additional challenge of a discrete bottleneck through which the weight updates propagate from the listener to the speaker. Since the agents are trained to distinguish the target object from the distractors, they tend to use shortcuts that allow them to solve this task, instead of learning a comprehensive input representation that could be used across different downstream tasks. For instance, in Section 5.1.3, the agents in a one-word graph referential game converge to a language with a vocabulary smaller than the total number of unique input graphs, which means that the agents learn to solve the task by mapping several graphs to a single symbol.

An emergent communication game that lends itself better to the idea of learning reusable discrete representations is referred to as reconstruction game [44] (Figure 5.13). In this game, the listener learns to reconstruct the original input representation based on a discrete message from the speaker. This is similar to unsupervised methods such as Variational Autoencoder (VAE) [139], where the latent representation can be later reused in tasks such as image generation [227]. Autoencoders can be investigated through the lens of emergent communication, using tools such as topographic similarity and disentanglement metrics proposed in the context of reconstruction games [44].

A link between visual referential games and self-supervised learning [39] has been investigated by Dessì et al. [69]. The authors evaluated the speaker in a visual referential game (described in Section 5.2.2) as a feature extractor on external classification tasks. They found that the performance of this feature extractor is comparable to SimCLR [48], a self-supervised method. They also argue that data augmentation benefits both emergent communication and self-supervised learning, in particular in the context of image data. Future work should focus on contrasting the benefits and limitations of a discrete information bottleneck used in emergent communication with those of a continuous representation used in SimCLR.

5.3 Summary

This chapter presents my contributions to the field of multi-agent systems with a communication channel. This aspect of machine learning requires the ability to generalise out-of-distribution to be deployed in real applications (for example, Internet of Things), as the listeners have to adapt to the speakers' actions.

Section 5.1 presents a new framework developed with the main goal of studying the influence of data representation on the ability to generalise in referential games, and with a further objective of expanding the research on how to represent and communicate relational information. This work connects the previously disjoint fields of graph representation learning and emergent communication. The experimental results corroborate the hypothesis that graph representations lead to a better compositional generalisation in machine learning [19], in the context of multi-agent systems with a discrete communication channel. This work is also grounded in the previous research on language evolution in biological and artificial systems [262, 163], confirming that more structured inputs lead to more structured/compositional language. From initial experiments we conclude:

- graph agents are able to recognise the target among distractors with accuracy of up to 90% even when using one-word messages;
- the accuracy is decreasing with the increase in the number of distractors;
- agents converge to a mapping which most frequently allows them to solve the task.

The main experiments show that:

- graph representations improve the compositionality of the emerged languages as measured by topographic similarity;
- graph representations lead to a higher accuracy on out-of-distribution samples than the counterpart representations of sequences and bags-of-words, especially as the number of distractors increases;
- agents converge to a language which allows them to solve the task most frequently, with even single-symbol distortions leading to a decrease in test accuracy.

The results suggest that organising data as graphs allows the agents to make a better use of the training dataset, which can potentially reduce the number of training samples needed to successfully train the agents in a referential game.

Section 5.2 shows the results of taking on the challenging topic of using realistic images in a referential game. The work builds upon Dessì et al. [69], where the authors use ImageNet in a visual referential game. Our new experiments show that using a pretrained

visual module is a valid choice in visual referential games, and can replace the costly data augmentation. The experiments varying the population size show that:

- it is possible to train a larger community of speakers and listeners with test accuracy above 90% both on the in-distribution and out-of-distribution examples;
- the agents do not reproduce the ground-truth ImageNet labels, which might be correlated with their ability to describe high-level domains and perform well on the out-of-distribution set;
- the increase in the population size does not lead to an increase in in-distribution or out-of-distribution performance and language interpretability.[4]

This chapter plays with the modularity theme at the environment level – instead of using a single agent, there are at least two neural networks separated with a discrete communication channel. This setup can be thought of as a modular system with the modules communicating in order to solve a shared task. Section 5.1 and Section 5.2 answer two respective parts of the last research question formulated in Section 1.2: *Do graph reprentations induce a better OOD generalisation in multi-agent games? Does increasing the number and diversity of agents improve OOD generalisation?* Regarding both parts of this question, the results in this chapter corroborate relevant findings and hypotheses in existing literature. At the same time, they expand what has already been done by (1) providing the first set of results of using graph representations in emergent communication, and comparing them with corresponding baselines in terms of OOD generalisation and language compositionality, (2) extending the work by Dessì et al. [69] with a population setup using pretrained visual modules.

[4]After these experiments were completed, the work of Rita et al. [232] appeared with a similar conclusion in the context of reconstruction games, where the listener reconstructs the original target rather than recognising it among the distractors. The authors find that language properties, including the ability to generalise to unseen objects, are not enhanced with an increase in the size of homogenous populations.

Chapter 6

Conclusion and further directions

So long, and thanks for all the fish.
–Douglas Adams (1984)

This dissertation presents the outcomes of research conducted during my PhD. My goal has been to contribute to the field of out-of-distribution generalisation in machine learning from multiple angles. The motivation behind this research goal stems from the astonishing shortcomings of existing methods when evaluated on new data [249, 40]. In order for the machine learning systems to be truly intelligent and reliable in practical applications, they need to become more robust to distribution shifts that come naturally from using a machine learning system over time and applying it to real, changing data. The work presented in this thesis answers some of the research questions posed in Chapter 1 and it uncovers new avenues for further research on out-of-distribution generalisation in machine learning.

6.1 Thesis summary

The first research effort (Chapter 3) is model-agnostic and it focuses on linear data and theoretical results, which makes it a natural beginning of the story on out-of-distribution generalisation in increasingly complex learning scenarios. Chapter 3 focuses on the approach of *learning from multiple training distributions*. I contribute to this well-established line of research on out-of-distribution generalisation [147] by (1) introducing Linear unit tests, a new set of tasks that probe the algorithms in their ability to learn invariant features while ignoring spurious features, which is one of the approaches to improving out-of-distribution generalisation by discarding the irrelevant noise in data, (2) introducing new theorems that explain the connection between assigning appropriate weights to training examples

and Distributionally Robust Optimisation (DRO) [21], the objective that represents the pessimism-based model of out-of-distribution generalisation [7]. Since publication, Linear unit tests presented in this thesis have been used to evaluate and compare new algorithms in research on out-of-distribution generalisation [248, 50, 136, 72, 184, 49, 80, 283, 64, 200] and learning invariant features [171, 51, 207]. The results on DRO fill an existing gap between the line of research on data-centric approaches and the line of research on algorithmic approaches to generalisation and fairness.

The second contribution (Chapter 4) focuses on the single-agent scenario under the assumption of a single training distribution, which is the standard learning setup in machine learning. Unlike Chapter 3 – which uses mathematical tools and unit tests where invariant and spurious features can be perfectly separated – Chapter 4 explores the angle of out-of-distribution generalisation to latent, higher-level features such as colour and shape. Chapter 4 approaches the goal of the thesis from the perspective of *image classification*, the application of machine learning that arguably enabled the resurgence of neural networks in late 1980s and their current popularity. This chapter contributes to the existing work on out-of-distribution generalisation in image classification by (1) introducing Neural Function Modules and new results that show the advantages that Neural Function Modules bring in the context of out-of-distribution generalisation, and (2) releasing a set of simple image datasets that can be used as a stepping stone for evaluation and comparison of new models and algorithms in the context of out-of-distribution generalisation in image classification, including the tasks of relational reasoning and multi-object classification. The results in Chapter 4 show my individual work based on my previous collaborative work on Neural Function Modules [159]. Chapter 4 also explores Dilated DenseNets as another approach to improving out-of-distribution generalisation through *multi-level feature aggregation*. Here, the results are not as encouraging outside of the context of relational reasoning, which suggests that Neural Function Modules are a more promising direction when tackling distribution shifts in broad image classification. Chapter 4 also provides the first results on *compositional generalisation* (Section 2.4.2), which is later explored in the context of multi-agent games (Chapter 5).

Chapter 5 seeks the thesis goal in the learning scenario of multi-agent interaction, with focus on *multi-agent communication* with both one-to-one and many-to-many interactions. This chapter contributes to the existing literature on generalisation and compositionality in multi-agent games in two major ways. First, it introduces graph referential games along with the results on the influence of data representation and the corresponding data representation learning models on out-of-distribution generalisation. These results connect the previously disjoint fields of graph representation learning and emergent communication. Second, it provides a negative result in the challenging domain of population-based communication

grounded in realistic images. While we show that it is possible to train a population of agents with a high accuracy on both in-distribution and out-of-distribution examples, the increase in the population size does not lead to an increase in out-of-distribution generalisation unlike it is hypothesised to be the case in human populations [191, 181]. This chapter investigates research questions at the intersection of several ideas that are important from the perspective of out-of-distribution generalisation in complex scenarios: data representation and representation learning, efficiency as the drive for emergence of reusable representations [224, 195, 242, 16, 70], and compositional languages. The results corroborate the previous hypotheses that (1) more structured input data lead to more structured/compositional language [262, 163] and (2) graph representation learning can improve compositional generalisation in machine learning [19].

The theme of modularity has turned out to be a fruitful source of inspiration. The benefits of modularity in the context of out-of-distribution generalisation are most apparent *at the architecture level* in Chapter 4, where Neural Function Modules consistently outperform the appropriate baselines. The research questions posed in Section 1.2 have been explored, which has led to tangible contributions to machine learning that I presented and discussed in the core research chapters.

6.2 Further work

Out-of-distribution generalisation is one of the most challenging and pressing issues in machine learning. The core research chapters in this dissertation cover a broad spectrum of machine learning tasks and aim to foster future research on out-of-distribution generalisation from multiple perspectives.

In the context of learning from multiple distributions, the crucial effort is directed at obtaining fair and reliable metrics of performance, as the average-based metrics and objectives that are typically used in single-distribution learning fail to capture the nuances of learning from multiple heterogeneous data sources (Section 3.2). Research in this direction should focus on providing more tools for a reliable and fine-grained evaluation of algorithms that learn from multiple distributions. The work of De Bartolomeis et al. [17] is a step in this direction based on using and extending Linear unit tests. Another future avenue of research in the context of learning from multiple distributions is to investigate the connection between Distributionally Robust Optimisation (as well as related recent methods that aim to improve worst-case error [8, 178, 124]) and *federated learning* [81, 115], as these approaches share the setup and the motivation of learning from disjoint and heterogeneous data sources. Distributionally Robust Optimisation can address the *Inter-client* data heterogeneity [132] which occurs in the applications of federated

learning; for example, in machine learning-driven drug discovery based on data produced by different pharmaceutical companies with their own screening methods. Finally, the related quests of improving out-of-distribution generalisation and learning invariant features across multiple distributions can benefit from the recent advances in causal machine learning from heterogeneous data [226], which will provide tools to answer questions beyond the associative statistical relations modelled by standard supervised learning methods.

The future directions presented above in the context of learning from multiple distributions (reliable evaluation tools, filling the gaps between different approaches to closely related problems, causal machine learning) are equally worth exploring in the context of out-of-distribution generalisation in complex domains such as image classification and multi-agent games. In both of these domains, *modularity* appears to be a promising concept [90]. In the single-agent setting, modularity can be incorporated into the agent architecture by encouraging a neural network to arrive at specialised layers, as it is the case in Neural Function Modules. Multi-agent games with a discrete channel are another instance of replacing a standard monolithic neural network with interactive components. Finally, *graph representations* incorporate modularity at the data level, and they have been shown to improve out-of-distribution generalisation (Chapter 5). Modularity as a learning paradigm is further motivated by neuroscience, with the *global workspace theory* positing that the brain is composed of a set of specialised modules that communicate using a shared 'language'. Future research should explore different ways of incorporating modularity in machine learning systems at different levels to improve out-of-distribution generalisation [202, 288, 84, 277]. For instance, future research questions might concern hierarchical vs relational connections in modular data and in modular architectures, as well as new tools to dissect the learning process based on entangled, noisy data such as images and sound.

Finally, this thesis is focused on *zero-shot out-of-distribution generalisation* – we assume no access to any samples from the future OOD test distribution. In certain applications, it is possible to access a small number of samples from a new distribution, and use them to update the trained model so that it takes into account both the original training distribution and the scarcely represented new distribution [73, 18, 284, 34, 188]. In my future work, I aim to contribute to this line of research on generalisation in machine learning.

6.3 Contemporary challenges and out-of-distribution generalisation

At the time of submission of this thesis for examination, the ubiquity of machine learning algorithms in people's everyday lives is entering public consciousness. This is attracting many legal, ethical, and political concerns regarding how models are used and trained. Discussions about bias and out-of-distribution generalisation might become more prevalent in the coming years. As this is mainly a thesis with a focus on technical contributions, and not their effect on society, I will not weigh in on these questions. However, I will briefly give my view on how the contributions in this thesis and out-of-distribution generalisation research relate to contemporary challenges in machine learning.

Large language models In 2022, OpenAI released ChatGPT[1], primarily a chatbot application utilising the GPT-3 *large language model* (LLM), and attracted widespread public attention. In January 2023, ChatGPT had around 100 million active users. Shortly thereafter, competitors released similar LLM-based products across various applications, with the most promising being those dealing with information retrieval such as advanced search engines. With all this interest, reports appeared of unexplainable strange responses given by large language models to their users, with popular press often anthromorphising such 'behaviour'.[2] [261, 233] While part of the reason for these responses stemmed from users' fundamental misunderstanding of the role and capabilities of LLMs, these issues – as well as reproducing human biases, despite the developers' best efforts – are instances of failure of OOD generalisation.

Large language models are based on training a Transformers-derived model on unprecedented amount of natural language data. Central in these architectures is attention, which is also key in Neural Function Models (introduced in Chapter 4). In this thesis, my hypothesis was that attention blocks help with out-of-distribution generalisation tasks in visual reasoning (using architectures based on convolutional neural networks), and there is evidence supporting this claim. The reported OOD failures in LLMs show that, contrary to the statement made in the title of a famous paper, attention is *not* all you need when dealing with distribution shift. Due to the closed-source nature of some of these LLMs, it is difficult to determine whether and how they attempt to deal with OOD generalisation. I believe this question will become more relevant in the coming years.

[1] chat.openai.com

[2] As an article in Science by Mitchell [199] discusses, humans are prone to anthromorphising AI – we talk about 'machines that think' and see intelligence in systems that show the slightest linguistic competence. While possibly interesting from a science-fiction or futurist perspective, this is not helpful in considering true ethical implications of AI. Even in the most society-oriented discussions on AI, I think Dijkstra's remark applies: 'the question of whether machines can think is no more interesting than the question of whether a submarine can swim' (paraphrased).

Vision Some time before the widespread media attention given to ChatGPT, the most visible area of machine learning put into practice were vision applications. Of particular attention were *self-driving vehicles* and *generative AI in images and video*. These still attract a lot of interest given their potential to change many industries. From a non-technical and business standpoints, the issues raised regarding labour, copyright, legal liability and ethics have been the most salient and notable. Particular attention has been paid to the danger of AI-generated images and video in spreading fake news. However, I will consider those issues that have technical definitions and can be observed as instance of distribution shift.

The safety and reliability of autonomous vehicles is a somewhat contentious topic, for reasons already alluded to. An important consideration is the data they were trained on and how compliance with local laws and customs regarding driving can be achieved, in the interest of all road users. Models which determine the behaviour of a vehicle ought to consider a variety of settings in which a vehicle operates: a sparsely populated car-centric city, cities with widely used public transport, pedestrisan-friendly streets, low-visibility residential neighbourhoods with many parked vehicles, mountainous and curved rural roads... The fact that the majority of training images in some models seem to originate from high-visibility daytime photos leads to entertaining failure modes – for example, a full moon is interpreted as an amber traffic light by a self-driving car [190]. In my view, the presence of such multiple environments is a good opportunity for considering questions raised in Chapter 3 in the interest of public safety.

Text-to-image systems (e.g., Dall-E [225]) can exhibit harmful societal biases, such as associating gender and ethnicity with attributes such as criminality or profession. A recent paper by Luccioni et al. [180] analyses the lack of diversity of generated images of professionals. This is yet another example in which methods such as distributionally robust optimisation might be applied, while keeping in mind the usual caveats with the ways they should be applied (for example, by considering the practical recommendations given in Section 3.4.3).

My aim in writing this thesis was to shed light on challenges in out-of-distribution generalisation across a wide spectrum of machine learning problems and settings. I considered three paradigms under which out-of-distribution data can be precisely defined: learning from multiple distributions, visual reasoning, and multi-agent communication. In all of these cases, I considered whether some notion of modularity helps agents generalise. In the simplest words, this problem is hard, and the proposed solutions come with many

caveats. Nevertheless, the overall conclusion I draw from this research is that modularity – in data representation, environment, or model design – helps tackle distribution shift. In my opinion, there are many opportunities to build upon this thesis, especially in light of contemporary widespread use of machine learning applications. If I have convinced the reader that the proposed methods and the problems they address are interesting and significant, then I have fulfilled my goal.

Bibliography

[1] Ashutosh Adhikari, Xingdi Yuan, Marc-Alexandre Côté, Mikuláš Zelinka, Marc-Antoine Rondeau, Romain Laroche, Pascal Poupart, Jian Tang, Adam Trischler, and Will Hamilton. Learning dynamic belief graphs to generalize on text-based games. *Advances in Neural Information Processing Systems*, 33:3045–3057, 2020.

[2] Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.

[3] Kartik Ahuja, Karthikeyan Shanmugam, and Amit Dhurandhar. Linear regression games: Convergence guarantees to approximate out-of-distribution solutions. In Arindam Banerjee and Kenji Fukumizu, editors, *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, volume 130 of *Proceedings of Machine Learning Research*, pages 1270–1278. PMLR, 2021. URL http://proceedings.mlr.press/v130/ahuja21a.html.

[4] Ekin Akyürek, Afra Feyza Akyürek, and Jacob Andreas. Learning to recombine and resample data for compositional generalization. *CoRR*, abs/2010.03706, 2020. URL https://arxiv.org/abs/2010.03706.

[5] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias: There's software used across the country to predict future criminals. and it's biased against blacks. *ProPublica*, 2020.

[6] Antreas Antoniou, Agnieszka Słowik, Elliot J Crowley, and Amos Storkey. Dilated densenets for relational reasoning. *arXiv preprint arXiv:1811.00410*, 2018.

[7] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv CoRR*, abs/1907.02893, 2019. URL http://arxiv.org/abs/1907.02893.

[8] Martin Arjovsky, Kamalika Chaudhuri, and David Lopez-Paz. Throwing away data improves worst-class error in imbalanced classification. *arXiv preprint arXiv:2205.11672*, 2022.

[9] Rim Assouel, Pau Rodriguez, Perouz Taslakian, David Vazquez, and Yoshua Bengio. Object-centric compositional imagination for visual abstract reasoning. In *ICLR2022 Workshop on the Elements of Reasoning: Objects, Structure and Causality*, 2022. URL https://openreview.net/forum?id=rCzfIruU5x5.

[10] Benjamin Aubin, Agnieszka Słowik, Martin Arjovsky, Leon Bottou, and David Lopez-Paz. Linear unit-tests for invariance discovery. *arXiv preprint arXiv:2102.10867*, 2021.

[11] Aharon Azulay and Yair Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? *Journal of Machine Learning Research*, 20 (184):1–25, 2019. URL http://jmlr.org/papers/v20/19-519.html.

[12] Bernard J Baars. *A cognitive theory of consciousness*. Cambridge University Press, 1993.

[13] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1409.0473.

[14] Dzmitry Bahdanau, Harm de Vries, Timothy J. O'Donnell, Shikhar Murty, Philippe Beaudoin, Yoshua Bengio, and Aaron Courville. Closure: Assessing systematic generalization of clevr models, 2019. URL https://arxiv.org/abs/1912.05783.

[15] Dzmitry Bahdanau, Shikhar Murty, Michael Noukhovitch, Thien Huu Nguyen, Harm de Vries, and Aaron C. Courville. Systematic generalization: What is required and can it be learned? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=HkezXnA9YX.

[16] Marco Baroni, Roberto Dessì, and Angeliki Lazaridou. Emergent language-based coordination in deep multi-agent systems. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, pages 11–16, 2022.

[17] Piersilvio De Bartolomeis, Antonio Orvieto, and Giambattista Parascandolo. Enhancing unit-tests for invariance discovery. In *ICML 2022: Workshop on Spurious Correlations, Invariance and Stability*, 2022. URL https://openreview.net/forum?id=-XVMGVmjNLs.

[18] Samyadeep Basu, Megan Stanley, John F Bronskill, Soheil Feizi, and Daniela Massiceti. Hard-meta-dataset++: Towards understanding few-shot performance on difficult tasks. In *The Eleventh International Conference on Learning Representations*, 2022.

[19] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinícius Flores Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Çaglar Gülçehre, H. Francis Song, Andrew J. Ballard, Justin Gilmer, George E. Dahl, Ashish Vaswani, Kelsey R. Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matthew Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks. *CoRR*, abs/1806.01261, 2018. URL http://arxiv.org/abs/1806.01261.

[20] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 472–489, Cham, 2018. Springer International Publishing. ISBN 978-3-030-01270-0.

[21] Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust Optimization*, volume 28 of *Princeton Series in Applied Mathematics*. Princeton University Press, 2009. ISBN 978-1-4008-3105-0. doi: 10.1515/9781400831050. URL https://doi.org/10.1515/9781400831050.

[22] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

[23] Yoshua Bengio, Yann Lecun, and Geoffrey Hinton. Deep learning for ai. *Commun. ACM*, 64(7):58–65, jun 2021. ISSN 0001-0782. doi: 10.1145/3448250. URL https://doi.org/10.1145/3448250.

[24] Leon Bergen, Timothy J. O'Donnell, and Dzmitry Bahdanau. Systematic generalization with edge transformers. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=UUds0Jr_XWk.

[25] Derek Bickerton. *More than nature needs*. Harvard University Press, 2014.

[26] Christopher M. Bishop. *Pattern recognition and machine learning, 5th Edition*. Information science and statistics. Springer, 2007. ISBN 9780387310732. URL https://www.worldcat.org/oclc/71008143.

[27] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[28] Léon Bottou. *On-Line Learning and Stochastic Approximations*, page 9–42. Cambridge University Press, USA, 1999. ISBN 0521652634.

[29] Diane Bouchacourt and Marco Baroni. How agents see things: On visual representations in an emergent language game. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 981–985, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1119. URL https://aclanthology.org/D18-1119.

[30] Diane Bouchacourt and Marco Baroni. Miss tools and mr fruit: Emergent communication in agents learning about object affordances. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3909–3918, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1380. URL https://aclanthology.org/P19-1380.

[31] Stephen P. Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2014.

[32] Henry Brighton and Simon Kirby. Understanding linguistic evolution by visualizing the emergence of topographic mappings. volume 12, page 229–242, Cambridge, MA, USA, March 2006. MIT Press. doi: 10.1162/106454606776073323. URL https://doi.org/10.1162/106454606776073323.

[33] Ted Briscoe. *Linguistic Evolution through Language Acquisition*. Cambridge University Press, 2002. doi: 10.1017/CBO9780511486524.

[34] John Bronskill, Daniela Massiceti, Massimiliano Patacchiola, Katja Hofmann, Sebastian Nowozin, and Richard Turner. Memory efficient meta-learning with large images. *Advances in Neural Information Processing Systems*, 34:24327–24339, 2021.

[35] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[36] Jonathon Byrd and Zachary Lipton. What is the effect of importance weighting in deep learning? In *International Conference on Machine Learning*, pages 872–881. PMLR, 2019.

[37] Jonathon Byrd and Zachary Lipton. What is the effect of importance weighting in deep learning? In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 872–881. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/byrd19a.html.

[38] Kris Cao, Angeliki Lazaridou, Marc Lanctot, Joel Z Leibo, Karl Tuyls, and Stephen Clark. Emergent communication through negotiation. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Hk6WhagRW.

[39] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *CoRR*, abs/2104.14294, 2021. URL https://arxiv.org/abs/2104.14294.

[40] Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*, 2023.

[41] Rahma Chaabouni, Eugene Kharitonov, Emmanuel Dupoux, and Marco Baroni. Anti-efficient encoding in emergent communication. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/31ca0ca71184bbdb3de7b20a51e88e90-Paper.pdf.

[42] Rahma Chaabouni, Eugene Kharitonov, Alessandro Lazaric, Emmanuel Dupoux, and Marco Baroni. Word-order biases in deep-agent emergent communication. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5166–5175, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1509. URL https://aclanthology.org/P19-1509.

[43] Rahma Chaabouni, Eugene Kharitonov, Diane Bouchacourt, Emmanuel Dupoux, and Marco Baroni. Compositionality and generalization in emergent languages. *CoRR*, abs/2004.09124, 2020. URL https://arxiv.org/abs/2004.09124.

[44] Rahma Chaabouni, Eugene Kharitonov, Diane Bouchacourt, Emmanuel Dupoux, and Marco Baroni. Compositionality and generalization in emergent languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4427–4442, Online, July 2020. Association for Computational

Linguistics. doi: 10.18653/v1/2020.acl-main.407. URL https://aclanthology.org/2020.acl-main.407.

[45] Rahma Chaabouni, Florian Strub, Florent Altché, Eugene Tarassov, Corentin Tallec, Elnaz Davoodi, Kory Wallace Mathewson, Olivier Tieleman, Angeliki Lazaridou, and Bilal Piot. Emergent communication at scale. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=AUGBfDIV9rL.

[46] Stephanie CY Chan, Adam Santoro, Andrew K Lampinen, Jane X Wang, Aaditya Singh, Pierre H Richemond, Jay McClelland, and Felix Hill. Data distributional properties drive emergent few-shot learning in transformers. *arXiv preprint arXiv:2205.05055*, 2022.

[47] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 6571–6583. Curran Associates, Inc., 2018. URL http://papers.nips.cc/paper/7892-neural-ordinary-differential-equations.pdf.

[48] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

[49] Yihang Chen, Grigorios Chrysos, and Volkan Cevher. A rate-distortion approach to domain generalization. 2021.

[50] Yining Chen, Elan Rosenfeld, Mark Sellke, Tengyu Ma, and Andrej Risteski. Iterative feature matching: Toward provable domain generalization with logarithmic environments. *arXiv preprint arXiv:2106.09913*, 2021.

[51] Yongqiang Chen, Kaiwen Zhou, Yatao Bian, Binghui Xie, Kaili Ma, Yonggang Zhang, Han Yang, Bo Han, and James Cheng. Pareto invariant risk minimization. *arXiv preprint arXiv:2206.07766*, 2022.

[52] Zhourong Chen, Yang Li, Samy Bengio, and Si Si. Gaternet: Dynamic filter selection in convolutional neural network via a dedicated global gating network. *arXiv preprint arXiv:1811.11205*, 2018.

[53] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. 2019.

[54] Noam Chomsky. *Aspects of the Theory of Syntax*, volume 1. MIT Press, 1965.

[55] Noam Chomsky. *Syntactic structures*. De Gruyter Mouton, 2009.

[56] Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big Data*, 5(2):153–163, 2017. doi: 10.1089/big.2016.0047. URL https://doi.org/10.1089/big.2016.0047.

[57] Michael Cogswell, Jiasen Lu, Stefan Lee, Devi Parikh, and Dhruv Batra. Emergence of compositional language with deep generational transmission. *CoRR*, abs/1904.09067, 2019. URL http://arxiv.org/abs/1904.09067.

[58] Tomasz Danel, Przemysław Spurek, Jacek Tabor, Marek Śmieja, Łukasz Struski, Agnieszka Słowik, and Łukasz Maziarka. Spatial graph convolutional networks. In *International Conference on Neural Information Processing*, pages 668–675. Springer, 2020.

[59] Jeffrey Dastin. Amazon scraps secret ai recruiting tool that showed bias against women. *Reuters*. URL https://www.theguardian.com/world/2017/mar/12/netherlands-will-pay-the-price-for-blocking-turkish-visit-erdogan.

[60] Amit Datta, Michael Carl Tschantz, and Anupam Datta. Automated experiments on ad privacy settings: A tale of opacity, choice, and discrimination. *CoRR*, abs/1408.6491, 2014. URL http://arxiv.org/abs/1408.6491.

[61] Harm de Vries, Dzmitry Bahdanau, Shikhar Murty, Aaron C. Courville, and Philippe Beaudoin. CLOSURE: assessing systematic generalization of CLEVR models. In *Visually Grounded Interaction and Language (ViGIL), NeurIPS 2019 Workshop, Vancouver, Canada, December 13, 2019*, 2019. URL https://vigilworkshop.github.io/static/papers/28.pdf.

[62] Grégoire Delétang, Anian Ruoss, Jordi Grau-Moya, Tim Genewein, Li Kevin Wenliang, Elliot Catt, Marcus Hutter, Shane Legg, and Pedro A Ortega. Neural networks and the chomsky hierarchy. *arXiv preprint arXiv:2207.02098*, 2022.

[63] Kevin Denamganaï and James Alfred Walker. On (emergent) systematic generalisation and compositionality in visual referential games with straight-through gumbel-softmax estimator. *arXiv preprint arXiv:2012.10776*, 2020.

[64] Bin Deng and Kui Jia. Counterfactual supervision-based information bottleneck for out-of-distribution generalization. *arXiv preprint arXiv:2208.07798*, 2022.

[65] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

[66] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[67] Remi Tachet des Combes, Mohammad Pezeshki, Samira Shabanian, Aaron C. Courville, and Yoshua Bengio. On the learning dynamics of deep neural networks. *CoRR*, abs/1809.06848, 2018. URL http://arxiv.org/abs/1809.06848.

[68] Roberto Dessì, Diane Bouchacourt, Davide Crepaldi, and Marco Baroni. Focus on what's informative and ignore what's not: Communication strategies in a referential game. *CoRR*, abs/1911.01892, 2019. URL http://arxiv.org/abs/1911.01892.

[69] Roberto Dessi, Eugene Kharitonov, and Marco Baroni. Interpretable agent communication from scratch (with a generic visual processor emerging on the side). In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=1AvtkM4H-y7.

[70] Roberto Dessì, Eleonora Gualdoni, Francesca Franzon, Gemma Boleda, and Marco Baroni. Communication breakdown: On the low mutual intelligibility between human and neural captioning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7998–8007, 2022.

[71] J.L.B.J.E.I. Donald A. Yates, J.L. Borges, D.A. Yates, J.E. Irby, A. Maurois, and S. Mangan. *Labyrinths: Selected Stories & Other Writings*. New Directions paperbook. New Directions Publishing Corporation, 1964. ISBN 9780811200127.

[72] Xin Du, Subramanian Ramamoorthy, Wouter Duivesteijn, Jin Tian, and Mykola Pechenizkiy. Beyond discriminant patterns: On the robustness of decision rule ensembles. *arXiv preprint arXiv:2109.10432*, 2021.

[73] Raman Dutt, Linus Ericsson, Pedro Sanchez, Sotirios A Tsaftaris, and Timothy Hospedales. Parameter-efficient fine-tuning for medical image analysis: The missed opportunity. *arXiv preprint arXiv:2305.08252*, 2023.

[74] Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. Exploring the landscape of spatial robustness. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages

1802–1811. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/engstrom19a.html.

[75] Katrina Evtimova, Andrew Drozdov, Douwe Kiela, and Kyunghyun Cho. Emergent communication in a multi-modal, multi-step referential game. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJGZq6g0-.

[76] Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. Language, Speech, and Communication. MIT Press, Cambridge, MA, 1998. ISBN 978-0-262-06197-1.

[77] Chelsea Finn. Principles for tackling distribution shift: Pessimism, adaptation, and anticipation at the deepmind ellis ucl csml seminar series., 2021. URL https://www.fields.utoronto.ca/talk-media/1/29/41/slides.pdf.

[78] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/finn17a.html.

[79] Jerry A. Fodor and Zenon W. Pylyshyn. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1):3–71, 1988. ISSN 0010-0277. doi: https://doi.org/10.1016/0010-0277(88)90031-5. URL https://www.sciencedirect.com/science/article/pii/0010027788900315.

[80] Ippei Fujisawa and Ryota Kanai. Addition for measuring the ability to understand regularity and to extrapolate. 2021.

[81] Shaoduo Gan, Akhil Mathur, Anton Isopoussu, Fahim Kawsar, Nadia Berthouze, and Nicholas D. Lane. Fruda: Framework for distributed adversarial domain adaptation. *IEEE Transactions on Parallel and Distributed Systems*, 33(11):3153–3164, 2022. doi: 10.1109/TPDS.2021.3136673.

[82] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. Domain-adversarial training of neural networks. In Gabriela Csurka, editor, *Domain Adaptation in Computer Vision Applications*, Advances in Computer Vision and Pattern Recognition, pages 189–209. Springer, 2017. doi: 10.1007/978-3-319-58347-1_10. URL https://doi.org/10.1007/978-3-319-58347-1_10.

[83] Shan Gao, Renmin Han, Xiangrui Zeng, Xuefeng Cui, Zhiyong Liu, Min Xu, and Fa Zhang. Dilated-densenet for macromolecule classification in cryo-electron tomography. In *Bioinformatics Research and Applications: 16th International Symposium, ISBRA 2020, Moscow, Russia, December 1–4, 2020, Proceedings*, page 82–94, Berlin, Heidelberg, 2020. Springer-Verlag. ISBN 978-3-030-57820-6. doi: 10.1007/978-3-030-57821-3_8. URL https://doi.org/10.1007/978-3-030-57821-3_8.

[84] Artur d'Avila Garcez and Luis C Lamb. Neurosymbolic ai: The 3 rd wave. *Artificial Intelligence Review*, pages 1–20, 2023.

[85] Robert Geirhos, Carlos R. Medina Temme, Jonas Rauber, Heiko H. Schütt, Matthias Bethge, and Felix A. Wichmann. Generalisation in humans and deep neural networks. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 7549–7561, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/0937fb5864ed06ffb59ae5f9b5ed67a9-Abstract.html.

[86] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard S. Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *CoRR*, abs/2004.07780, 2020. URL https://arxiv.org/abs/2004.07780.

[87] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, 2014.

[88] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6572.

[89] Geoffrey P Goodwin and PN Johnson-Laird. Reasoning about relations. *Psychological Review*, 112(2):468, 2005.

[90] Anirudh Goyal and Yoshua Bengio. Inductive biases for deep learning of higher-level cognition. *Proceedings of the Royal Society A*, 2022.

[91] Klaus Greff, Sjoerd van Steenkiste, and Jürgen Schmidhuber. On the binding problem in artificial neural networks. *CoRR*, abs/2012.05208, 2020. URL https://arxiv.org/abs/2012.05208.

[92] R. L. Gregory. *The intelligent eye*. Weidenfeld & Nicolson London, 1970. ISBN 0297000217.

[93] Romain Guillaume and Didier Dubois. A min-max regret approach to maximum likelihood inference under incomplete data. *International Journal of Approximate Reasoning*, 121:135–149, 2020. ISSN 0888-613X. doi: https://doi.org/10.1016/j.ijar.2020.03.003. URL https://www.sciencedirect.com/science/article/pii/S0888613X19301215.

[94] Shangmin Guo. Emergence of numeric concepts in multi-agent autonomous communication. *CoRR*, abs/1911.01098, 2019. URL http://arxiv.org/abs/1911.01098.

[95] Shangmin Guo, Yi Ren, Agnieszka Słowik, and Kory Mathewson. Inductive bias and language expressivity in emergent communication. *arXiv preprint arXiv:2012.02875*, 2020.

[96] Shangmin Guo, Yi Ren, Kory W. Mathewson, Simon Kirby, Stefano V. Albrecht, and Kenny Smith. Expressivity of emergent language is a trade-off between contextual complexity and unpredictability. *CoRR*, abs/2106.03982, 2021. URL https://arxiv.org/abs/2106.03982.

[97] Abhinav Gupta, Agnieszka Słowik, William L Hamilton, Mateja Jamnik, Sean B Holden, and Christopher Pal. Analyzing structural priors in multi-agent communication.

[98] Abhinav Gupta, Cinjon Resnick, Jakob Foerster, Andrew Dai, and Kyunghyun Cho. Compositionality and capacity in emergent languages. In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 34–38, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.repl4nlp-1.5. URL https://aclanthology.org/2020.repl4nlp-1.5.

[99] Ankit Gupta and Alexander M. Rush. Dilated convolutions for modeling long-distance genomic dependencies. 2017. doi: 10.48550/ARXIV.1710.01278. URL https://arxiv.org/abs/1710.01278.

[100] Alon Y. Halevy, Peter Norvig, and Fernando Pereira. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24(2):8–12, 2009. URL http://dblp.uni-trier.de/db/journals/expert/expert24.html#HalevyNP09.

[101] Ryuhei Hamaguchi, Aito Fujita, Keisuke Nemoto, Tomoyuki Imaizumi, and Shuhei Hikosaka. Effective use of dilated convolutions for segmenting small object instances in remote sensing imagery. 2017. doi: 10.48550/ARXIV.1709.00179. URL https://arxiv.org/abs/1709.00179.

[102] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *NIPS*, pages 1024–1034. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/6703-inductive-representation-learning-on-large-graphs.pdf.

[103] Laura Harding Graesser, Kyunghyun Cho, and Douwe Kiela. Emergent linguistic phenomena in multi-agent communication games. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3700–3710, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1384. URL https://aclanthology.org/D19-1384.

[104] Serhii Havrylov and Ivan Titov. Emergence of Language with Multi-agent Games: Learning to Communicate with Sequences of Symbols. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 2149–2159. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/6810-emergence-of-language-with-multi-agent-games-learning-to-communicate-with-sequences-of-symbols.pdf.

[105] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, abs/1502.01852, 2015. URL http://dblp.uni-trier.de/db/journals/corr/corr1502.html#HeZR015.

[106] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[107] Yue He, Zimu Wang, Peng Cui, Hao Zou, Yafeng Zhang, Qiang Cui, and Yong Jiang. CausPref: Causal preference learning for out-of-distribution recommendation. In *Proceedings of the ACM Web Conference 2022*. ACM, apr 2022. doi: 10.1145/3485447.3511969. URL https://doi.org/10.1145%2F3485447.3511969.

[108] Marti A. Hearst. Support vector machines. *IEEE Intelligent Systems*, 13(4):18–28, July 1998. ISSN 1541-1672. doi: 10.1109/5254.708428. URL https://doi.org/10.1109/5254.708428.

[109] James Heckman. Sample selection bias as a specification error (with an application to the estimation of labor supply functions). NBER Working Papers 0172, National

Bureau of Economic Research, Inc, 1977. URL https://EconPapers.repec.org/RePEc:nbr:nberwo:0172.

[110] Michael A. Hedderich and Dietrich Klakow. Training a neural network in a low-resource setting on automatically annotated noisy data. In *Proceedings of the Workshop on Deep Learning Approaches for Low-Resource NLP*, pages 12–18, Melbourne, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-3402. URL https://aclanthology.org/W18-3402.

[111] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=Hkg4TI9xl.

[112] Felix Hill, Andrew Lampinen, Rosalia Schneider, Stephen Clark, Matthew Botvinick, James L. McClelland, and Adam Santoro. Environmental drivers of systematicity and generalization in a situated agent. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SklGryBtwr.

[113] Geoffrey E. Hinton, James L. Mcclelland, and David E. Rumelhart. Distributed representations. In David E. Rumelhart and James L. Mcclelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations*, pages 77–109. MIT Press, Cambridge, MA, 1986.

[114] Sara Hooker. Moving beyond "algorithmic bias is a data problem". *Patterns*, 2(4): 100241, 2021. ISSN 2666-3899. doi: https://doi.org/10.1016/j.patter.2021.100241. URL https://www.sciencedirect.com/science/article/pii/S2666389921000611.

[115] Samuel Horváth, Stefanos Laskaridis, Mario Almeida, Ilias Leontiadis, Stylianos Venieris, and Nicholas Lane. Fjord: Fair and accurate federated learning under heterogeneous targets with ordered dropout. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 12876–12889. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/6aed000af86a084f9cb0264161e29dd3-Paper.pdf.

[116] D.G. Horvitz and DJ Thompson. A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association*, 47(260): 663–685, 1952.

[117] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei. Relation networks for object detection. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3588–3597, Los Alamitos, CA, USA, jun 2018. IEEE Computer Society. doi: 10.1109/CVPR.2018.00378. URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2018.00378.

[118] Weihua Hu, Gang Niu, Issei Sato, and Masashi Sugiyama. Does distributionally robust supervised learning give robust classifiers? In *International Conference on Machine Learning*, pages 2029–2037. PMLR, 2018.

[119] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2261–2269. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.243. URL https://doi.org/10.1109/CVPR.2017.243.

[120] Drew A. Hudson and Christopher D. Manning. Compositional attention networks for machine reasoning. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=S1Euwz-Rb.

[121] Freiherr von Humboldt, Wilhelm. *Über die Verschiedenheit des menschlichen Sprachbaues und ihren Einfluss auf die geistige Entwicklung des Menschengeschlechts*. Druckerei der Königlichen Akademie der Wissenschaften, Berlin, 1836.

[122] Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. Compositionality decomposed: how do neural networks generalise? 2019. doi: 10.48550/ARXIV.1908.08351. URL https://arxiv.org/abs/1908.08351.

[123] Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. The compositionality of neural networks: integrating symbolism and connectionism. *arXiv:1908.08351 [cs, stat]*, August 2019. URL http://arxiv.org/abs/1908.08351. arXiv: 1908.08351.

[124] Badr Youbi Idrissi, Martin Arjovsky, Mohammad Pezeshki, and David Lopez-Paz. Simple data balancing achieves competitive worst-group-accuracy. In Bernhard Schölkopf, Caroline Uhler, and Kun Zhang, editors, *Proceedings of the First Conference on Causal Learning and Reasoning*, volume 177 of *Proceedings of Machine Learning Research*, pages 336–351. PMLR, 11–13 Apr 2022. URL https://proceedings.mlr.press/v177/idrissi22a.html.

[125] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 448–456. JMLR.org, 2015.

[126] Eric Jang, Shixiang Gu, and Ben Poole. Categorical Reparameterization with Gumbel-Softmax. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=rkE3y85ee.

[127] T.M.V. Janssen and B. Partee. Handbook of logic and language. chapter 7. Elsevier Science Publishers B. V., NLD, 2nd edition, 2010. ISBN 9780444537263.

[128] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2901–2910, 2017.

[129] Justin M. Johnson and Taghi M. Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):27, March 2019. ISSN 2196-1115. doi: 10.1186/s40537-019-0192-5. URL https://doi.org/10.1186/s40537-019-0192-5.

[130] Emilio Jorge, Mikael Kågebäck, and Emil Gustavsson. Learning to play guess who? and inventing a grounded language as a consequence. *CoRR*, abs/1611.03218, 2016. URL http://arxiv.org/abs/1611.03218.

[131] Rishabh Kabra, Daniel Zoran, Goker Erdogan, Loic Matthey, Antonia Creswell, Matthew Botvinick, Alexander Lerchner, and Christopher P. Burgess. SIMONe: View-invariant, temporally-abstracted object representations via unsupervised video decomposition. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=YSzTMntO1KY.

[132] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5132–5143. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/karimireddy20a.html.

[133] Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak,

Tibor Tihon, Dmitry Tsarkov, Xiao Wang, Marc van Zee, and Olivier Bousquet. Measuring compositional generalization: A comprehensive method on realistic data. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SygcCnNKwr.

[134] Eugene Kharitonov and Marco Baroni. Emergent language generalization and acquisition speed are not tied to compositionality. In Afra Alishahi, Yonatan Belinkov, Grzegorz Chrupala, Dieuwke Hupkes, Yuval Pinter, and Hassan Sajjad, editors, *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2020, Online, November 2020*, pages 11–15. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.blackboxnlp-1.2. URL https://doi.org/10.18653/v1/2020.blackboxnlp-1.2.

[135] Eugene Kharitonov, Rahma Chaabouni, Diane Bouchacourt, and Marco Baroni. Entropy minimization in emergent languages. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5220–5230. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/kharitonov20a.html.

[136] Kia Khezeli, Arno Blaas, Frank Soboczenski, Nicholas Chia, and John Kalantari. On invariance penalties for risk minimization. *arXiv preprint arXiv:2106.09777*, 2021.

[137] Junkyung Kim, Matthew Ricci, and Thomas Serre. Not-so-clevr: learning same–different relations strains feedforward neural networks. *Interface Focus*, 8(4):20180011, 2018. doi: 10.1098/rsfs.2018.0011. URL https://royalsocietypublishing.org/doi/abs/10.1098/rsfs.2018.0011.

[138] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

[139] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun, editors, *ICLR*, 2014.

[140] Thomas Kipf, Gamaleldin Fathy Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff. Conditional object-centric learning from video. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=aD7uesX1GF_.

[141] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=SJU4ayYgl.

[142] Satwik Kottur, José Moura, Stefan Lee, and Dhruv Batra. Natural language does not emerge 'naturally' in multi-agent dialog. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2962–2967, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1321. URL https://www.aclweb.org/anthology/D17-1321.

[143] Satwik Kottur, José M. F. Moura, Stefan Lee, and Dhruv Batra. Natural language does not emerge 'naturally' in multi-agent dialog. *CoRR*, abs/1706.08502, 2017. URL http://arxiv.org/abs/1706.08502.

[144] Masanori Koyama and Shoichiro Yamaguchi. Out-of-distribution generalization with maximal invariant predictor. *arXiv preprint arXiv:2008.01883*, 2020.

[145] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *Int. J. Comput. Vision*, 123(1):32–73, may 2017. ISSN 0920-5691. doi: 10.1007/s11263-016-0981-7. URL https://doi.org/10.1007/s11263-016-0981-7.

[146] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 1106–1114, 2012. URL https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html.

[147] David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex). In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5815–5826. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/krueger21a.html.

[148] Miroslav Kubat, Robert C. Holte, and Stan Matwin. Machine learning for the detection of oil spills in satellite radar images. *Mach. Learn.*, 30(2–3):195–215, feb 1998. ISSN 0885-6125. doi: 10.1023/A:1007452223027. URL https://doi.org/10.1023/A:1007452223027.

[149] Daniel Kuhn, Peyman Mohajerin Esfahani, Viet Anh Nguyen, and Soroosh Shafieezadeh-Abadeh. Wasserstein distributionally robust optimization: Theory and applications in machine learning. In *Operations research & management science in the age of analytics*, pages 130–166. Informs, 2019.

[150] Ankit Kumar, Piyush Makhija, and Anuj Gupta. Noisy Text Data: Achilles' Heel of BERT. *arXiv e-prints*, art. arXiv:2003.12932, March 2020.

[151] Alexandre Lacoste, Pau Rodríguez, Frédéric Branchaud-Charron, Parmida Atighehchian, Massimo Caccia, Issam Laradji, Alexandre Drouin, Matt Craddock, Laurent Charlin, and David Vázquez. Synbols: Probing learning algorithms with synthetic datasets. *arXiv preprint arXiv:2009.06415*, 2020.

[152] Brenden Lake and Marco Baroni. Still not systematic after all these years: On the compositional skills of sequence-to-sequence recurrent networks, 2018. URL https://openreview.net/forum?id=H18WqugAb.

[153] Brenden Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2873–2882. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/lake18a.html.

[154] Brenden M. Lake. Compositional generalization through meta sequence-to-sequence learning. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 9788–9798, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/f4d0e2e7fc057a58f7ca4a391f01940a-Abstract.html.

[155] Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015. doi: 10.1126/science.aab3050. URL https://www.science.org/doi/abs/10.1126/science.aab3050.

[156] Brenden M Lake, Tomer D Ullman, Joshua B. Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40:e253, 2017 Jan 2017. ISSN 1469-1825. doi: https://doi.org/10.1017/S0140525X16001837. URL https://www.cambridge.org/core/journals/behavioral-and-brain-sciences/article/building-machines-that-learn-and-think-like-people/A9535B1D745A0377E16C590E14B94993/core-reader.

[157] Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40:e253, 2017. doi: 10.1017/S0140525X16001837.

[158] Kevin N. Laland. The origins of language in teaching. *Psychonomic Bulletin & Review*, 24:225 – 231, 2017.

[159] Alex Lamb, Anirudh Goyal, Agnieszka Słowik, Michael Mozer, Philippe Beaudoin, and Yoshua Bengio. Neural function modules with sparse arguments: A dynamic approach to integrating information across layers, 2020.

[160] Alex Lamb, Anirudh Goyal, Agnieszka Słowik, Michael Mozer, Philippe Beaudoin, and Yoshua Bengio. Neural function modules with sparse arguments: A dynamic approach to integrating information across layers. In *International Conference on Artificial Intelligence and Statistics*, pages 919–927. PMLR, 2021.

[161] Angeliki Lazaridou and Marco Baroni. Emergent multi-agent communication in the deep learning era. *arXiv preprint arXiv:2006.02419*, 2020.

[162] Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. Multi-agent cooperation and the emergence of (natural) language. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=Hk8N3Sclg.

[163] Angeliki Lazaridou, Karl Moritz Hermann, Karl Tuyls, and Stephen Clark. Emergence of linguistic communication from referential games with symbolic and pixel input. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=HJGv1Z-AW.

[164] Angeliki Lazaridou, Adhi Kuncoro, Elena Gribovskaya, Devang Agrawal, Adam Liska, Tayfun Terzi, Mai Gimenez, Cyprien de Masson d'Autume, Tomas Kocisky, Sebastian Ruder, Dani Yogatama, Kris Cao, Susannah Young, and Phil Blunsom. Mind the gap: Assessing temporal generalization in neural language models. In

M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 29348–29363. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/f5bf0ba0a17ef18f9607774722f5698c-Paper.pdf.

[165] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Comput.*, 1(4):541–551, December 1989. ISSN 0899-7667. doi: 10.1162/neco.1989.1.4.541. URL http://dx.doi.org/10.1162/neco.1989.1.4.541.

[166] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. volume 86, pages 2278–2324. IEEE, 1998. doi: 10.1109/5.726791.

[167] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

[168] Yann LeCun, Patrick Haffner, Léon Bottou, and Yoshua Bengio. Object recognition with gradient-based learning. In *Shape, Contour and Grouping in Computer Vision*, pages 319–, London, UK, UK, 1999. Springer-Verlag. ISBN 3-540-66722-9. URL http://dl.acm.org/citation.cfm?id=646469.691875.

[169] David Lewis. *Convention: A philosophical study.* Harvard University Press, 1969.

[170] Yuanpeng Li, Liang Zhao, Jianyu Wang, and Joel Hestness. Compositional generalization for primitive substitutions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4293–4302, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1438. URL https://aclanthology.org/D19-1438.

[171] Yong Lin, Qing Lian, and Tong Zhang. An empirical study of invariant risk minimization on deep models. In *ICML 2021 Workshop on Uncertainty and Robustness in Deep Learning*, volume 1, page 7, 2021.

[172] Zachary Lipton, Yu-Xiang Wang, and Alexander Smola. Detecting and correcting for label shift with black box predictors. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3122–3130. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/lipton18a.html.

[173] Chenyao Liu, Shengnan An, Zeqi Lin, Qian Liu, Bei Chen, Jian-Guang Lou, Lijie Wen, Nanning Zheng, and Dongmei Zhang. Learning algebraic recombination for

compositional generalization. *CoRR*, abs/2107.06516, 2021. URL https://arxiv.org/abs/2107.06516.

[174] Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 6781–6792. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/liu21f.html.

[175] Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/60106888f8977b71e1f15db7bc9a88d1-Paper.pdf.

[176] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3730–3738, 2015.

[177] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *NeurIPS 2020*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/8511df98c02ab60aea1b2356c013bc0f-Abstract.html.

[178] David Lopez-Paz, Diane Bouchacourt, Levent Sagun, and Nicolas Usunier. Measuring and signing fairness as performance under multiple stakeholder distributions. *arXiv preprint arXiv:2207.09960*, 2022.

[179] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

[180] Alexandra Sasha Luccioni, Christopher Akiki, Margaret Mitchell, and Yacine Jernite. Stable bias: Analyzing societal representations in diffusion models, 2023.

[181] Gary Lupyan and Rick Dale. Language structure is partly determined by social structure. *PLOS ONE*, 5(1):1–10, 01 2010. doi: 10.1371/journal.pone.0008559. URL https://doi.org/10.1371/journal.pone.0008559.

[182] Spandan Madan, Timothy Henry, Jamell Dozier, Helen Ho, Nishchal Bhandari, Tomotake Sasaki, Frédo Durand, Hanspeter Pfister, and Xavier Boix. When and how convolutional neural networks generalize to out-of-distribution category–viewpoint combinations. *Nature Machine Intelligence*, 4(2):146–153, Feb 2022. ISSN 2522-5839. doi: 10.1038/s42256-021-00437-5. URL https://doi.org/10.1038/s42256-021-00437-5.

[183] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=S1jE5L5gl.

[184] David Madras and Richard Zemel. Identifying and benchmarking natural out-of-context prediction problems. *Advances in Neural Information Processing Systems*, 34:15344–15358, 2021.

[185] Mateusz Malinowski. Towards holistic machines : From visual recognition to question answering about real-world images, 2017.

[186] Masoud Mansoury, Himan Abdollahpouri, Mykola Pechenizkiy, Bamshad Mobasher, and Robin Burke. Feedback loop and bias amplification in recommender systems. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, page 2145–2148, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450368599. doi: 10.1145/3340531.3412152. URL https://doi.org/10.1145/3340531.3412152.

[187] Gary Marcus. Deep learning: A critical appraisal. *CoRR*, abs/1801.00631, 2018. URL http://arxiv.org/abs/1801.00631.

[188] Daniela Massiceti, Luisa Zintgraf, John Bronskill, Lida Theodorou, Matthew Tobias Harris, Edward Cutrell, Cecily Morrison, Katja Hofmann, and Simone Stumpf. Orbit: A real-world few-shot dataset for teachable object recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10818–10828, 2021.

[189] R. Thomas McCoy, Junghyun Min, and Tal Linzen. BERTs of a feather do not generalize together: Large variability in generalization across models with similar test set performance. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 217–227, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.blackboxnlp-1.21. URL https://www.aclweb.org/anthology/2020.blackboxnlp-1.21.

[190] Jil McIntosh. Watch: Did this tesla's sensors just mistake the moon for a traffic light? *Driving*. URL https://driving.ca/auto-news/news/watch-did-this-teslas-sensors-just-mistake-the-moon-for-a-traffic-light.

[191] John McWhorter. What happened to English? *Diachronica*, 19(2):217–272, 2002. ISSN 0176-4225. doi: https://doi.org/10.1075/dia.19.2.02wha. URL https://www.jbe-platform.com/content/journals/10.1075/dia.19.2.02wha.

[192] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *CoRR*, abs/1908.09635, 2019. URL http://arxiv.org/abs/1908.09635.

[193] Nicolai Meinshausen, Peter Bühlmann, et al. Maximin effects in inhomogeneous large-scale data. *The Annals of Statistics*, 43(4):1801–1830, 2015.

[194] Cade Metz and Adam Satariano. An algorithm that grants freedom, or takes it away. *The New York Times*, 2020. ISSN 0362-4331.

[195] Grégoire Mialon, Roberto Dessì, Maria Lomeli, Christoforos Nalmpantis, Ram Pasunuru, Roberta Raileanu, Baptiste Rozière, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, et al. Augmented language models: a survey. *arXiv preprint arXiv:2302.07842*, 2023.

[196] Marvin Minsky. Steps toward artificial intelligence. *Proceedings of the IRE*, 49(1): 8–30, 1961. doi: 10.1109/JRPROC.1961.287775.

[197] Marvin Minsky. *The Society of Mind*. Simon & Schuster, Inc., USA, 1986. ISBN 0671607405.

[198] Marvin Minsky and Seymour A. Papert. *Perceptrons: An Introduction to Computational Geometry*. The MIT Press, 09 2017. ISBN 9780262343930. doi: 10.7551/mitpress/11301.001.0001. URL https://doi.org/10.7551/mitpress/11301.001.0001.

[199] Melanie Mitchell. How do we know how smart ai systems are? *Science*, 381(6654): adj5957, 2023. doi: 10.1126/science.adj5957. URL https://www.science.org/doi/abs/10.1126/science.adj5957.

[200] Mazda Moayeri, Kiarash Banihashem, and Soheil Feizi. Explicit tradeoffs between adversarial and natural distributional robustness. *arXiv preprint arXiv:2209.07592*, 2022.

[201] Milton Llera Montero, Casimir JH Ludwig, Rui Ponte Costa, Gaurav Malhotra, and Jeffrey Bowers. The role of disentanglement in generalisation. In *International*

Conference on Learning Representations, 2021. URL https://openreview.net/forum?id=qbH974jKUVy.

[202] Pablo Morales-Alvarez, Wenbo Gong, Angus Lamb, Simon Woodhead, Simon Peyton Jones, Nick Pawlowski, Miltiadis Allamanis, and Cheng Zhang. Simultaneous missing value imputation and structure learning with groups. *Advances in Neural Information Processing Systems*, 35:20011–20024, 2022.

[203] Jesse Mu and Noah Goodman. Emergent communication of generalizations. *Advances in Neural Information Processing Systems*, 34, 2021.

[204] Vaishnavh Nagarajan, Anders Andreassen, and Behnam Neyshabur. Understanding the failure modes of out-of-distribution generalization. In *International Conference on Learning Representations*, 2020.

[205] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124003, 2021.

[206] David Navon. Forest before trees: The precedence of global features in visual perception. *Cognitive Psychology*, 9(3):353–383, 1977. ISSN 0010-0285. doi: https://doi.org/10.1016/0010-0285(77)90012-3. URL https://www.sciencedirect.com/science/article/pii/0010028577900123.

[207] Thuan Nguyen, Boyang Lyu, Prakash Ishwar, Matthias Scheutz, and Shuchin Aeron. Conditional entropy minimization principle for learning domain invariant representation features. *arXiv preprint arXiv:2201.10460*, 2022.

[208] Inbar Oren, Jonathan Herzig, Nitish Gupta, Matt Gardner, and Jonathan Berant. Improving compositional generalization in semantic parsing. In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 2482–2495. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.findings-emnlp.225. URL https://doi.org/10.18653/v1/2020.findings-emnlp.225.

[209] Aditya Paliwal, Sarah M. Loos, Markus N. Rabe, Kshitij Bansal, and Christian Szegedy. Graph representations for higher-order logic and theorem proving. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 2967–2974. AAAI Press, 2020. URL https://aaai.org/ojs/index.php/AAAI/article/view/5689.

[210] Giambattista Parascandolo, Alexander Neitz, Antonio Orvieto, Luigi Gresele, and Bernhard Schölkopf. Learning explanations that are hard to vary. *arXiv preprint arXiv:2009.00329*, 2020.

[211] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[212] Francis Jeffry Pelletier. The principle of semantic compositionality. *Topoi*, 13(1): 11–24, 1994. doi: 10.1007/bf00763644.

[213] Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference using invariant prediction: identification and confidence intervals. *arXiv preprint arXiv:1501.01332*, 2015.

[214] Mohammad Pezeshki, Sékou-Oumar Kaba, Yoshua Bengio, Aaron C. Courville, Doina Precup, and Guillaume Lajoie. Gradient starvation: A learning proclivity in neural networks. *CoRR*, abs/2011.09468, 2020. URL https://arxiv.org/abs/2011.09468.

[215] Steven T. Piantadosi and Richard N. Aslin. Compositional reasoning in early childhood. *PLoS ONE*, 11, 2016. doi: https://doi.org/10.1371/journal.pone.0147734.

[216] Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*, 2022.

[217] Doina Precup, Richard S. Sutton, and Satinder P. Singh. Eligibility traces for off-policy policy evaluation. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, page 759–766, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. ISBN 1558607072.

[218] Foster Provost. Machine learning from imbalanced data sets 101. In *Proceedings of the AAAI'2000 workshop on imbalanced data sets*, volume 68, pages 1–3. AAAI Press, 2000.

[219] Zhaozhi Qian, Ahmed M. Alaa, and Mihaela van der Schaar. When and how to lift the lockdown? global covid-19 scenario analysis and policy assessment using compartmental gaussian processes. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 10729–10740. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/79a3308b13cd31f096d8a4a34f96b66b-Paper.pdf.

[220] Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence. *Dataset Shift in Machine Learning*. The MIT Press, 2009. ISBN 0262170051.

[221] Md Rafiqul Islam Rabin, Aftab Hussain, Mohammad Amin Alipour, and Vincent J. Hellendoorn. Memorization and generalization in neural code intelligence models. *Information and Software Technology*, 153:107066, 2023. ISSN 0950-5849. doi: https://doi.org/10.1016/j.infsof.2022.107066. URL https://www.sciencedirect.com/science/article/pii/S0950584922001756.

[222] Hamed Rahimian and Sanjay Mehrotra. Distributionally robust optimization: A review. *arXiv preprint arXiv:1908.05659*, 2019.

[223] Aida Rahmattalabi, Phebe Vayanos, Anthony Fulginiti, Eric Rice, Bryan Wilder, Amulya Yadav, and Milind Tambe. Exploring algorithmic fairness in robust graph covering problems. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 15750–15761, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/1d7c2aae840867027b7edd17b6aaa0e9-Abstract.html.

[224] Nathanaël Carraz Rakotonirina, Roberto Dessì, Fabio Petroni, Sebastian Riedel, and Marco Baroni. Can discrete information extraction prompts generalize across language models? *arXiv preprint arXiv:2302.09865*, 2023.

[225] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *CoRR*, abs/2102.12092, 2021. URL https://arxiv.org/abs/2102.12092.

[226] Rajat Rasal, Daniel C Castro, Nick Pawlowski, and Ben Glocker. Deep structural causal shape models. *arXiv preprint arXiv:2208.10950*, 2022.

[227] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/5f8e2fa1718d1bbcadf1cd9c7a54fb8c-Paper.pdf.

[228] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet classifiers generalize to ImageNet? In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5389–5400. PMLR, 09–15 Jun 2019. URL http://proceedings.mlr.press/v97/recht19a.html.

[229] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6517–6525, 2017.

[230] Yi Ren, Shangmin Guo, Matthieu Labeau, Shay B. Cohen, and Simon Kirby. Compositional languages emerge in a neural iterated learning model. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HkePNpVKPB.

[231] Matthew Ricci, Junkyung Kim, and Thomas Serre. Same-different problems strain convolutional neural networks. In Chuck Kalish, Martina A. Rau, Xiaojin (Jerry) Zhu, and Timothy T. Rogers, editors, *Proceedings of the 40th Annual Meeting of the Cognitive Science Society, CogSci 2018, Madison, WI, USA, July 25-28, 2018*. cognitivesciencesociety.org, 2018. URL https://mindmodeling.org/cogsci2018/papers/0188/index.html.

[232] Mathieu Rita, Florian Strub, Jean-Bastien Grill, Olivier Pietquin, and Emmanuel Dupoux. On the role of population heterogeneity in emergent communication. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=5Qkd7-bZfI.

[233] Kevin Roose. A conversation with bing's chatbot left me deeply unsettled. *The New York Times*. URL https://www.nytimes.com/2023/02/16/technology/bing-chatbot-microsoft-chatgpt.html.

[234] F. Rosenblatt. The perceptron – a perceiving and recognizing automaton. Technical Report 85-460-1, Cornell Aeronautical Laboratory, Ithaca, New York, January 1957.

[235] Laura Ruis, Jacob Andreas, Marco Baroni, Diane Bouchacourt, and Brenden M Lake. A benchmark for systematic generalization in grounded language understanding. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 19861–19872. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/e5a90182cc81e12ab5e72d66e0b46fe3-Paper.pdf.

[236] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. *Learning Internal Representations by Error Propagation*, page 318–362. MIT Press, Cambridge, MA, USA, 1986. ISBN 026268053X.

[237] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning Representations by Back-propagating Errors. *Nature*, 323(6088):533–536, 1986. doi: 10.1038/323533a0. URL http://www.nature.com/articles/323533a0.

[238] Jake Russin, Jason Jo, Randall C. O'Reilly, and Yoshua Bengio. Compositional generalization in a deep seq2seq model by separating syntax and semantics. *CoRR*, abs/1904.09708, 2019. URL http://arxiv.org/abs/1904.09708.

[239] Devendra Singh Sachan, Yuhao Zhang, Peng Qi, and William L. Hamilton. Do syntax trees help pre-trained transformers extract information? In Paola Merlo, Jörg Tiedemann, and Reut Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 2647–2661. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.eacl-main.228. URL https://doi.org/10.18653/v1/2021.eacl-main.228.

[240] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations (ICLR)*, 2020.

[241] Adam Santoro, David Raposo, David G. T. Barrett, Mateusz Malinowski, Razvan Pascanu, Peter W. Battaglia, and Timothy P. Lillicrap. A simple neural network module for relational reasoning. In *NIPS*, 2017.

[242] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761*, 2023.

[243] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2

014.09.003. URL https://www.sciencedirect.com/science/article/pii/S0893608014002135.

[244] Eric Schulz, Josh Tenenbaum, David K Duvenaud, Maarten Speekenbrink, and Samuel J Gershman. Probing the compositionality of intuitive functions. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/file/49ad23d1ec9fa4bd8d77d02681df5cfa-Paper.pdf.

[245] Dana Scott. Outline of a mathematical theory of computation. Technical Report PRG02, OUCL, November 1970.

[246] Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9573–9585. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/6cfe0e6127fa25df2a0ef2ae1067d915-Paper.pdf.

[247] Uri Shalit, Fredrik D. Johansson, and David Sontag. Estimating individual treatment effect: Generalization bounds and algorithms. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 3076–3085. JMLR.org, 2017.

[248] Zheyan Shen, Jiashuo Liu, Yue He, Xingxuan Zhang, Renzhe Xu, Han Yu, and Peng Cui. Towards out-of-distribution generalization: A survey. *arXiv preprint arXiv:2108.13624*, 2021.

[249] Shoaib Ahmed Siddiqui, David Krueger, and Thomas Breuel. Investigating the nature of 3d generalization in deep neural networks. *arXiv preprint arXiv:2304.09358*, 2023.

[250] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Vedavyas Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy P. Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nat.*, 529(7587):484–489, 2016. doi: 10.1038/nature16961. URL https://doi.org/10.1038/nature16961.

[251] Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L. Hamilton. CLUTRR: A diagnostic benchmark for inductive reasoning from text. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 4505–4514. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1458. URL https://doi.org/10.18653/v1/D19-1458.

[252] Agnieszka Słowik and Léon Bottou. Algorithmic bias and data bias: Understanding the relation between distributionally robust optimization and data curation. *arXiv preprint arXiv:2106.09467*, 2021.

[253] Agnieszka Słowik and Léon Bottou. On distributionally robust optimization and data rebalancing. In *International Conference on Artificial Intelligence and Statistics*, pages 1283–1297. PMLR, 2022.

[254] Agnieszka Słowik, Abhinav Gupta, William L. Hamilton, Mateja Jamnik, and Sean B. Holden. Towards graph representation learning in emergent communication. In *AAAI-20 Workshop on Reinforcement Learning in Games*, 2020. URL http://aaai-rlg.mlanctot.info/papers/AAAI20-RLG_paper_27.pdf. arXiv:2001.09063v2.

[255] Agnieszka Słowik, Abhinav Gupta, William L Hamilton, Mateja Jamnik, and Sean B Holden. Towards graph representation learning in emergent communication. *arXiv preprint arXiv:2001.09063*, 2020.

[256] Agnieszka Słowik, Abhinav Gupta, William L Hamilton, Mateja Jamnik, Sean B Holden, and Christopher Pal. Structural inductive biases in emergent communication. *arXiv preprint arXiv:2002.01335*, 2020.

[257] Agnieszka S{\l}owik, Chaitanya Mangla, Mateja Jamnik, Sean Holden, and Lawrence Paulson. Bayesian optimisation for heuristic configuration in automated theorem proving. In Laura Kovacs and Andrei Voronkov, editors, *Vampire 2018 and Vampire 2019. The 5th and 6th Vampire Workshops*, volume 71 of *EPiC Series in Computing*, pages 45–51. EasyChair, 2020. doi: 10.29007/q91g. URL https://easychair.org/publications/paper/K7Zd.

[258] Agnieszka Słowik, Chaitanya Mangla, Mateja Jamnik, Sean B Holden, and Lawrence C Paulson. Bayesian optimisation for premise selection in automated theorem proving (student abstract). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13919–13920, 2020.

[259] Agnieszka Słowik, Abhinav Gupta, William L. Hamilton, Mateja Jamnik, Sean B. Holden, and Christopher Pal. Exploring structural inductive biases in emergent communication. In *Proceedings of the 43rd Annual Meeting of the Cognitive Science Society*, volume 43, page 3156, July 2021. URL https://escholarship.org/uc/item/09b496t7. arXiv:2002.01335v4.

[260] Agnieszka Słowik, Léon Bottou, Sean B Holden, and Mateja Jamnik. On the relation between distributionally robust optimization and data curation (student abstract). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 13053–13054, 2022.

[261] Craig S. Smith. Hallucinations could blunt chatgpt's success. *IEEE Spectrum*. URL https://spectrum.ieee.org/ai-hallucination.

[262] Kenny Smith, Henry Brighton, and Simon Kirby. Complex Systems In Language Evolution: The Cultural Emergence Of Compositional Structure. *Advances in Complex Systems (ACS)*, 6(04):537–558, 2003. doi: 10.1142/S0219525903001055. URL https://ideas.repec.org/a/wsi/acsxxx/v06y2003i04ns0219525903001055.html.

[263] Kenny Smith, Simon Kirby, and Henry Brighton. Iterated learning: A framework for the emergence of language. *Artif. Life*, 9(4):371–386, sep 2003. ISSN 1064-5462. doi: 10.1162/106454603322694825. URL https://doi.org/10.1162/106454603322694825.

[264] Matthew Spike, Kevin Stadler, Simon Kirby, and Kenny Smith. Minimal requirements for the emergence of learned signaling. *Cognitive Science*, March 2016. ISSN 0364-0213. doi: 10.1111/cogs.12351. Copyright © 2016 The Authors. Cognitive Science published by Wiley Periodicals, Inc. on behalf of Cognitive Science Society.

[265] Przemysław Spurek, Tomasz Danel, Jacek Tabor, Marek Smieja, Lukasz Struski, Agnieszka Slowik, and Lukasz Maziarka. Geometric graph convolutional neural networks. *arXiv preprint arXiv:1909.05310*, 2019.

[266] Sanjana Srivastava, Guy Ben-Yosef, and Xavier Boix. Minimal images in deep neural networks: Fragile object recognition in natural images. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=S1xNb2A9YX.

[267] Matthew Staib and Stefanie Jegelka. Distributionally robust optimization and generalization in kernel methods. *CoRR*, abs/1905.10943, 2019. URL http://arxiv.org/abs/1905.10943.

[268] Emma Strubell and Andrew McCallum. Dependency parsing with dilated iterated graph cnns. 2017. doi: 10.48550/ARXIV.1705.00403. URL https://arxiv.org/abs/1705.00403.

[269] David Stutz, Matthias Hein, and Bernt Schiele. Disentangling adversarial robustness and generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6976–6987, 2019.

[270] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper/2014/file/a14ac55a4f27472c5d894ec1c3c743d2-Paper.pdf.

[271] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[272] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6199.

[273] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR)*, 2015. URL http://arxiv.org/abs/1409.4842.

[274] Naoya Takahashi and Yuki Mitsufuji. D3net: Densely connected multidilated densenet for music source separation. *CoRR*, abs/2010.01733, 2020. URL https://arxiv.org/abs/2010.01733.

[275] Henning Tiedemann, Yaniv Morgenstern, Filipp Schmidt, and Roland W Fleming. One-shot generalization in humans revealed through a drawing task. *eLife*, 11: e75485, may 2022. ISSN 2050-084X. doi: 10.7554/eLife.75485. URL https://doi.org/10.7554/eLife.75485.

[276] Antonio Torralba, Robert Fergus, and William T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(11):1958–1970, 2008. doi: 10.1109/TPAMI.2008.128. URL https://doi.org/10.1109/TPAMI.2008.128.

[277] Joe Townsend, Esma Mansouri-Benssassi, Kwun Ho Ngan, and Artur d'Avila Garcez. Discovering visual concepts and rules in convolutional neural networks. *Compendium of Neurosymbolic Artificial Intelligence*, pages 337–372, 2023.

[278] V. Vapnik. Principles of risk minimization for learning theory. In *Proceedings of the 4th International Conference on Neural Information Processing Systems*, NIPS'91, page 831–838, San Francisco, CA, USA, 1991. Morgan Kaufmann Publishers Inc. ISBN 1558602224.

[279] Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.

[280] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[281] Ramakrishna Vedantam, Arthur Szlam, Maximilian Nickel, Ari Morcos, and Brenden M. Lake. CURI: A benchmark for productive concept learning under uncertainty. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 10519–10529. PMLR, 2021. URL http://proceedings.mlr.press/v139/vedantam21a.html.

[282] Robin Vogel, Mastane Achab, Stéphan Clémençon, and Charles Tillier. Weighted empririical risk minimization: Transfer learning based on importance sampling. In *28th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2020, Bruges, Belgium, October 2-4, 2020*, pages 515–520, 2020. URL https://www.esann.org/sites/default/files/proceedings/2020/ES2020-120.pdf.

[283] Haoxiang Wang, Haozhe Si, Bo Li, and Han Zhao. Provable domain generalization via invariant-feature subspace recovery. *arXiv preprint arXiv:2201.12919*, 2022.

[284] Jianfeng Wang, Thomas Lukasiewicz, Daniela Massiceti, Xiaolin Hu, Vladimir Pavlovic, and Alexandros Neophytou. Np-match: When neural processes meet semi-supervised learning. In *International Conference on Machine Learning*, pages 22919–22934. PMLR, 2022.

[285] Mingzhe Wang, Yihe Tang, Jian Wang, and Jia Deng. Premise selection for theorem proving by deep graph embedding. In *Proceedings of the 31st International Conference*

on Neural Information Processing Systems, NIPS'17, page 2783–2793, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

[286] Minjie Wang, Lingfan Yu, Da Zheng, et al. Deep graph library: Towards efficient and scalable deep learning on graphs. *International Conference on Learning Representations Workshop on Representation Learning on Graphs and Manifolds*, 2019. URL https://arxiv.org/abs/1909.01315.

[287] Yilin Wang and Farzan Farnia. On the role of generalization in transferability of adversarial examples. *arXiv preprint arXiv:2206.09238*, 2022.

[288] Adam White, Kwun Ho Ngan, James Phelan, Kevin Ryan, Saman Sadeghi Afgeh, Constantino Carlos Reyes-Aldasoro, and Artur d'Avila Garcez. Contrastive counterfactual visual explanations with overdetermination. *Machine Learning*, pages 1–29, 2023.

[289] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1101. URL https://aclanthology.org/N18-1101.

[290] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8(3–4):229–256, may 1992. ISSN 0885-6125. doi: 10.1007/BF00992696. URL https://doi.org/10.1007/BF00992696.

[291] Zhengxuan Wu, Elisa Kreiss, Desmond Ong, and Christopher Potts. ReaSCAN: Compositional reasoning in language grounding. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021. URL https://openreview.net/forum?id=Rtquf4Jk0jN.

[292] Kai Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. *arXiv preprint arXiv:2006.09994*, 2020.

[293] Annie Xie, James Harrison, and Chelsea Finn. Deep reinforcement learning amidst lifelong non-stationarity. *CoRR*, abs/2006.10701, 2020. URL https://arxiv.org/abs/2006.10701.

[294] Danfei Xu, Yuke Zhu, Christopher Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.

[295] Kelvin Xu, Jimmy Lei Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 2048–2057. JMLR.org, 2015.

[296] Yang You, Igor Gitman, and Boris Ginsburg. Scaling SGD batch size to 32k for imagenet training. *CoRR*, abs/1708.03888, 2017. URL http://arxiv.org/abs/1708.03888.

[297] Roozbeh Yousefzadeh and Xuenan Cao. To what extent should we trust AI models when they extrapolate? *CoRR*, abs/2201.11260, 2022. URL https://arxiv.org/abs/2201.11260.

[298] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1511.07122.

[299] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. 2016. doi: 10.48550/ARXIV.1605.07146. URL https://arxiv.org/abs/1605.07146.

[300] Runtian Zhai, Chen Dan, Arun Suggala, J Zico Kolter, and Pradeep Ravikumar. Boosted cvar classification. *Advances in Neural Information Processing Systems*, 34, 2021.

[301] Cheng Zhang, Stefan Bauer, Paul Bennett, Jiangfeng Gao, Wenbo Gong, Agrin Hilmkil, Joel Jennings, Chao Ma, Tom Minka, Nick Pawlowski, et al. Understanding causality with large language models: Feasibility and opportunities. *arXiv preprint arXiv:2304.05524*, 2023.

[302] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Michael C. Mozer, and Yoram Singer. Identity crisis: Memorization and generalization under extreme overparameterization. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=B1l6y0VFPr.

[303] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

[304] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. 2018. doi: 10.48550/ARXIV.1805.08318. URL https://arxiv.org/abs/1805.08318.

[305] Marvin Mengxin Zhang, Henrik Marklund, Nikita Dhawan, Abhishek Gupta, Sergey Levine, and Chelsea Finn. Adaptive risk minimization: Learning to adapt to domain shift. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=-zgb2v8vV_w.

[306] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251, 2017. doi: 10.1109/ICCV.2017.244.

Appendix A

Additional background

This Appendix aims to cover a minimal background on neural networks and their training with the goal of producing a self-contained thesis that is accessible to general technical audience. However, these are the basic concepts seen in most machine learning textbooks, which is why they are not in the core part of this thesis.

A.1 Neural networks

All architectures used in Chapter 4 and Chapter 5 are neural networks, motivated by the rich visual and relational input data used in these chapters (for which neural networks are the 'gold standard' solution). Chapter 3 contains model-agnostic results: a set of problems where any classification/regression model can be applied, and a set of theoretical results that hold for any machine learning model (including neural networks).

A.1.1 Multi-layer perceptron

This section introduces the terminology which is used throughout the experimental chapters focused on neural networks (Chapter 4 and Chapter 5). The architectures used in the experiments in this thesis (Dilated DenseNets, Neural Function Modules and the baseline methods in Chapter 4, variants of graph neural networks, variants of convolutional neural networks and the baseline methods in Chapter 5) are all extensions of the multi-layer perceptron described in this section.

The *multi-layer perceptron* [234, 198, 243] is one of the simplest neural network architectures. Its aim is to learn to approximate a function mapping some input vectors $\boldsymbol{x}$ (*features*) into outputs $\boldsymbol{r}$. It consists of several *layers* of neurons, also called *perceptrons*. The concept of features is used throughout the dissertation, with a distinction between raw input features,

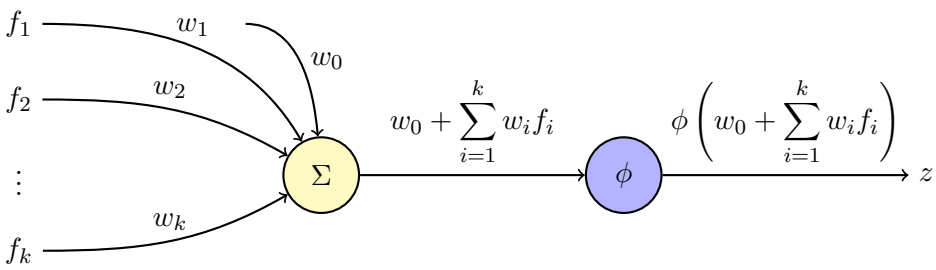


Figure A.1: A perceptron. The bias term is w_0, and the inputs f_i are multiplied by the corresponding weights w_i. The weighted sum is passed to the activation function ϕ and produces the output z of the perceptron.

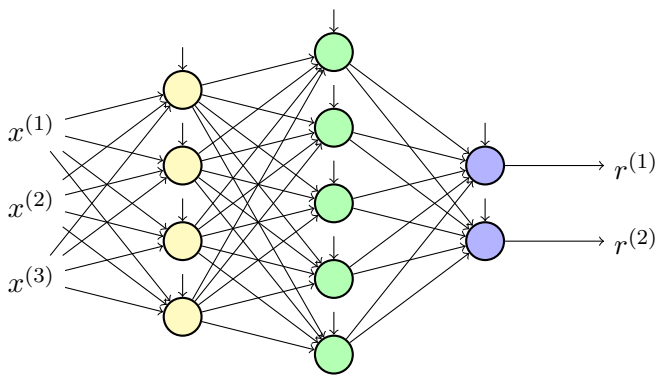


Figure A.2: A multi-layer perceptron neural network with three inputs and two outputs. Bias terms of each perceptron are represented as vertical arrows above the neuron. There are two hidden layers: one with four and another with five neurons.

such as image pixels throughout Chapter 4, and higher-level features, such as image shapes in Chapter 4.

Each perceptron (Figure A.1) is a function mapping k inputs $f_1, \ldots, f_k$ into a single output z (k is not necessarily the same for all neurons). The output is computed by evaluating a weighted sum $w_0 + \sum_{i=1}^{k} w_i f_i$, where the parameters $w_0, w_1, \ldots, w_n$ are called *weights* and w_0 in particular is the *bias* term. This weighted sum is then given as input to a function ϕ called the *activation function*, whose role is usually to add more complexity to the representation and introduce non-linearity. There are several commonly used activation functions:

Rectified linear unit (ReLU) $\mathsf{ReLU}(x) = \max(0, x)$.

Sigmoid $\mathsf{Sigmoid}(x) = \dfrac{1}{1 + \exp(-sx)}$ for some $s > 0$, which is a monotonically increasing function with asymptotes $y = 0$ for $x \to -\infty$ and $y = 1$ for $x \to +\infty$.

Hyperbolic tangent $\tanh(x) = \dfrac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)}$, whose shape is similar to the sigmoid.

In a multi-layer perceptron (Figure A.2), the output of a perceptron in one *hidden layer* (not the last layer) is used as the input for a perceptron in the next layer. The standard practice of using the entire output of a hidden layer as the input to the consecutive layer is modified in both Dilated DenseNets (Section 4.1.1) and Neural Function Modules (Section 4.1.2) with the aim of improving generalisation. The outputs of the final layer, the *output layer*, are the outputs of the neural network as a whole. Each neuron has a distinct set of weights.

In regression tasks, there is usually only one output neuron. In binary classification tasks, we can also get away with using only one output neuron, and the value of the output is used to assign the input to one of two classes by thresholding (for example, assign to class A if the output is less than 0.5, otherwise assign to class B). For more general classification tasks with $C > 2$ classes, we typically use C output neurons whose outputs need to then be transformed into a probability distribution over classes (indicating how likely it is that a particular input belongs to the corresponding class). The Softmax function serves this purpose – see Section 2.3.1.1.

A.1.2 Convolutional Neural Networks

All contributions made in Chapter 4 are based on Convolutional Neural Networks (CNNs) [165] and their shortcomings in the context of out-of-distribution generalisation in image recognition. This section describes Convolutional Neural Networks and how they learn local image features through the use of a small 'sliding window' known as *kernel* (Section A.1.2.2). The topic of learning local features by CNNs is later revisited in the description of Dilated DenseNets (Section 4.1.1), a CNN variant able to capture global features as well as local features. CNNs are also used in the second part of Chapter 5 in the experiments on multi-agent communication based on image data (Section 5.2). All methods that are applied to image data in this dissertation use a CNN as the backbone component.

It is a standard practice to use CNNs for processing data of a regular grid-like topology, such as images (two-dimensional grids of pixels) or time series (a one-dimensional grid of samples taken at regular time intervals). CNNs differ from other families of neural networks in using *convolution* instead of matrix multiplication in at least one of their layers [23]. CNNs are an appropriate choice for processing images due to *parameter sharing* – unlike in matrix multiplication, the number of parameters is constant with respect to the input size, which allows handling of large inputs and reduces the memory requirements. Furthermore, convolutions are ubiquitous in computer vision, since image processing operations such as Gaussian blurring and edge detection can be represented using convolutions.

A.1.2.1 Convolution

This section describes the concept of a *convolution*, on which *dilated convolution* described in Section 4.1.1.1 and used throughout the Chapter 4 is based.

A *convolution* of real functions $f : \mathbb{R} \to \mathbb{R}$ and $g : \mathbb{R} \to \mathbb{R}$ is a function $f * g$ defined as follows:

$$(f * g)(t) = \int_{-\infty}^{+\infty} f(x)g(t - x)\, dx. \tag{A.1}$$

As values stored in a digital form are necessarily discrete (for example, pixel intensities), convolution in the context of CNNs refers to a *discrete convolution*:

$$s[t] = (f * g)(t) = \sum_{x=-\infty}^{\infty} f[x]g[t - x]. \tag{A.2}$$

Equation (A.2) refers to convolution of two 1-dimensional signals. When working with images, the signals are 2-dimensional and they are represented as matrices I and K, where I is the input image and K is the *kernel* representing an image processing operation (the values are the inferred weights in a CNN). To implement an infinite sum over a finite number of elements, the functions are assumed to be equal to zero outside of the matrix dimensions. The convolution is applied over both axes of the input matrices:

$$S[i, j] = (I * K)[i, j] = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} I(n_1, n_2)K(i - n_1, j - n_2). \tag{A.3}$$

where i, j are discrete values indexing the matrices. Convolution is commutative, so Equation (A.3) is equivalent to:

$$S[i, j] = (I * K)[i, j] = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} I(i - n_1, j - n_2)K(n_1, n_2). \tag{A.4}$$

An example of a two-dimensional discrete convolution is shown in Figure A.3.

A standard image I has three *channels* (red, green and blue), and each of them can be represented as a matrix of pixel intensity values in the range of 0 to 255. Therefore, the Equation (A.3) is applied to the three matrices that represent the separate colour channels of I: I_R, I_G, I_B. In this case, each channel has a unique convolution kernel K.

Many neural network libraries (such as PyTorch [211]) implement a related operation called *cross-correlation* but call it convolution. Cross-correlation is the same as convolution

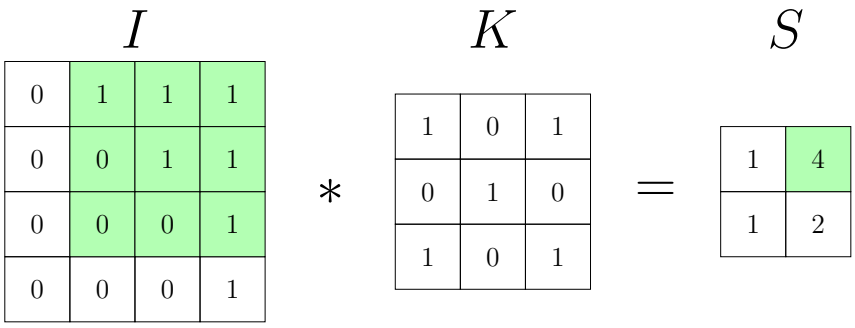


Figure A.3: Discrete convolution on a binary input. The first matrix represents an input I, the second one the kernel K, and the last one the output S. The middle cell in K is $K[0,0]$.

but the kernel is not flipped:

$$S[i,j] = (K * I)[i,j] = \sum_{n_1=-\infty}^{+\infty} \sum_{n_2=-\infty}^{+\infty} I[i+n_1, j+n_2] K[n_1, n_2]. \tag{A.5}$$

In this thesis I follow the convention of referring to both convolution and cross-correlation as convolution.

A.1.2.2 Convolutional neural networks

In the simplest case, a convolutional neural network is a sequence of *convolutional layers* (implementations of a convolution), *non-linear activation functions*, *pooling layers* and *fully connected* layers (matrix multiplication, as in a standard neural network).

The input I to the first *convolutional layer* in the network is the original image, and the consecutive layers receive a transformed representation of that image as an input.

The matrix K (*kernel*) contains the parameters to be learned through optimisation. Analogously to signal processing, a kernel can be viewed as a sliding window of a relatively small size that is convolved step by step with the whole image area (Figure A.3). The size of the kernel is typically much smaller than the image size, which leads to the desired properties of parameter efficiency and parameter sharing mentioned earlier. The output S represents local patterns in the image, such as corners and edges of shapes.

Non-linear activation functions are used to represent non-linearity in real image data. The Rectified Linear Unit (ReLU) (defined in Section A.1.1) has become the most popular choice since the publication of AlexNet [146]. The ReLU induces a sparse representation which has regularising properties [2]. The gradient of the ReLU function is either 0 for $a < 0$, or 1 for $a > 0$, which accelerates convergence in comparison to other conventional activation functions (sigmoid, tanh). These properties bring a substantial reduction in the

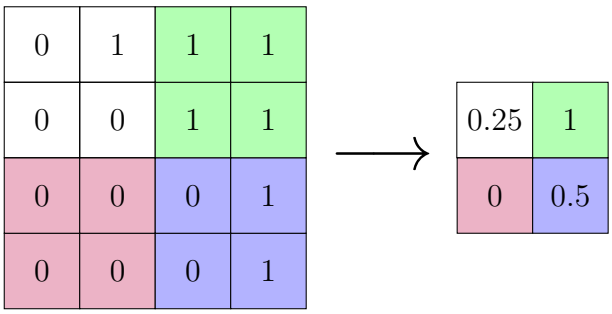


Figure A.4: Average 2D pooling of stride 2, kernel 2×2.

computational cost and learning time, which is particularly important in computationally expensive computer vision applications.

The final building block, *pooling*, replaces values in a matrix with a summary statistic of their neighborhoods (Figure A.4). Similarly to the convolution, the function is applied to each rectangular neighborhood of a given size. It reduces the spatial size of the representation, and the number of model parameters. As a result, it also controls overfitting. Moreover, pooling introduces approximate invariance to small local translations. This is desirable in image classification tasks since the existence of certain features is often more important than their precise locations. Similarly as in a convolutional layer, pooling is parametrised by the window size and stride (shift value at each step).

A.2 Training

The chapters focused on the experimental results obtained using neural networks (Chapter 4 and Chapter 5) mention training, for instance, in the context of the number of the training *epochs* (for instance, Section 4.2.2.2) and the distinction between training accuracy and test accuracy (for instance, Figure 4.10). This section contains a minimal description of how all the neural networks described in this thesis were trained in order to obtain the results shown in Chapter 4 and in Chapter 5.

To train a neural network means to find a set of weights that best fit the training inputs to the corresponding labels. This can be phrased as minimising a *error function* $\mathcal{L}$ which represents the deviation between expected outputs and the outputs computed by the neural network.[1] The error function is equal to the arithmetic mean of the values of an *loss function* ℓ when all data points are considered. The choice of loss function depends on the nature of the problem being solved (regression or classification) and the data

[1] In optimisation, when finding the set of parameters maximising some function, we usually refer to it as the *objective function*.

distribution (for example, if it is likely that a model is going to overfit to the training data, a regularisation term is added to the loss function).

Let the input to a neural network be a vector $\boldsymbol{x} \in \mathbb{R}^{d_{\text{in}}}$ and let the expected output be $\boldsymbol{y} \in \mathbb{R}^{d_{\text{out}}}$. The operation of the neural network is described by a function f which takes the input features $\boldsymbol{x}$ and a vector of weights $\boldsymbol{w}$ (which includes bias terms). We denote the output computed by such a neural network with $f(\boldsymbol{x}; \boldsymbol{w})$. The loss function $\mathcal{L}$ takes as arguments a sequence of n inputs $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$, a sequence of corresponding labels $\boldsymbol{Y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n)$, and weights $\boldsymbol{w}$, and we use a semicolon to separate the learned parameters $\boldsymbol{w}$ from the data operated on: $\mathcal{L}(\boldsymbol{X}, \boldsymbol{Y}; \boldsymbol{w})$. The goal is to find weights $\boldsymbol{w}^*$ minimising the error function:

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{X}, \boldsymbol{Y}; \boldsymbol{w}). \tag{A.6}$$

The following discussion about loss functions and solving Equation (A.6) is model-agnostic, since it applies to machine learning approaches beyond neural networks. The parameters being optimised are not necessarily neural network weights, so I use the symbol $\boldsymbol{\Theta}$ to refer to the more general setting. Equation (A.6) can thus be written more generally as:

$$\boldsymbol{\Theta}^* = \arg\min_{\boldsymbol{\Theta}} \mathcal{L}(\boldsymbol{X}, \boldsymbol{Y}; \boldsymbol{\Theta}). \tag{A.7}$$

A.2.1 Loss functions and regularisation

In regression tasks, there is usually one output of the function being approximated by a neural network. The output vectors $\boldsymbol{y}_i$ have a dimension of 1, indicating that they are scalars and we instead write y_i. In these tasks, a popular choice of error function is *mean squared error* (MSE), inspired by least-squares curve fitting:

$$\text{MSE}(\boldsymbol{X}, \boldsymbol{Y}; \boldsymbol{\Theta}) = \frac{1}{n} \sum_{i=1}^{n} (f(\boldsymbol{x}_i; \boldsymbol{\Theta}) - y_i)^2.$$

Optimising MSE can be shown to be equivalent to finding the *maximum likelihood* estimator, under the assumption of Gaussian noise. The *maximum a posteriori* estimator, in which the prior distribution exhibits a preference for smaller weights,[2] is achieved by using *L2-regularised* MSE with a *hyperparameter* λ:

$$\text{MSE\_L2}(\boldsymbol{X}, \boldsymbol{Y}; \boldsymbol{\Theta}) = \frac{1}{n} \sum_{i=1}^{n} (f(\boldsymbol{x}_i; \boldsymbol{\Theta}) - y_i)^2 + \lambda \|\boldsymbol{\Theta}\|^2.$$

[2]Specifically, the prior distribution on Θ is a zero-mean multivariate normal distribution in which the variance along each coordinate is λ.

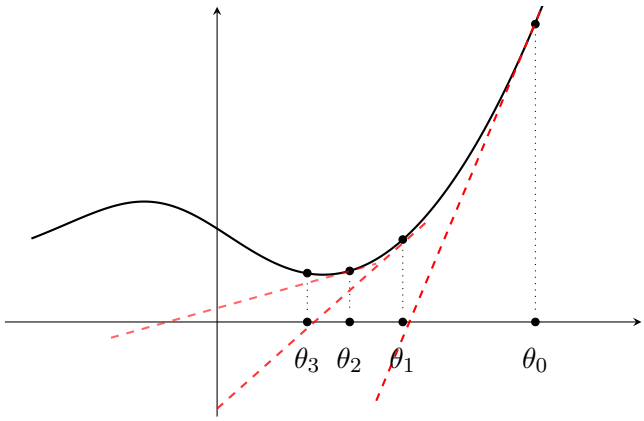


Figure A.5: Illustration of the first few three steps of the basic gradient descent algorithm (for 1-dimensional inputs) for a function $g(\theta)$. A sequence of solutions $\theta_0, \theta_1, \ldots$ is constructed recurrently by computing the gradients $\nabla g(\theta_i)$ – the greater the magnitude of the gradient, the greater the distance to the next approximate solution θ_{i+1}. The tangent (gradient in the 1-dimensional case) at $x = \theta_0$ is very steep, so a large correction in the direction of $\nabla g(\theta_0)$ is made. As we approach the minimum of g, the steps become smaller. Note that the function is not convex on its entire domain; if we are unlucky with the initial solution θ_0, the sequence might converge to another (local) minimum.

In classification tasks, loss functions based on entropy are more common.

A.2.2 Gradient descent

Finding an approximate solution to Equation (A.7) is possible numerically when the region within which a function is optimised is convex. *Gradient descent* is a local search algorithm based on the observation that, if the gradient $\nabla g(\boldsymbol{x})$ of a convex function g evaluated at $\boldsymbol{x}$ has a large magnitude (the length of the vector), it is likely to be further away from the minimum than if it is small. The gradient is usually interpreted as the direction in which the function is the steepest, meaning that $\boldsymbol{x} - \eta \nabla g(\boldsymbol{x})$ for a small $\eta > 0$ is likely to be closer to the minimum of the function than $\boldsymbol{x}$.

Assuming the initial values of the parameters are given by $\boldsymbol{\Theta}_0$, the algorithm proceeds by generating the sequence $\{\boldsymbol{\Theta}_i\}_{i=0}^{\infty}$ as follows:

$$\boldsymbol{\Theta}_{i+1} = \boldsymbol{\Theta}_i - \eta \nabla_{\boldsymbol{\Theta}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\Theta}_i). \tag{A.8}$$

The constant $\eta > 0$ is called the *learning rate* and it affects the speed of convergence of the sequence: higher values may get the sequence into a close neighbourhood of the true solution but have the risk of oscillating around it once they get there, whereas lower values risk taking too many steps to get to the close neighbourhood in the first place.

Optimisation algorithms based on gradient descent are commonly used in machine learning.

Examples of frequently used modern optimisation algorithms include Adam [138], also frequently used throughout this dissertation (Chapter 3 and Chapter 4), and Stochastic Gradient Descent [28] (Chapter 5).

A.2.3 Computing gradients

To use gradient descent for training neural networks such as MLPs, it is necessary to compute gradients of the loss function – meaning its partial derivatives with respect to all of the weights of the neural network. Assuming MSE, this means computing:

$$\frac{\partial}{\partial w_k}\left[\frac{1}{n}\sum_{i=1}^{n}\left(f(\boldsymbol{x}_i;\boldsymbol{w})-y_i\right)^2\right]=\frac{2}{n}\sum_{i=1}^{n}\left[\left(f(\boldsymbol{x}_i;\boldsymbol{w})-y_i\right)\frac{\partial f(\boldsymbol{x}_i;\boldsymbol{w})}{\partial w_k}\right].$$

It suffices to compute the partial derivatives of f with respect to the weights ($\partial f/\partial w_k$). The *backpropagation* algorithm [237] is used to achieve this and it relies on each operation of the neural network – trivially the weighted sums, but also the activation functions – being differentiable.

Chapter 3 discusses the limitations of the training method presented in this section. Minimising the average error over the entire dataset can lead to poor performance on under-represented groups (Section 3.2). Distributionally Robust Optimisation (Section 2.4.3.2 and Section 3.4) is an alternative to the Equation (A.6) and it aims to minimise the largest error over the set of predefined groups.